

Bachelorarbeit

Clustering- und Gruppierungstechniken für die Analyse von fMRT-Daten

Enrico Gebert

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik, Institut für Simulation und Graphik
Betreuer: Dipl.-Ing. Karin Engel
Gutachter: Prof. Dr.-Ing. Klaus Tönnies

Angefertigt am Leibniz Institut für Neurobiologie Magdeburg,
Speziallabor Nicht-invasive Bildgebung
Betreuer: Dr. André Brechmann

Eingereicht am 31.03.2010



Erklärung der Selbstständigkeit

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate deutlich kenntlich gemacht zu haben.

Magdeburg, den 31.03.2010

Enrico Gebert

Inhaltsverzeichnis

Zusammenfassung.....	6
1. Einleitung	6
1.1. Motivation	7
1.2. Zielstellung.....	8
2. Grundlagen.....	9
2.1. Funktionelle Magnetresonanztomographie.....	10
2.2. Statistische Karten	11
2.3. Verfahren zur Bestimmung von Regionen	13
2.4. Segmentierung.....	15
3. Umsetzung	15
3.1. Problemanalyse	15
3.1.1. Datenanalyse	16
3.1.2. Annahmen	17
3.1.3. Algorithmenwahl.....	21
3.2. Verwendete Verfahren zur Segmentierung	24
3.2.1. Region-Growing auf statistischen Karten.....	24
3.2.2. Region-Growing Variante 2	27
3.2.3. Wasserscheidentransformation	29
3.2.4. Multi-Scale-Region-Tree.....	31
3.2.5. Ausgabe	34
3.3. Implementierung.....	35
4. Experimentelle Validierung.....	37
4.1. Experimente.....	37
4.1.1. Ziel	37
4.1.2. Verwendete Daten	38
4.1.3. Durchführung	38
4.1.4. Ergebnisse	40
4.1.5. Auswertung	45
5. Fazit.....	47
6. Ausblick	50
7. Bibliographie	51

Zusammenfassung

In dieser Arbeit werden verschiedene Methoden präsentiert, um mithilfe von statistischen Karten, die aus funktionellen Magnetresonanztomographie-Untersuchungen gewonnen wurden, funktionelle Areale im Gehirn zu identifizieren. Dabei kommen die aus der Bildverarbeitung bekannten Verfahren des Region-Growing und der Wasserscheidentransformation zum Einsatz, die auf die Problemstellung der Segmentierung auf statistischen Karten angepasst wurden.

Weiterhin wird mit dem Multi-Scale-Region-Tree eine Datenstruktur präsentiert, die mithilfe der Wasserscheidentransformation auf Graphenebene eine hierarchische Repräsentation der erzeugten Regionen und das Speichern von Zusatzinformation für die gefundenen Regionen ermöglicht und als Grundlage für ein interaktives Interface dienen kann.

Zusätzlich wurde ein Programm entwickelt, mit dem die vorgestellten Methoden demonstriert und getestet werden können, um eine Plausibilitätsprüfung durchzuführen und damit das Potenzial und die Einschränkungen der gewählten Verfahren zu demonstrieren.

1. Einleitung

Die Identifikation funktioneller Hirnareale mittels funktioneller Magnetresonanztomographie (fMRT) stellt ein komplexes Problem dar. Zum einen ist aufgrund der großen hirnanatomischen Unterschiede die räumliche Zuordnung von aktivierten Regionen zwischen verschiedenen Probanden erschwert, zum anderen ist auch die exakte Relation zwischen funktionellen Arealen und anatomischen Landmarken nicht eindeutig. Weiterhin sind die Daten zumeist durch Artefakte wie Rauschen oder Bewegungsartefakte gestört und es können keine genauen Aussagen über die Zuverlässigkeit der Daten getroffen werden. Das Ziel der Auswertung von fMRT-Daten ist es, funktionelle Hirnregionen zu identifizieren und diesen bestimmte Aufgabentypen zuzuordnen.

1.1. Motivation

Im Rahmen dieser Arbeit sollen Verfahren vorgeschlagen werden, mittels derer Cluster in statistischen Aktivierungskarten gebildet werden können. Diese Cluster stellen zusammenhängende Gruppen von Voxeln mit bestimmten Eigenschaften (z.B. Größe, Lage, Form, signifikanter und ähnlicher Signalverlauf) dar. Aufgrund von vorgegebenen Einschränkungen können diese dazu dienen, funktionelle Regionen zu lokalisieren und zu unterscheiden.

Die Analyse solcher statistischen Karten erfolgt in der Regel durch Experten und benötigt für eine umfassende Analyse der Daten viel Zeit (mehrere Wochen). Die lange Analysedauer begründet sich zum einen in der Menge der Daten und zum anderen in den Programmen, die den Experten zur Analyse zur Verfügung stehen. Ein solches Programm ist zum Beispiel BrainVoyager™. Mithilfe von BrainVoyager™ können die statistischen Karten berechnet, angezeigt und auf anatomische Datensätze abgebildet werden, um den Ort der Aktivierung zu identifizieren. Die genaue Definition von Clustern auf den Karten, die mit BrainVoyager™ angezeigt werden, gestaltet sich allerdings schwierig, da es nötig ist, die Darstellungsparameter immer wieder anzupassen, um die Cluster zu lokalisieren. Ein Beispiel für solch einen Darstellungsparameter ist der obere Schwellenwert. Alle Werte die größer als dieser Schwellenwert sind, werden in einer Farbe dargestellt, was dazu führt, dass höhere Werte nicht mehr unterscheidbar sind. Ein hoch gewählter Schwellenwert sorgt im Gegenzug dafür, dass zwar die hohen Werte unterscheidbar sind, aber dafür ist es schwer, lokale Unterschiede in den Bereichen mit niedrigen Aktivierungen zu finden. Das führt dazu, dass der Schwellenwert je nach untersuchter Region im Gehirn immer wieder angepasst werden muss, um einzelne Cluster zu identifizieren.

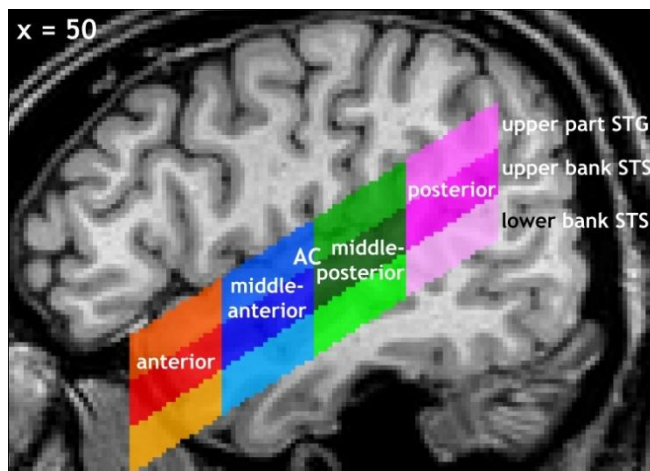


Abbildung 1: Darstellung von definierten Regionen des Temporallappens.

Weiterhin wird aufgrund des enormen Zeitaufwands und der Unterschiede der Anatomie zwischen verschiedenen Probanden häufig auch auf eine genaue Einteilung der Cluster verzichtet und stattdessen auf eine grobe Einteilung zurückgegriffen. Eine Möglichkeit ist es, Regions of Interest (ROI) zu definieren, die zum Beispiel eine rechteckige Form haben und die Region abdeckt, die analysiert werden soll (Abbildung 1). Durch die Form der ROIs werden allerdings auch Voxel in die statistische Auswertung einbezogen, die eigentlich nicht zu der zu untersuchenden Region gehören. Diese Voxel können das Ergebnis der Berechnungen verfälschen.

Ein automatisches Clustering auf den statistischen Karten würde eine enorme Zeitersparnis für die Experten bedeuten. Die Ergebnisse könnten als Anhaltspunkt für die Experten dienen, um eine genauere Analyse der Cluster durchzuführen. Weiterhin könnte eine solche Methode dazu dienen Cluster zu erzeugen, die sich besser an die Form der Daten (die Anatomie) anpassen. Eine statistische Analyse bei einzelnen Probanden könnte somit genauer durchgeführt werden, da weniger Voxel vorhanden sind, die eigentlich nicht zu einer Region gehören.

1.2. Zielstellung

Für die maschinelle Analyse der statistischen Karten werden im Rahmen dieser Arbeit verschiedene Ansätze untersucht, wobei die eingesetzten Verfahren weitestgehend automatisiert und möglichst wenig von Modellannahmen des zugrundeliegenden Experimentes beeinflusst sein sollen. Letzteres ist wichtig, damit verhindert wird, dass die Ergebnisse durch zu spezifische Modellannahmen (z.B. Form, Lage, Aktivierungsstärke der Region...) beeinflusst werden.

Wünschenswert ist ein Verfahren, welches eine allgemeine Analyse des Datensatzes berechnet und Ergebnisse dann anhand von spezifischen Anfragen des Nutzers konkretisiert werden können.

Um ein möglichst robustes Verfahren zu erhalten ist es weiterhin nötig geeignete Kriterien und Maßzahlen zu finden, anhand derer Regionen mit spezifischen Eigenschaften – sprich mögliche funktionelle Areale – als solche erkannt oder voneinander unterschieden werden können. Ein robustes Verfahren ist wünschenswert, da Ergebnisse leichter reproduziert werden können und die Abhängigkeit von Parametereingaben reduziert wird.

Das Ziel dieser Arbeit ist es, Verfahren vorzuschlagen, welche die aus den fMRT-Daten entstandenen statistischen Karten auf mögliche funktionell abgrenzbare Gehirnareale untersuchen. Hierzu sollen die eingesetzten Verfahren datengetrieben sein, um die Unabhängigkeit von Modellannahmen der zugrundeliegenden Experimente zu gewährleisten. Die Unabhängigkeit von Modellannahmen aus dem Experiment ist wichtig, damit ein möglichst objektives Ergebnis der Analyse erzeugt werden kann.

Um den Einsatz der Verfahren zu erleichtern und die Robustheit des Verfahrens zu gewährleisten ist es nötig, die Anzahl der vom Anwender zu bestimmenden Parameter für die Analyse möglichst gering zu halten. Weiterhin sollten sich die Parameter auf Kriterien und Maßzahlen beschränken, die den Experten vertraut sind (zum Beispiel: Regionsgröße, ein unterer Schwellenwert, der angibt, ab wann ein Voxel betrachtet wird). Solche Parameter erleichtern es dem Anwender, die richtigen Parameterwerte für die Analyse auszuwählen.

Um die Verfahren später mit einem Interface und Interaktionsmethoden ergänzen zu können, soll eine Datenstruktur entwickelt werden, die die relevanten Daten aus den statistischen Karten sowie gefundene Regionen und ihre statistische Werte und Maßzahlen speichern und verwalten kann. Das Interface soll sich im Rahmen dieser Arbeit auf eine einfache Konsolenausgabe beschränken, da diese genügt, um die Verfahren zu testen. Dennoch soll die genannte Datenstruktur eine spätere Erweiterung um Interaktionsmethoden erleichtern.

Die Auswertung der ermittelten Regionen soll insofern erfolgen, dass bestimmt werden kann, ob die Ergebnisse plausibel sind und es sinnvoll ist, eine Segmentierung auf statistischen Karten durchzuführen. Eine genaue Analyse der Ergebnisse ist nicht Teil dieser Arbeit, da es keinen „Goldstandard“ für die untersuchten Daten gibt. Es ist nicht möglich, eine konkrete Aussage über die Korrektheit der Ergebnisse zu machen.

2. Grundlagen

In diesem Abschnitt der Arbeit sollen einige grundlegende Begriffe und Methoden erläutert werden, die für das Verständnis der folgenden Abschnitte nötig sind.

2.1. Funktionelle Magnetresonanztomographie

Bei der funktionellen Magnetresonanztomographie (fMRT) handelt es sich um ein nicht-invasives, hochauflösendes bildgebendes Verfahren. Die fMRT ist eine Weiterentwicklung der klassischen Magnetresonanztomographie. Durch die fMRT wird versucht die neuronale Aktivität des Gehirns zu messen. Dabei wird die Aktivität des Gehirns nicht direkt gemessen, sondern es werden Durchblutungsänderungen (BOLD-Effekt), die auf Stoffwechselfvorgänge zurückgeführt werden, gemessen. Es wird angenommen, dass diese Stoffwechselfvorgänge im Zusammenhang mit der neuronalen Aktivität des Gehirns stehen.

Der Blood Oxygen Level Dependency Effekt (BOLD-Effekt) führt zu einer Veränderung der gemessenen Signale, da sauerstoffarmes Blut andere magnetische Eigenschaften besitzt als sauerstoffreiches Blut. Ist ein Hirnareal aktiv, benötigt es mehr Sauerstoff. Dieser Sauerstoff wird durch eine erhöhte Durchblutung der betreffenden Region zugeführt, was zu einer Veränderung des lokalen Magnetfelds in dem betreffenden Bereich führt. Durch den Vergleich von Aufnahmen desselben Ortes zu unterschiedlichen Zeitpunkten (Experimentalbedingung, Ruhebedingung) können nun mithilfe von statistischen Verfahren Rückschlüsse auf die neuronale Aktivität gezogen werden (siehe 2.2). Wichtig für die Auswertung der gemessenen Signale ist, dass der BOLD-Effekt verzögert auftritt, da der Regulierungsmechanismus für die Durchblutung (hämodynamische Antwort (HDA)) relativ träge ist (Kruggel et al. 1999). Einen typischen Signalverlauf für den BOLD-Effekt zeigt das folgende Diagramm (Abbildung 2). Die Baseline stellt die neuronale Aktivität bei Ruhe dar. Nach der Präsentation eines Stimulus ist für kurze Zeit (1-2 Sekunden) weniger

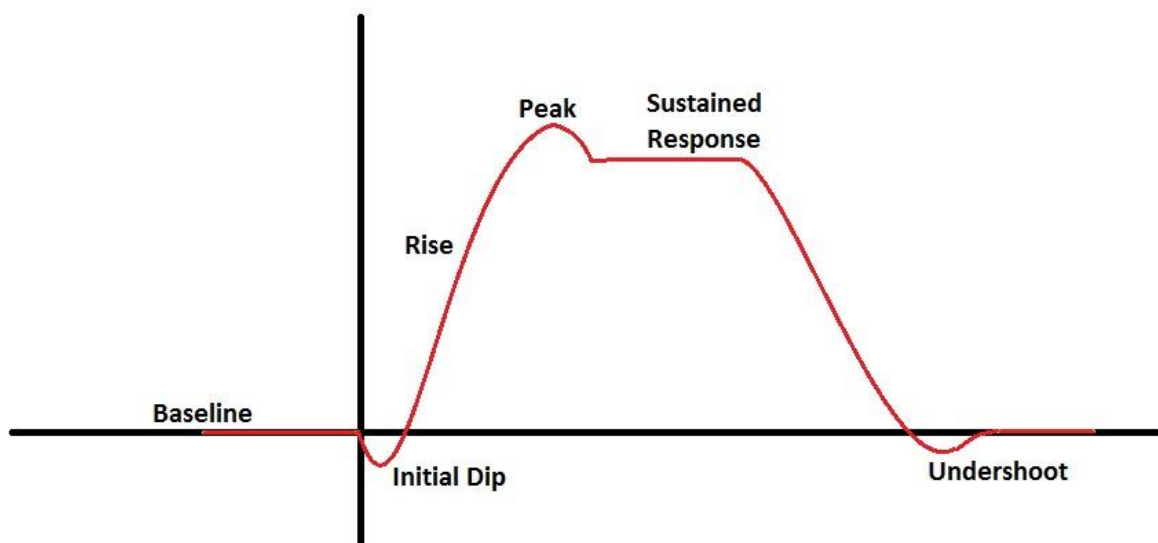


Abbildung 2: Diagramm einer 2-Gamma hämodynamischen Antwortfunktion.

sauerstoffreiches Blut in der aktivierten Region (Initial Dip), da diese den vorhandenen Sauerstoff verbraucht. Durch den erhöhten Sauerstoffverbrauch der Region reguliert der Körper die Zufuhr von sauerstoffreichem Blut „nach oben“ (Rise). Diese Zufuhr geschieht solange, bis die Menge an neu zugeführtem sauerstoffreichem Blut den Verbrauch der aktiven Gehirnregion deckt. Dabei entsteht ein Peak circa nach 4-6 Sekunden. Ist der Peak erreicht, befindet sich in der Region mehr Blut als benötigt, was dazu führt, dass der Körper erneut nachreguliert und die Zufuhr von sauerstoffreichen Blut senkt. Nach der Regulierung der Blutzufuhr hält der Blutfluss ein gewisses Niveau, solange die Region aktiv ist. Ist die Aktivität der Region beendet, wird die Zufuhr von sauerstoffreichem Blut wieder vermindert, bis die Baseline wieder erreicht ist. Dabei entsteht bei der Regulierung ein kurzer Undershoot, bis die Blutzufuhr wieder ihren ursprünglichen Wert erreicht hat.

Damit es möglich ist, aus den gemessenen Daten Rückschlüsse auf den Ort der Aktivierung zu ziehen, wird in der Regel vor einem fMRT-Experiment eine hochauflösende MRT-Aufnahme der Anatomie des Probanden gemacht. Mit dieser MRT-Aufnahme werden dann die fMRT-Aufnahmen registriert.

2.2. Statistische Karten

Bei den im Rahmen dieser Arbeit verwendeten Daten handelt es sich nicht um die fMRT-Daten. Vielmehr werden aus diesen Daten statistische Karten erzeugt, auf denen anschließend die Datenanalyse durchgeführt wird.

Um zu verstehen, welche Art von Daten zu erwarten ist, ist es nötig, etwas über den Versuchsaufbau der fMRT-Experimente zu wissen. Aus diesem lässt sich erklären, auf welchen Annahmen die statistischen Karten basieren.

In der Regel besteht ein fMRT-Experiment aus verschiedenen Stimulus- und Ruhephasen. Dem Probanden wird für eine kurze Zeitspanne (in der Regel im Sekundenbereich, beispielsweise beim event-related Design etwa 5 Sekunden oder beim block Design etwa 10 Sekunden) ein Stimulus präsentiert. Anschließend erfolgt eine kurze Ruhephase, in der kein Stimulus präsentiert wird. Nach der Ruhephase beginnt ein neuer Stimulus mit anschließender Ruhephase bis das Experiment abgeschlossen ist. Dabei können die Stimuli unterschiedlicher Modalität sein (zum Beispiel: auditiv, visuell, sensorisch).

Während des gesamten Experimentes wird die Aktivität des Gehirns gemessen, auf deren Basis die in dieser Arbeit verwendeten statistischen Karten erzeugt werden können.

Diese statistischen Karten werden zum Beispiel mithilfe des General Linear Model (GLM) (Friston et al. 1995) erzeugt, welches auch in BrainVoyagerTM implementiert ist.

Das GLM versucht die Variation von abhängigen Variablen mithilfe einer linearen Kombination (gewichtete Summe) von verschiedenen Referenzfunktionen zu „erklären“ oder „vorherzusagen“. Hierbei entsprechen die abhängigen Variablen dem gemessenen Zeitverlauf eines Voxels und die Referenzfunktionen stehen für einen angenommenen idealisierten Antwortzeitverlauf abgeleitet von den präsentierten Stimuli (Bedingungen). Eine Anzahl von spezifischen Referenzfunktionen bildet die Designmatrix (auch Modell genannt). Der Zeitverlauf einer Referenzfunktion wird mit der standardisierten hämodynamischen Antwortfunktion (2-Gamma-HDA) gefaltet. Jede Referenzfunktion erhält eine Gewichtung (Beta-Gewicht b), welche angibt, wie gut die Funktion den Zeitverlauf des Voxels y erklärt. Der Zeitverlauf des Voxel y ergibt sich aus der gewichteten Summe aller Referenzfunktionen X mit ihren korrespondierenden Beta-Gewichten b . Da das Modell aufgrund von Rauschen den Voxelzeitverlauf nicht perfekt erklären kann, wird dem Gleichungssystem ein Fehlerterm e hinzugefügt. Das so entstandene Gleichungssystem besteht aus n Datenpunkten und p Referenzfunktionen:

$$y = bx + e. \quad (1)$$

Hierbei entspricht der Vektor y den Zeitpunkten des gemessenen Zeitverlaufes des Voxels, wobei im Vektor y y_1 dem ersten gemessenen Zeitpunkt und y_n den n -ten gemessenen Zeitpunkt entspricht. Dieser Zeitverlauf wird durch die rechte Seite der Gleichung „erklärt“. Nach der Berechnung geben die Beta-Gewichte an, wie stark die jeweilige Referenzfunktion den Zeitverlauf des Voxels erklärt. Die exakte Interpretation der Beta-Werte hängt von der zugrundeliegenden Designmatrix ab. Ein großer positiver oder negativer Wert deutet auf eine starke Aktivierung oder Deaktivierung während der modellierten Bedingung gegenüber der Baseline hin. Alle Beta-Gewichte zusammen charakterisieren die Übereinstimmung eines Voxelzeitverlaufes mit einer oder mehreren der im Experiment gestellten Bedingungen. Die letzte Spalte des Gleichungssystems enthält die Fehler, auch Residuals genannt. Diese Fehlerwerte geben die Abweichung der berechneten gewichteten Referenzfunktion vom jeweiligen Voxelzeitverlauf an.

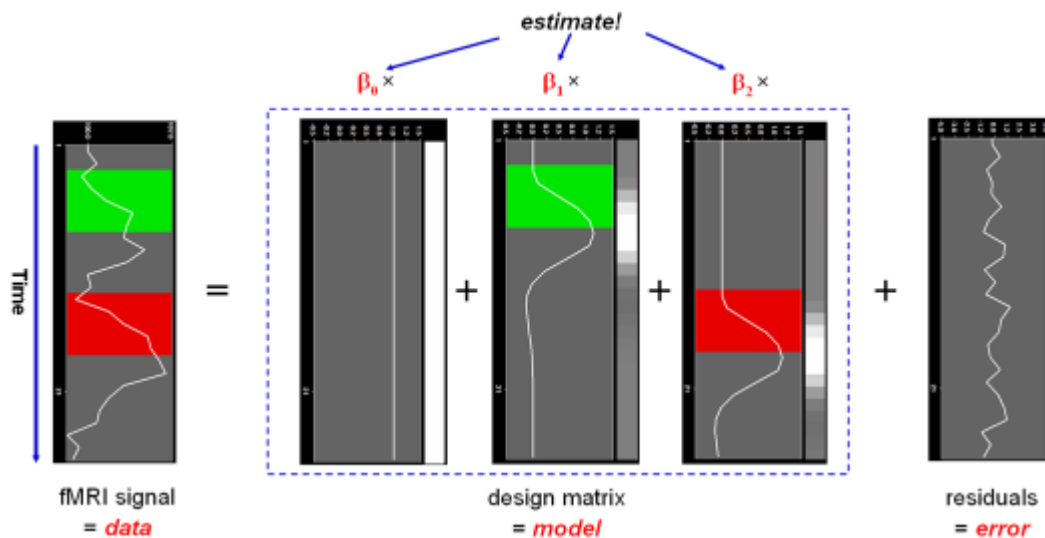


Abbildung 3: Grafische Darstellung des GLM. Links ist der Zeitverlauf aus den Daten zu sehen (data). Das Modell rechts besteht aus der Baseline und zwei Referenzfunktionen, die als Erklärung des Zeitverlaufes dienen sollen. Die Residuals zeigen den Verlauf der Fehlerfunktion an.

Das GLM wird für jedes Voxel im fMRT-Datensatz angepasst, wodurch ein Satz von Beta-Gewichten für jedes dieser Voxel entsteht. Durch statistische Hypothesen-Tests (F-Test, t-Test) kann die Signifikanz eines Voxels bestimmt werden. Dabei ist es auch möglich, spezielle Kontraste (zum Beispiel bei Bedingung 1 aktiv, aber nicht bei Bedingung 2) in die Berechnung mit einfließen zu lassen. Die aus einem t-Test resultierende statistische Karte enthält dann für jedes Voxel einen Wert, der angibt wie gut er dem Kontrast entspricht (je höher der t-Wert, desto höher die Übereinstimmung).

Das Ziel der Analyse der fMRT-Daten mithilfe der statistischen Karten ist es, Areale mit einer spezifischen Funktion im Gehirn zu lokalisieren, um so ein besseres Verständnis für die Arbeitsweise des menschlichen Gehirns zu erhalten.

2.3. Verfahren zur Bestimmung von Regionen

Es gibt verschiedene Ansätze, um Cluster in fMRT-Daten zu segmentieren. Dabei verwenden diese Verfahren in der Regel die Informationen über die Zeitverläufe aus den fMRT-Daten und nicht die in dieser Arbeit verwendeten statistischen Karten. Beispiele für solche Verfahren sind unter anderen eine konnektivitätsbasierte Methode (Deleus et al. 2009), bei der Regions of Interest (ROI) basierend auf ihrer funktionellen Verbundenheit gebildet werden. Diese Methode basiert auf einer Korrelationsanalyse, um maximal korrelierte

Signale, die nicht mit dem Signalrauschen korrelieren, zu identifizieren. Anschließend wird getestet ob ROIs mit ähnlicher funktioneller Verbundenheit zusammengefasst werden können. Dabei handelt es sich um ROIs, die sich teilweise überlappen.

Eine weitere Methode ist das Support Vector Clustering (SVC) (Wang et al. 2005). Beim SVC werden Fourierkoeffizienten als Features aus den fMRT Zeitverläufen extrahiert und für ein Clustering verwendet. Diese Features werden in einen hochdimensionalen Kernel-Space transformiert. In diesem Kernel-Space wird eine Menge an Clustergrenzen im Feature-Raum gesucht.

Die Clustered Components Analysis (Chen et al. 2004) ist eine Methode, bei der es sich um ein amplitudenunabhängiges Clustering handelt, welches auf einem expliziten statistischen Datenmodell basiert.

Eine weit verbreitete Methode zum Clustern von fMRT-Daten ist die Independent Component Analysis (ICA) (Formisano et al. 2004), welche auch in BrainVoyagerTM implementiert ist. Es handelt sich um ein Verfahren, welches entwickelt wurde, um eine Anzahl an statistisch unabhängigen Quellsignalen zu trennen wobei keine Information über ihre Verteilung oder Dynamik vorliegt. Bei der ICA wird angenommen, dass N statistisch unabhängige Eingaben zu N Ausgabekanälen linear überlagert wurden. Mit der Annahme, dass die Zeitverläufe unabhängig sind, wird versucht die, ICA zu lösen.

Da die gesuchten Areale im Gehirn in sich geschlossen sind, es muss also ein direkter Pfad zwischen jedem einer Region zugeordneten Voxel bestehen, folgt, dass der Einsatz von Methoden, die geschlossene Regionen bilden, für die Problemstellung sinnvoll ist. Es bietet sich an, anstelle der hier genannten Clustertechniken Segmentierungsverfahren einzusetzen, insbesondere auch weil Informationen über die Zeitverläufe der Voxel, die bei den genannten Verfahren genutzt werden, in den, im Rahmen dieser Arbeit genutzten statistischen Karten, nicht vorhanden sind. Wie im ersten Abschnitt erwähnt, ist es ein Ziel dieser Arbeit datengetriebene Verfahren einzusetzen, was ebenfalls gegen den Einsatz von Verfahren wie der Clustered Components Analyse, die ein explizites Datenmodell benötigt, spricht.

2.4. Segmentierung

Bei der Segmentierung handelt es sich um eine Zerlegung des Datensatzes in Einheiten, die Träger von semantischen Informationen sein können. Diese Einheiten zeichnen sich dadurch aus, dass sie in sich geschlossene Segmente bilden. Im Rahmen dieser Arbeit handelt es sich bei Segmenten (Regionen) um mögliche funktionelle Areale im Kortex.

Eine Übersegmentierung bezeichnet im Rahmen dieser Arbeit das Entstehen von mehr als durch die Experten vorhergesagten Segmenten während einer Segmentierung. Die Übersegmentierung ist meist eine Folge von verrauschten Daten, die dazu führen, dass an Orten Regionengrenzen gefunden werden, an denen keine Regionengrenzen vorhanden sind.

Im Verlauf dieser Arbeit bezeichnet der Begriff regionenbildendes Zentrum (regionenbildendes Maximum) oder auch kurz Regionenzentrum das Voxel, welches das initiale (erste) Voxel für die gebildete Region darstellt.

3. Umsetzung

3.1. Problemanalyse

Aus der Aufgabenstellung ergeben sich die folgenden Problemstellungen, auf die in diesem Abschnitt genauer eingegangen wird.

Als erstes ist die Frage zu klären, ob es möglich ist, auf den vorliegenden t-Karten funktionell abgegrenzte Regionen zu finden. Dazu muss geklärt werden, nach welchen Kriterien Regionen getrennt werden können.

Ein weiterer Punkt ist das Ermitteln von Einschränkungen und Rahmenbedingungen für die Algorithmenwahl, die sich aus den gegebenen Daten und ihrem Aufbau ableiten.

Anhand der getroffenen Analysen kann anschließend entschieden werden, welche Algorithmen und Methoden zur Lösung des Problems herangezogen werden können.

3.1.1. Datenanalyse

Bei den vorliegenden Daten handelt es um t-Karten. Wie unter 2.2 beschrieben, handelt es sich hierbei um statistische Karten die eine konkrete Fragestellung beantworten sollen. Ein Beispiel hierfür ist ein Experiment mit drei Klassen von experimentellen Stimuli: Menschenstimmen, Tierlauten und Musikinstrumenten. Eine konkrete Fragestellung im Bezug auf dieses Experiment ist, ob es eine stärkere Aktivierung bei Menschenstimmen als bei Tierlauten oder Musikinstrumenten gibt. Diese Fragestellung zielt darauf ab, Regionen im Gehirn zu finden, die selektiv auf Menschenstimmen reagieren (Bethmann et al. 2009).

Abbildung 4 zeigt ein Beispiel für eine t-Karte. Auf einer solchen Karte sind die positiven Werte von orange (niedrig) bis gelb (hoch) und die negativen Werte von blau (niedrig) bis grün (hoch) eingefärbt. Die Frage ist nun, welche Informationen aus den Daten extrahieren werden können und wie sie aufgebaut sind.

Zum Aufbau lässt sich sagen, dass es sich um einen 3D-Datensatz von Fließkommazahlen handelt. Wie unter 2.2 beschrieben gilt, dass, je höher der statistische Wert eines Voxels ist, desto stärker ist die Aktivierung. Negative Werte geben an, dass der Zeitverlauf der Voxel der entgegengesetzten Hypothese entspricht.

Für Experten ist es möglich, auf solchen statistischen Karten verschiedene Regionen zu erkennen. Zum Beispiel wird ein Datensatz mithilfe von BrainVoyager™ betrachtet und untersucht. Um Regionen zu identifizieren wird auf der statistischen Karte nach Gebieten mit

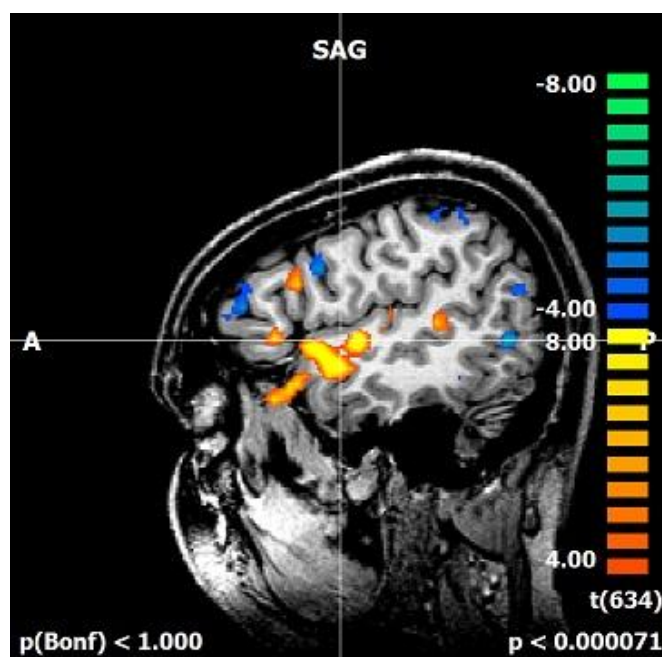


Abbildung 4: Abbildung einer statistischen (t) Karte auf die Anatomie. Die gelben Farbtöne repräsentieren Voxel die der Hypothese entsprechen, die blauen Farbtöne repräsentieren Voxel, die der entgegengesetzten Hypothese entsprechen.

einer ähnlichen Färbung gesucht. Allerdings gibt es viele Regionen, die man beim Betrachten der Karten mit Programmen wie BrainVoyager™ nur dann unterscheiden kann, wenn man die Anzeigeparameter stark modifiziert. Diese Modifikationen müssen für die Regionen mit verschiedenen Parametern gemacht werden und so ist es sehr umständlich die einzelnen Regionen genau zu lokalisieren. Beispiele für solche Parameter die in BrainVoyager™ verwendet werden sind ein unterer und oberer Schwellenwert. Hierbei werden Werte, die kleiner als der untere Schwellenwert sind, nicht angezeigt. Werte, die größer als der obere Schwellenwert sind, werden in die Anzeigefarbe des oberen Schwellenwertes gefärbt, wodurch Unterschiede zwischen Werten größer des oberen Schwellenwertes nicht mehr erkennbar sind. Ein weiterer Anzeigeparameter ist die Interpolation. Mit der Interpolation kann bestimmt werden, ob die Daten interpoliert angezeigt werden sollen. Diese Anzeige dient dazu, die Daten besser an die Anatomie anzupassen. Durch die Interpolation werden allerdings zumeist die Originalwerte der Daten verfälscht. Die Einstellung der minimalen Clustergröße dient dazu, kleine zusammenhängende Gebiete aus der Anzeige zu entfernen. Durch die vielen Parameter ist auch ein gleichzeitiges Anzeigen verschiedener Regionen erschwert.

Man geht allgemein davon aus, dass Regionen ein lokales Maximum als Zentrum haben und dass die Werte um solch ein Maximum ähnlich zu diesem sind und zum Rand der Regionen hin abfallen. Für die Werteverteilung nimmt man eine Gaußverteilung an. Die Regionen kann man sich dann als „Gauß-Blobs“ vorstellen. Durch die angenommene Gaußverteilung ergeben sich zwischen einzelnen Regionen Täler, die als Regionsgrenzen angesehen werden können.

Mit den Informationen aus der Datenexploration können nun Rahmenbedingungen und Einschränkungen getroffen werden, die für die Wahl der Algorithmen notwendig sind.

3.1.2. Annahmen

Die folgenden Annahmen, die sich aus dem vorhergehenden Abschnitt begründen, ergeben Kriterien, die für die Wahl und Umsetzung von geeigneten Methoden zum Auffinden von Regionen dienen:

1. Wie im vorherigen Abschnitt erläutert, handelt es sich um einen 3D-Datensatz von statistischen Werten. Daraus folgt, dass der verwendete Algorithmus auf 3D-Daten arbeiten muss, die pro Datenpunkt einen diskreten Wert darstellen.
2. Das Zentrum einer Region stellt sich als lokales Maximum dar. Um dieses Maximum fallen die Werte wieder ab. Die Werte werden als gaußverteilt angenommen und bilden wie in 3.1.1 beschrieben sogenannte „Gauß-Blobs“.
3. Aus (2) ergeben sich zwischen den einzelnen Regionen Täler, die als Regionsgrenzen angenommen werden können.
4. Werte innerhalb einer Region sind sich ähnlich. Zusammen mit (2) und (3) ergibt sich, dass es möglich ist, die lokalen Maxima als Eingabe für einen Segmentierungsalgorithmus zu nutzen. Weiterhin ist die Verwendung von Ähnlichkeitsmaßen (beispielsweise ein Unterschied der Intensitäten der Voxel) möglich.
5. Die den statistischen Karten zugrundeliegenden fMRT-Daten enthalten, wie im Abschnitt 1 beschrieben, Artefakte wie Rauschen oder Bewegungsartefakte. Daraus folgt, dass auch die aus diesen Daten entstehenden statistischen Karten nicht rauschfrei sind. Für die Verwendung von lokalen Maxima als Eingabe für Segmentierungsalgorithmen bedeutet dies, dass davon ausgegangen werden muss, dass durch das Rauschen bedingt nicht jedes Maximum auch ein Regionenzentrum ist. Zusätzlich verschlechtert sich die lokale Ähnlichkeit der Werte. Es werden also geeignete Vor- und Nachbearbeitungsschritte zur Berücksichtigung des Rauschens benötigt.
6. Nicht alle Daten, die die statistischen Karten enthalten, sind für die Segmentierung relevant. Wie im Abschnitt 2.2 beschrieben, geben hohe positive Werte eine starke Übereinstimmung eines Voxelzeitverlaufes mit der aufgestellten Hypothese an. Negative Werte geben an, dass sie nicht zutrifft. Weiterhin gelten Voxel erst dann als signifikant aktiv, wenn das Signal eine gewisse Stärke erreicht, also einen Schwellenwert überschreitet. Die je nach Fragestellung des Anwenders für die Untersuchung nicht relevanten Daten können in einem Vorverarbeitungsschritt entfernt werden. Im Rahmen dieser Arbeit beschränkt sich die Untersuchung auf die positiven Werte. Durch das Wegfallen von negativen Werten und positiven Werten unter einem festgelegten Schwellenwert wird der Suchraum für die

Regionen eingeschränkt. Dabei muss ein geeigneter Schwellenwert von einem Experten vorgegeben werden und ist abhängig von den zu untersuchenden Daten.

7. Da die den statistischen Karten zugrundeliegenden fMRT-Daten häufig eine geringere räumliche Auflösung als die T1-gewichteten Anatomiedaten besitzen, haben diese Karten ebenfalls eine geringere räumliche Auflösung. Der Unterschied in der Auflösung der dieser Arbeit zugrundeliegenden Daten besteht darin, dass die Anatomiedaten eine Auflösung von $1 \times 1 \times 1$ mm und die t-Karten eine Auflösung von $3 \times 3 \times 3$ mm haben. Eine genaue Abbildung auf die Anatomie ist dadurch nicht ohne Interpolation möglich. Durch geeignete Verfahren ist eine Abschätzung erreichbar. Im Rahmen dieser Arbeit wird allerdings auf die Verwendung Abbildungsmethoden verzichtet, da eine solche Abschätzung die Originaldaten verändert und Daten erzeugt, die eigentlich nicht vorhanden sind.
8. Wie im Abschnitt 2.3 erwähnt, handelt es sich bei den gesuchten Arealen im Gehirn um in sich geschlossen Gebiete, wodurch der Einsatz von Segmentierungsverfahren sinnvoll ist. Aus der Bildverarbeitung sind Verfahren wie das Region-Growing und die Wasserscheidentransformation bekannt. Auch die bis hierher getroffenen Überlegungen lassen den Schluss zu, dass diese Methoden geeignet sein können, funktionelle Areale auf t-Karten zu finden.
9. Aus Erfahrungswerten ist bekannt (Zilles et al.), dass Regionen eine gewisse Mindestgröße haben. Weiterhin können Regionen nicht beliebig groß werden, da sie auf den Kortex beschränkt sind. Eine weitere Einschränkung des Suchraums wäre durch das Einbeziehen von Wissen über die Lage neuroanatomischer Strukturen möglich. Allerdings ist dies durch (7) erschwert. Auf die Größe einer Region kann recht einfach reagiert werden. Nach der Bestimmung aller Regionen können die Regionen entfernt oder anderen zugeordnet werden, die zu klein sind, wobei das Kriterium für zu kleine Regionen vom Anwender aus dem bisherigem Wissen über die funktionelle Organisation des Kortex definiert werden muss.
10. Die Grenzen zwischen den Regionen können fließend sein. Da angenommen wird, dass die Werte einer Region gaußverteilt sind, ist nicht eindeutig zu bestimmen, ob ein Voxel wirklich zu einer Region gehört. Durch diesen Umstand ist davon auszugehen, dass keine vollständige und völlig korrekte Zuordnung von Voxeln getroffen werden kann. Daher ist es unumgänglich, dass die durch die Algorithmen

bestimmten Ergebnisse durch Experten ausgewertet und auf ihre Plausibilität überprüft werden.

Durch die getroffenen Annahmen sind auch Rückschlüsse auf geeignete Maßzahlen und Kriterien zum Bestimmen der Zugehörigkeit eines Voxels zu einer Region möglich.

Folgende Maße und Kriterien sind dabei für die Berechnung relevant:

Größe $|r|$ einer Region r : $|r|$ ist die Anzahl an Voxeln, aus der eine Region besteht. Ist die Region kleiner als eine angegebene Mindestregionsgröße g_{\min} , ist sie nicht relevant und kann entfernt oder einer benachbarten größeren Region zugeordnet werden.

Mittelwert \bar{r} einer Region r : $\bar{r} = \frac{\sum_{i=1}^{|r|} v_i}{|r|}$ wobei v_i der statistische Wert eines Voxels aus r ist. Da laut (4) angenommen wird, dass Werte einer Region ähnlich sind, sollten sich Voxel, die einer Region zugeordnet werden, nicht zu stark von dem Mittelwert der Region unterscheiden.

Distanz d_s zum Regionszentrum: d_s ist der kürzester Pfad in 6er Nachbarschaft zwischen dem Regionszentrum und einem Voxel. Da laut (9) eine Region nicht beliebig groß sein kann und besonders an Regionsgrenzen nicht eindeutig bestimmt werden kann, zu welcher Region ein Voxel gehört, ist es nötig, den Abstand zu den nächsten Regionszentren zu beachten.

Variation benachbarter Voxel, Var: $Var = \frac{|v_i - v_j| * 100}{v_i}$. Die statistischen Werte der Voxel einer Region sind laut (2) gaußverteilt und fallen zum Rand einer Region hin ab, d.h. je weiter sie vom Regionszentrum entfernt sind, desto größer ist der Unterschied zwischen ihnen und dem Regionszentrum. Durch diesen Umstand kann der Mittelwert einer Region nicht alleine als Zuordnungskriterium dienen, da Voxel am Rand einer Region stärker vom Mittelwert der Region abweichen können als Voxel in der Nähe des Regionszentrums. Deswegen ist es nötig zusätzlich zu überprüfen, wie sich Voxel in lokaler Nachbarschaft unterscheiden.

Standardabweichung σ : $\sigma = \sqrt{\frac{1}{|r|} (\sum_{i=1}^{|r|} v_i^2) - \bar{r}^2}$ Die Standardabweichung in den statistischen Werten wird als Ähnlichkeitsmaß innerhalb einer Region verwendet, um zu bestimmen wie homogen die entstandene Region ist. Durch die Gaußverteilung der Werte ist allerdings damit zu rechnen, dass mit der Größe der Region auch die Standardabweichung

zunimmt. Weiterhin ist eine Region nicht nur durch ihre Homogenität charakterisiert. Daher dient dieses Maß nur als Anhaltspunkt für die Plausibilität einer Region.

Mit den genannten Annahmen und den vorgeschlagenen Maßzahlen ist es nun möglich, geeignete Algorithmen zu wählen. Im folgenden Abschnitt wird diese Wahl getroffen.

3.1.3. Algorithmenwahl

Wie schon im vorherigen Abschnitt genannt, lässt der Aufbau der Daten die Annahme zu, dass die aus der Bildverarbeitung bekannten Algorithmen des Region-Growing und der Wasserscheidentransformation geeignet sind, um das vorliegende Problem zu lösen.

Beim Region-Growing (Adams et al. 1994) handelt es sich um einen aus der Bildverarbeitung wohlbekannten Algorithmus zur Segmentierung. Eine Region in einem Bild wird durch bestimmte Homogenitätskriterien (z.B. Intensitätswerte der Bildpunkte) als zusammenhängend bestimmt. Der Algorithmus erhält einen Punkt aus der Region als Eingabe. Von diesem Saatpunkt aus werden alle benachbarten Bildpunkte verglichen. Entsprechen diese Bildpunkte dem vorher festgelegten Homogenitätskriterium, werden sie zur Region hinzugefügt, andernfalls nicht (Abbildung 5). Wurde ein Bildpunkt hinzugefügt, werden anschließend auch seine Nachbarn überprüft. Dieser Vorgang wird solange fortgesetzt, bis keine Punkte mehr der Region hinzugefügt wurden oder ein anderes vorher definiertes Abbruchkriterium zutrifft (Regionsgröße, maximaler Abstand zum Saatpunkt).

Die Wasserscheidentransformation (Lin et al. 2003, Haris et al. 1998, Vincent et al. 1991, Wegner et al. 1997) ist ein alternativer Ansatz zur Segmentierung. Hierbei interpretiert

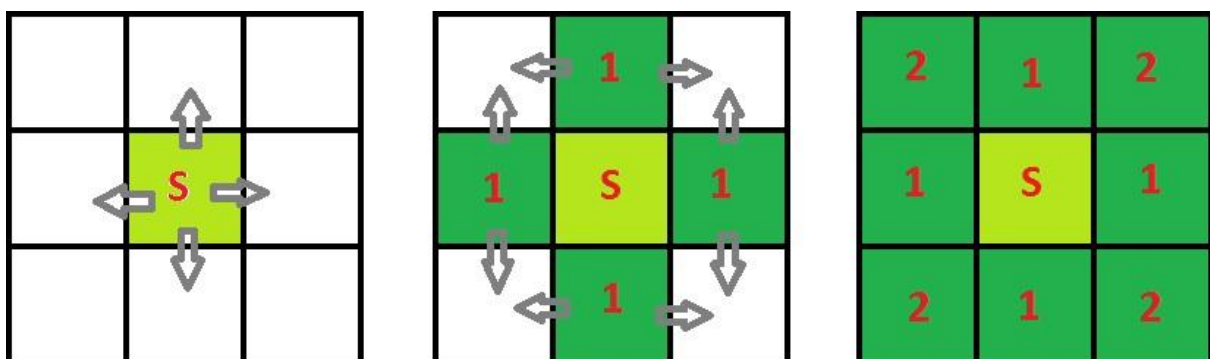


Abbildung 5: Die Abbildung zeigt von links nach rechts: Die Initialisierung des Region-Growings mit einem Saatpunkt S. Von diesem aus werden die ersten vier Nachbarn (mit 1 markiert) überprüft und der Region hinzugefügt. Von jedem dieser (mit 1 markierten) Nachbarn werden die Nachbarn (mit 2 markiert) überprüft und ebenfalls zur Region hinzugefügt.

man das zu segmentierende Bild als ein Wertgebirge, was bedeutet dass jeder Wert im Bild als Erhebung an dem Punkt betrachtet wird. Es wird angenommen, dass Täler homogene Regionen und Berggrate starke, kontrastreiche Konturen, also die Regionengrenzen repräsentieren. Im Algorithmus werden nun, beginnend mit den niedrigsten Werten, initiale Regionen bestimmt. Anschließend wird die Schwelle in Richtung der höheren Werte verschoben. Dabei werden Bildpunkte, die an eine bereits bestehende Region angrenzen, dieser Region hinzugefügt. Gibt es Bildpunkte, die an keine Region angrenzen, bilden diese eine neue Region. Bildpunkte, die an mehr als eine Region angrenzen werden als Wasserscheiden bezeichnet und stellen nach Abschluss des Algorithmus die Grenzen zwischen den Regionen dar (Abbildung 6).

In der Regel wird die Wasserscheidentransformation allerdings nicht direkt auf einem Bild durchgeführt, sondern das Bild vorher mittels Kantenfiltern in ein Gradientenbild umgewandelt. Auf einem Gradientenbild sind die Kanten, also Regionsgrenzen, durch hohe Werte repräsentiert.

Beide Verfahren liefern in sich geschlossene Regionen. Da die lokalen Maxima laut (2) im Abschnitt 3.1.2 als Regionszentren angesehen werden, können diese als Saatpunkte für das Region-Growing dienen.

Die Verwendung des Region-Growing soll dazu dienen, die in Abschnitt 3.1.2 getroffenen Annahmen zu überprüfen und herauszufinden, ob mit ihnen mögliche funktionell getrennte Areale auf den t-Karten lokalisiert werden können. Dabei ist zu beachten, dass für das Region-Growing verschiedene Parameter durch den Anwender vorgegeben werden müssen. Solche Parameter können zum Beispiel das gewünschte Ähnlichkeitsmaß beim Hinzufügen eines Voxels (zum Beispiel der Unterschied benachbarter Voxel Δ oder die Abweichung eines Voxels vom Mittelwert \bar{r} einer Region) oder die Schrittweite des Algorithmus (der maximale Abstand zum Regionszentrum d_s) sein.

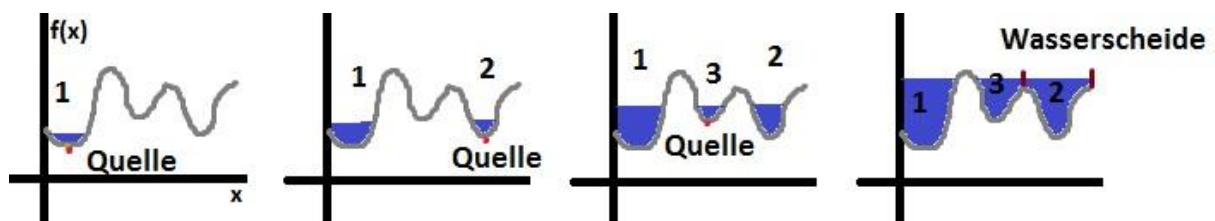


Abbildung 6: Die Abbildung zeigt von links nach rechts: Die Initialisierung des Region-Growings mit einem Saatpunkt S. Von diesem aus werden die ersten vier Nachbarn (mit 1 markiert) überprüft und der Region hinzugefügt. Von jedem dieser (mit 1 markierten) Nachbarn werden die Nachbarn (mit 2 markiert) überprüft und ebenfalls zur Region hinzugefügt.

Die Abhängigkeit von Parametern führt dazu, dass Ergebnisse schlechter werden können, wenn ungünstige Parameterwerte gewählt wurden. Weiterhin ist es dadurch möglich, dass nicht alle Voxel im Datensatz Regionen zugeordnet werden. Es kann passieren, dass Voxel am Rand von Regionen die Kriterien nicht erfüllen und somit keiner Region zugeordnet werden. Durch Rauschen bedingte Ausreißer innerhalb einer Region (Voxel deren Werte sich stark von den Werten der umliegenden Voxel unterscheiden) könnten eventuell auch zu einer Verschlechterung der Ergebnisse führen.

Um diese Parameterabhängigkeit zu umgehen und eine möglichst vollständige Segmentierung der Daten zu erreichen, ist die Verwendung eines anderen Verfahrens nötig. Wie bereits erwähnt, bietet sich die Wasserscheidentransformation an. Betrachtet man die lokalen Maxima in den Daten als Quellen, kann von ihnen aus eine Flutung erfolgen.

Da davon ausgegangen wird, dass die Daten verrauscht sind und die Wasserscheidentransformation empfindlich gegenüber Rauschen ist, ist davon auszugehen, dass eine Übersegmentierung die Folge ist. Durch eine weitere Zusammenfassung der initial durch die Wasserscheidentransformation erzeugten Regionen kann der Übersegmentierung jedoch entgegengewirkt werden. Es bietet sich auch die Durchführung der Wasserscheidentransformation auf Graphenebene an, da es mit Hilfe einer Graph-/Baumstruktur leicht möglich ist die Wasserscheidentransformation hierarchisch durchzuführen. Zum einen ist die Wasserscheidentransformation wie bereits erwähnt parameterfrei und zum anderen bietet eine Multiskalenstrategie auf einem Graphen den Vorteil, dass die verschiedenen Zusammenfassungsstufen der Regionen innerhalb einer Datenstruktur gespeichert werden können, was die Verwaltung der Daten einfacher macht. Eine wichtige Anforderung an eine solche Graph-/Baumstruktur zum Verwalten der Segmentierungsergebnisse der statistischen Karten ist, dass sie neben den erzeugten Regionen auch statistische Maßzahlen wie den in Abschnitt 3.1.2 genannte Mittelwert oder die Standardabweichung speichert, um die Qualität der Regionen besser überprüfen zu können. Weiterhin sollte die Datenstruktur die Möglichkeit bieten, die Beziehungen zwischen den Regionen (Nachbarschaft, welche Regionen werden zu größeren Regionen zusammengefasst) abzubilden. Diese Anforderung ist besonders im Hinblick auf eine spätere Interaktion mit den Daten wichtig.

3.2. Verwendete Verfahren zur Segmentierung

In diesem Abschnitt werden zwei Varianten des Region-Growing erläutert. Die erste Variante (RG1) dient zum Überprüfen, ob eine Segmentierung auf statistischen Karten sinnvoll ist. Die zweite Variante (RG2) ist eine Verfeinerung von RG1, um die Qualität der Ergebnisse zu verbessern. Weiterhin werden die Wasserscheidentransformation (WST) mit verschiedenen Nachbarschaftsbeziehungen (6er und 26er Nachbarschaft) und die final aus den Ergebnissen der Wasserscheidentransformation entstandene Datenstruktur des Multi-Scale-Region-Tree (MSRTree) zum Erzeugen und Speichern von gefundenen Regionen erläutert.

3.2.1. Region-Growing auf statistischen Karten

Aus der Problemanalyse ist bekannt, dass die lokalen Maxima in den t-Karten als Saatpunkte für Region-Growing-Verfahren dienen können. Daraus folgt, dass das Bestimmen der lokalen Maxima die Grundlage für ein Region-Growing-Verfahren ist. Hierzu dient Algorithmus 1:

Bestimmung von lokalen Maxima	
Eingabe	Menge M an auf Maxima zu untersuchenden Voxel
1	$m_1 :=$ Bestimme alle lokalen Maxima in X-Richtung aus M .
2	$m_2 :=$ Bestimme alle lokalen Maxima in Y-Richtung aus m_1 .
3	$m_3 :=$ Bestimme alle lokalen Maxima in Z-Richtung aus m_2 .
4	$M_{Max} :=$ Bestimme alle lokalen Maxima, die in einer 26er Nachbarschaft die größten Maxima aus m_3 sind.
Ausgabe	Menge M_{Max} an lokalen Maxima in 26er Nachbarschaft.

Algorithmus 1

Die Voxelmenge M wird schrittweise reduziert, um die Laufzeit des Algorithmus zu beschleunigen. Hierzu werden zuerst alle Voxel aus M bestimmt, deren statistischen Werte in X-Richtung maximal sind, also die Bedingung $x_{i-1} < x_i > x_{i+1}$ erfüllen. Diese Voxel bilden die Menge m_1 . Alle Voxel in m_1 werden auf die Bedingung $y_{i-1} < y_i > y_{i+1}$ überprüft. Die daraus entstandene Menge m_2 enthält nun nur noch Voxel, die in X- und Y-Richtung maximal sind. Aus m_2 entsteht m_3 , indem die in m_2 enthaltenen Voxel auf die Bedingung $z_{i-1} < z_i > z_{i+1}$ überprüft werden. Die Menge m_3 enthält nun nur noch solche Voxel, die in einer 6er Nachbarschaft maximal sind. Bei der Überprüfung der Voxel verweisen x , y und z auf die x -, y - beziehungsweise z -Koordinate der Voxel.

In Schritt 4 des Algorithmus wird nun für jedes Voxel v_1 aus m_3 überprüft, ob ein weiteres Voxel v_2 aus m_3 in der 26er Nachbarschaft liegt. Ist dies der Fall, wird v_1 der Menge M_{Max} zugewiesen wenn $v_1 > v_2$, ansonsten wird v_2 M_{Max} zugewiesen. Die so entstandene Menge M_{Max} aus lokalen Maxima wird nun als Eingabe für das Region-Growing-Verfahren verwendet.

RG1 dient dazu, zu bestimmen, ob eine sinnvolle Segmentierung der t-Karte in getrennte Hirnregionen möglich ist. Dazu werden folgende Schritte nach dem Bestimmen der lokalen Maxima durchgeführt:

Region-Growing - Variante 1	
Eingabe	Menge an lokalen Maxima M_{Max}
1	$m_a :=$ alle Maxima in M_{Max} die größer Schwellenwert s sind.
2	Überprüfen für alle Maxima max_i , ob weitere Maxima max_j in einem Umkreis u_k liegen.
3	Wenn ja, dann vergleiche max_i mit max_j .
4	Ist $max_i \in [max_j - var, max_j + var]$, dann entferne max_i wenn max_j größer, ansonsten max_j aus m_a .
5	$m_b :=$ Wende Regionenschwellenwert s_r in einer kleinen lokalen Umgebung u_r bei alle lokalen Maxima aus m_a an.
6	Starte Region-Growing mit lokalen Maxima m_b als Saatpunkte.
7	Für jeden Saatpunkt ps_i aus m_b
8	Erzeuge Queue q und füge ps_i hinzu.
9	Solange q nicht leer
10	Vergleiche Voxel v aus q mit allen Nachbarn nb_j in 6er Nachbarschaft.
11	Ist $nb_j \in [v - var_{rg}, v + var_{rg}]$, füge nb_j in q ein.
12	Füge v Region r_i hinzu.
13	Für jede Region r_i
14	Ist r_i kleiner Mindestregionsgröße g_{min} , entferne Regionssaatpunkt ps_i aus m_b .
15	Wurde ein Regionssaatpunkt aus m_b entfernt, gehe zu 7.
16	Speichern aller erzeugten Regionen r .
Ausgabe	.voi Datei, die alle erzeugten Regionen r enthält.

Algorithmus 2

Der in Schritt 1 angewendete Schwellenwert s ist ein vom Anwender vorgegebener Schwellenwert. Er gibt an, ab welchem Wert ein Voxel v als signifikant betrachtet werden kann. Die Menge m_a , die zur weiteren Bearbeitung verwendet wird, enthält nur solche v , für die $v > s$ gilt.

Schritt 2 dient zur Behandlung des Rauschens. Da davon ausgegangen wird, dass die Daten verrauscht sind, kann auch davon ausgegangen werden, dass einige Maxima durch das

Rauschen entstanden sind. Die Umgebung u_k gibt alle Voxel an, die für max_i auf weitere Maxima überprüft werden soll. In diesem Algorithmus wurde k mit 2 festgelegt, was bedeutet, dass alle Voxel, die in der 6er Nachbarschaft in 2 Schritten erreicht werden können, der zu untersuchenden Umgebung u_2 angehören. Für alle gefundenen max_j wird überprüft, ob $max_i \in [max_j - var, max_j + var]$ ist. Wobei var eine vom Anwender angegebene Varianz in Prozent darstellt, um die ein solches Maxima abweichen darf. Maxima, die nicht innerhalb dieser Varianz liegen, werden nicht überprüft. Liegt max_j innerhalb der Varianz, so wird überprüft, ob $max_i > max_j$. Wenn dies zutrifft, wird max_j aus m_a entfernt, andernfalls wird max_i aus m_a entfernt. Der Grund, warum k so klein gewählt wurde, ist, um zu verhindern, dass mögliche Regionszentren, die dicht beieinander liegen, durch eine Rauschbehandlung entfernt werden. Maxima, die nicht innerhalb der Varianz liegen, werden nicht entfernt, da davon ausgegangen wird, dass der Unterschied zwischen max_i und max_j in diesem Fall nicht durch Rauschen zu erklären ist.

Der Regionenschwellenwert s_r dient dazu Maxima zu entfernen, um die sich keine weiteren signifikanten Voxel v ($v > s$) befinden. Solche Maxima entstehen entweder durch Messfehler oder deuten auf unbedeutend kleine Regionen hin. Aus diesen Gründen können sie aus der Liste der relevanten Maxima entfernt werden. Für die Überprüfung wurde r auf 3 festgelegt, was bedeutet, dass alle Voxel in einer 6er Nachbarschaft, die in 3 Schritten erreicht werden können, der Umgebung u_3 angehören. Mit der folgenden Formel wird die Überprüfung des Regionenschwellenwerts durchgeführt:

$$\frac{\sum_{i=1}^N v_i}{N} > s_r, \quad (2)$$

N ist die Anzahl der Voxel in der Umgebung u_r und s_r wird aus s ermittelt:

$$s_r = 0.75 * s. \quad (3)$$

Alle Maxima, die Ungleichung (2) erfüllen, werden der Menge m_b hinzugefügt. Diese Menge enthält nun alle lokalen Maxima, die als Saatpunkte für das Region-Growing verwendet werden sollen.

Für jeden übergebenen Saatpunkt ps_i aus m_b wird das Region-Growing durchgeführt. Dabei wird ein Voxel v der Region r_i hinzugefügt, wenn er signifikant ($v > s$) ist und er sich nicht mehr als eine gegebene Varianz var_{rg} von seinen Nachbarn nb_j unterscheidet:

$$v \in [nb_j - var_{rg}, nb_j + var_{rg}]. \quad (4)$$

Die maximale Varianz var_{rg} wird hierbei vom Nutzer angegeben, sie bestimmt, wie homogen die Regionen werden. Je größer var_{rg} ($\text{var}_{\text{rg}} > 0.7$) ist, desto inhomogener wird die Region und je kleiner var_{rg} ($\text{var}_{\text{rg}} < 0.3$) ausfällt, desto homogener wird die Region. Bei kleinen var_{rg} werden die Regionen allerdings auch kleiner, da weniger Voxel die Bedingung erfüllen. Die Werte 0.3 und 0.7 wurden dabei experimentell bestimmt.

Nach dem Bestimmen der Regionen werden solche Regionen entfernt, die nicht die geforderte Mindestgröße g_{min} besitzen, also für die gilt:

$$|r_i| < g_{\text{min}}, \quad (5)$$

hierbei ist $|r_i|$ die Anzahl Voxel der Region r_i . Die Minimale Regionsgröße wird, wie die Varianzen var und var_{rg} , durch den Anwender angegeben. Aus Erfahrungswerten ist ungefähr bekannt, welche Größe mögliche Regionen im Gehirn haben. Mit der Menge an übrigen Saatpunkten m_b wird das Region-Growing mit Schritt 7 erneut durchgeführt. Damit werden Voxel, die vorher zu kleinen Regionen (Ungleichung 5) gehörten, den im erneuten Durchgang gebildeten Regionen zugeordnet. Wurden keine Saatpunkte aus m_b entfernt endet das Region-Growing.

Bei den Ergebnissen von RG1 ist zu erwarten, dass, je nach Parameterwahl, kleine Regionen ($|r_i| = g_{\text{min}}$) entstehen (wenn die Varianzen niedrig gewählt werden). Weiterhin kann das Entfernen von Maxima in Schritt 2 unter Umständen dazu führen, dass wichtige Saatpunkte entfernt werden. Deswegen ist es sinnvoll die Vorverarbeitung weniger strikt zu gestalten. Zu diesem Zweck wurde eine zweite Version des Region-Growing entwickelt.

3.2.2. Region-Growing Variante 2

Das Region-Growing Verfahren RG2 basiert auf RG1 und wurde in folgenden Punkten abgeändert: Die Vorverarbeitung wurde darauf reduziert, dass aus dem zu bearbeitenden Datensatz alle Voxel entfernt werden, deren Werte kleiner als der gegebene Schwellenwert s sind. Auf dem so reduzierten Datensatz werden die lokalen Maxima wie in Algorithmus 1 beschrieben bestimmt, wodurch die Menge M_{Max} an zu untersuchenden lokalen Maxima entsteht. M_{Max} dient als Eingabe für das Region-Growing wie in Algorithmus 2 Schritte 6-12 beschrieben. Eine Filterung der Maxima zur Rauschbehandlung wird im

Gegensatz zu RG1 nicht durchgeführt. Die Rauschbehandlung erfolgt als Nachverarbeitungsschritt.

Auf Basis der gegebenen Saatpunkte aus M_{Max} wird eine Distanzkarte erstellt, die für jedes Voxel im Datensatz die Distanz zum nächsten Maximum speichert. Diese Distanzkarte wird als Kriterium zur Regionszuordnung einbezogen. Das Längenmaß ist der kürzeste Pfad in 6er Nachbarschaft von einem Voxel v zu dem ihm nächsten Saatpunkt ps_i . Wenn v zu mehr als einer Region zugeordnet werden kann, wird es der Regionen zugeordnet, für die gilt, dass ps_i der v am nächsten liegende Saatpunkt ist.

Zusätzlich zu dem in RG1 verwendeten Kriterium für die Regionszuordnung eines Voxels (Bedingung 4) wird in RG2 der Mittelwert m einer Region miteinbezogen. Ein Voxel v , das einer Region r_i hinzugefügt werden soll, muss die Bedingung

$$\frac{v + \sum_{i=1}^{|r_i|} v_i}{|r_i|+1} - var_{rg} \leq m \leq \frac{v + \sum_{i=1}^{|r_i|} v_i}{|r_i|+1} + var_{rg} \quad (6)$$

mit

$$m = \frac{v + \sum_{i=1}^{|r_i|} v_i}{|r_i|+1} \quad (7)$$

erfüllen. Hierbei ist $|r_i|$ die Anzahl der Voxel in der Region, bevor v der Region hinzugefügt wurde.

Es ist anzunehmen, dass viele kleine Regionen ($|r_i| < g_{\text{min}}$) entstehen, da im Gegensatz zu RG1 keine Maxima vor dem Start des Region-Growing entfernt werden. Deshalb werden alle Regionen entfernt, die kleiner als g_{min} (Algorithmus 2, Schritte 13 und 14) sind. Anschließend wird die Distanzkarte neu bestimmt und das Region-Growing mit der nun reduzierten Menge an lokalen Maxima M_{Max} erneut durchgeführt. Dieser Schritt wird solange wiederholt, bis keine Regionen mehr entfernt wurden.

3.2.3. Wasserscheidentransformation

Aus der Problemanalyse geht hervor, dass die gesuchten Regionen durch Täler voneinander getrennt sind. Aus diesem Grund bietet sich die Wasserscheidentransformation (WST) zur Bestimmung von Regionen an. Ein Vorteil der WST gegenüber dem Region-Growing ist, dass sie parameterfrei ist. Weiterhin wird die Liste der lokalen Maxima nicht als Eingabe benötigt.

Um die Wasserscheidentransformation auf den vorhandenen Daten durchzuführen, müssen folgende Dinge beachtet werden:

- Der zugrunde liegende Datensatz ist ein 3D-Datensatz, daraus folgt, dass die WST in 3D implementiert wird.
- In der Regel ist es üblich, die WST auf einem Gradientenbild durchzuführen. Die Form der gegebenen Daten lässt allerdings den Schluss zu, dass dies hier nicht nötig ist. Tatsächlich würde das Erzeugen eines Gradientenbildes die Qualität der Daten negativ beeinflussen und die gesuchten absoluten Unterschiede zwischen den einzelnen Voxeln in relative Unterschiede umwandeln.

Die Implementation der WST auf 3D erfolgt durch den Einsatz einer 6er beziehungsweise 26er Nachbarschaft.

Durch die Form der vorliegenden Daten ist als Vorverarbeitungsschritt für die Wasserscheidentransformation nur das Entfernen aller nicht relevanten Daten, also der negativen Werte und der Werte unterhalb des gegebenen Schwellenwert s nötig:

$$v_i = \begin{cases} -v_i, & \text{wenn } v_i > s \\ 0, & \text{sonst} \end{cases} . \quad (8)$$

Hierbei bezeichnet v_i den statistischen Wert eines Voxels aus dem Datensatz. Alle Voxel, die nach dieser Operation ungleich null sind, bilden die Menge an zu bearbeitenden Voxeln V_s . Der Schwellenwert s ist der unter 3.2.1 und 3.2.2 bereits erwähnte Schwellenwert. Anschließend kann der invertierte Datensatz als Eingabe für die WST dienen. Der Algorithmus kann dann als Flutungsalgorithmus wie folgt formuliert werden:

Wasserscheidentransformation	
Eingabe	Menge an zu bearbeitenden Voxeln V_s
1	Bilde eine Referenztabelle R_t für alle Voxel aus V_s .
2	Sortiere R_t nach Werten aus V_s aufsteigend.
3	Für jeden Voxel v_i aus R_t
4	Flute alle v_i mit aktuellem Wert.
5	Hat v_i Nachbarn einer anderen Region r_j ?
6	Voxel r_j hinzufügen.
7	Hat v_i nur Nachbarn, die keiner Region zugeordnet sind?
8	Neue Region r_{j+1} bilden.
9	Alle benachbarten Voxel, die denselben Wert wie v_i haben, r_{j+1} hinzufügen.
10	Hinzugefügte Voxel als bearbeitet markieren und überspringen.
11	Hat v_i Nachbarn aus verschiedenen Regionen?
12	v_i als Wasserscheidenvoxel v_{ws} markieren und Region r_{ws} merken, zu der er am ähnlichsten ist.
13	Nächsten Voxel v_{i+1} wählen.
14	Wenn alle Voxel aus R_t Regionen zugeordnet sind, die Wasserscheidenvoxel v_{ws} den ihnen jeweils ähnlichsten Regionen r_{ws} zuweisen.
15	Regionskarte K erzeugen und speichern.
Ausgabe	.voi Datei, die aller erzeugten Regionen r enthält.

Algorithmus 3

Die nach Schritt 2 vorliegende nach statistischen Werten der Voxel aufsteigend sortierte Referenztabelle R_t dient dazu, den Algorithmus zu beschleunigen. Dabei besteht die Tabelle aus zwei Einträgen pro Voxel. Der erste Eintrag ist der statistische Wert des Voxels und der zweite Eintrag ist eine Referenz auf die Position (x,y,z-Koordinate) im Datensatz. Über die Referenz kann direkt auf ein Voxel im Datensatz zugegriffen werden. Dadurch erübrigt sich ein wiederholtes Durchsuchen des Datensatzes auf den nächsten zu bearbeitenden Voxel.

Die Flutung der Voxel erfolgt inkrementell, beginnend bei dem niedrigsten bis zu den höchsten Werten (entsprechend der Sortierung von R_t). Die gefundenen Wasserscheiden entsprechen den aus 3.1.2 bekannten Grenzen zwischen den Regionen. Da eine vollständige Zuordnung aller Voxel zu jeweils einer Region gewünscht ist, werden nach der Flutung alle Wasserscheidenvoxel v_{wst} ihrer jeweils ähnlichsten Region r_{wst} hinzugefügt. Dabei wurde während der Flutung (Schritt 11/12 Algorithmus 3) mittels der Bedingung

$$r_{wst} = f(R, v_{wst}) \quad (9)$$

mit R als der Menge der umliegenden Regionen und

$$f(R, v_{wst}) = \min_{0 < i \leq |R|} \left(\frac{v_{wst} + \sum_{k=1}^{|r_i|} v_k}{|r_i| + 1} \right) \quad (10)$$

ermittelt, zu welcher umgebenden Region v_{wst} ähnlicher ist.

Die in dieser Arbeit implementierte Variante der Wasserscheidentransformation arbeitet sowohl in einer 26er als auch in einer 6er Nachbarschaft. Die erste Implementierung erfolgte mit einer 26er Nachbarschaft, um möglichst alle benachbarten Voxel zu beachten. Es ist zu erwarten, dass bei dieser Nachbarschaft entstehende Regionen etwas größer und weniger zahlreich als bei der 6er Nachbarschaft sind. Der Kortex ist sehr dünn und Areale breiten sich in der Regel orthogonal auf ihm aus. Die eingeschränkte Nachbarschaft dient dazu, die bekannten Informationen (Form, Lage der Regionen, siehe 3.1.2) aus der Anatomie besser zu berücksichtigen.

Im Gegensatz zu den beiden Region-Growing-Verfahren wird bei der implementierten WST nicht auf die minimale Regionsgröße g_{min} geachtet. Es werden alle Regionen gespeichert, unabhängig von ihrer Größe.

Aus dem Nichtbeachten von g_{min} und der bekannten Empfindlichkeit der WST für lokale Variation ist anzunehmen, dass es sich bei K um eine Übersegmentierung handelt.

3.2.4. Multi-Scale-Region-Tree

Um das Problem der Übersegmentierung zu lösen und eine bessere Verwaltung der Ergebnisse zu ermöglichen, wird der Multi-Scale-Region-Tree (MSRTree) eingeführt. Hierbei handelt es sich um eine Baumstruktur, auf der eine hierarchische Wasserscheidentransformation durchgeführt wird.

Der MSRTree besteht aus Knoten und Kanten, wobei Knoten den während der Wasserscheidentransformation entstandenen Regionen entsprechen und Kanten einer Bauebene die Nachbarschaftsbeziehungen zwischen den Regionen darstellen. Kanten zwischen den Bauebeneben geben an, welche Regionen zur nächstgrößeren Region zusammengefasst wurden (Abbildung 7).

Neben der Zuordnung von Voxeln zu Regionen speichert der MSRTree zusätzlich statistische Werte und Maßzahlen für jeden Knoten. Folgende Informationen werden dabei pro Knoten gespeichert:

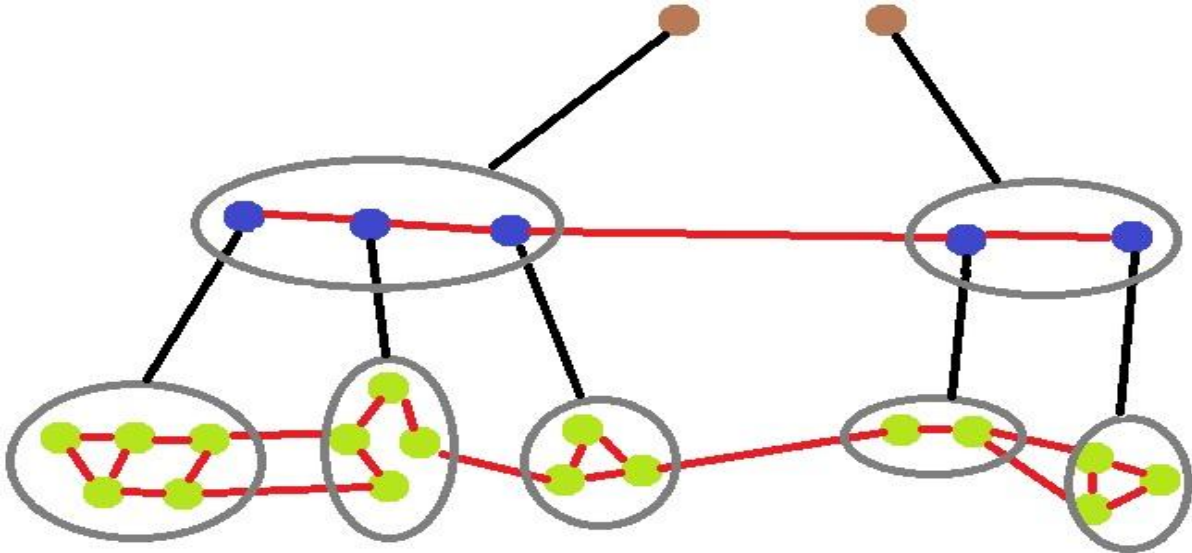


Abbildung 7: Die Abbildung zeigt eine schematische Darstellung vom Aufbau des MSRTree. Die grünen Knoten stellen Voxel aus dem Originaldatensatz dar (Blattknoten). Die blauen Knoten (Astknoten) stellen Regionen dar, die aus den grünen Knoten gebildet wurden (graue Kreise zeigen die zusammengefassten Regionen an). Die braunen Knoten (Stammknoten) stellen die Zusammenfassung der blauen Knoten dar. Die roten Kanten stellen die Nachbarschaftsbeziehungen zwischen Knoten desselben Typs dar. Die Schwarzen Kanten stellen die Zugehörigkeit zu den zusammenfassenden Knoten der nächsten Ebene dar.

- Die Ebene n , auf der sich der Knoten befindet.
- Alle Knoten der Ebene $n-1$, die einer Region (einem Knoten) der Ebene n angehören.
- Die Anzahl der Knoten, der Ebene $n-1$, die diesen Knoten bilden.
- Den Mittelwert \bar{r} der Region.
- Die Standardabweichung σ der Region.
- Einen Verweis auf den Knoten der Ebene $n+1$, falls es eine nächsthöhere Zusammenfassungsstufe gibt.
- Ein Verweis auf alle Knoten der untersten Ebene, die am Rand der Region liegen.
- Eine Liste von Nachbarknoten derselben Ebene.

Durch das Speichern der Referenzen zwischen Knoten ist es möglich, von jedem Punkt im Baum aus entsprechende Knoten auf höheren und tieferen Bauebene zu finden. Weiterhin können für jeden Knoten die Nachbarknoten ermittelt werden. Dabei besteht der MSRTree aus drei verschiedenen Arten von Knoten, die nachfolgend genauer erläutert werden.

Der erste Knotentyp ist der **Blattknoten**. Blattknoten bilden die unterste Ebene des Baums. Jeder Blattknoten entspricht einem relevanten Voxel v ($v > s$) aus dem

Originaldatensatz. Das Erzeugen der Blattknoten ist der Initialisierungsschritt für den MSRTree. Hierbei werden wie bei der WST alle Werte aus dem Originaldatensatz entfernt, die unterhalb des gegebenen Schwellenwerts s liegen:

$$b = \begin{cases} 1, & \text{wenn } v > s \\ 0, & \text{sonst} \end{cases}, \quad (11)$$

hierbei wird aus jedem der Voxel für die $b = 1$ gilt ein Blattknoten erzeugt.

Im Blattknoten wird zusätzlich eine Liste von benachbarten Blattknoten angelegt. Diese Nachbarschaftsliste entsteht auf Basis einer 6er Nachbarschaft im Originaldatensatz, wobei nur relevante Voxel ($b=1$) als Nachbarn gespeichert werden. Durch diesen Umstand können mehrere untereinander nicht verbundene Blattknotengraphen entstehen.

Der nächste Knotentyp ist der **Astknoten**. Astknoten entstehen bei der ersten Wasserscheidentransformation. Sie enthalten nur Blattknoten. Erzeugt werden sie, indem die Wasserscheidentransformation auf die Blattknoten angewendet wird. Alle Blattknoten einer Region werden nun einem Astknoten zugeordnet. Anhand der Nachbarschaftslisten der im Astknoten enthaltenen Blattknoten kann anschließend die Nachbarschaftsliste des Astknotens bestimmt werden. Ist diese bestimmt, kann auf die Astknoten erneut eine Wasserscheidentransformation angewendet werden, wodurch die Stammknoten entstehen.

Stammknoten können entweder Ast- oder weitere Stammknoten enthalten. Stammknoten einer Ebene werden wie Astknoten erzeugt, indem auf die Knoten der unteren Ebene die Wasserscheidentransformation angewendet wird und die sich ergebenden Regionen als neue Stammknoten gespeichert werden.

Die verwendete Wasserscheidentransformation arbeitet wie die zuvor unter 3.2.3 beschriebene WST. Der Unterschied zwischen beiden Verfahren besteht darin, dass die im MSRTree verwendete Variante auf Graphen angepasst wurde und als Ergebnis keine Regionskarte, sondern eine Knotenliste K_1 liefert. K_1 entspricht immer einer neuen Ebene im Baum. Weiterhin entfernt der MSRTree alle Knoten aus dem Baum, die keine Nachbarn besitzen und kleiner als die Mindestregionsgröße g_{\min} sind. Solche Knoten entsprechen Regionen, die nicht weiter zusammengefasst werden können und nicht der von Anwender geforderten Mindestregionsgröße entsprechen. Somit können sie aus dem MSRTree entfernt werden.

3.2.5.Ausgabe

Alle genannten Verfahren speichern die von ihnen erzeugten Regionslisten als „.voi“ Datei. Bei der „.voi“ Datei handelt es sich um ein von BrainVoyager™ vorgegebenes Format um ROIs zu speichern. Hierbei ist in der Datei die Anzahl der enthaltenen ROIs gespeichert, deren Namen, Farbe und alle Koordinaten der Voxel, die der ROI zugeordnet sind. Jede gespeicherte ROI entspricht einer der gefundenen Regionen. Das folgende Beispiel demonstriert den Aufbau einer solchen „.voi“ Datei:

```
NrOfVOIs: 75
```

```
NameOfVOI: Nr_1_s_7  
ColorOfVOI: 0 255 0
```

```
NrOfVoxels: 189  
-58 -11 9  
-58 -11 10  
-58 -11 11  
-58 -10 9  
-58 -10 10  
-58 -10 11  
-58 -9 9  
-58 -9 10
```

In der ersten Zeile wird die Anzahl der Regionen angegeben. Die zweite Zeile enthält den Namen der Region und die dritte Zeile enthält die Farbe als RGB-Wert kodiert. Anschließend folgt die Anzahl der Voxel, aus der die Region besteht, gefolgt von den Koordinaten der Voxel. Wichtig zu beachten ist, dass BrainVoyager™ die Darstellung von ROIs auf 255 Stück beschränkt, 255 ist auch die maximale Anzahl an Regionen, die pro „.voi“ Datei gespeichert werden können. Werden mehr Regionen gefunden, müssen entsprechend mehrere „.voi“ Dateien erstellt werden. Nähere Informationen zu diesem Datenformat und seiner Verwendung können der BrainVoyager™-Hilfe entnommen werden.

3.3. Implementierung

Für die Umsetzung der Implementierung der genannten Verfahren wurde Microsoft Visual Studio 2008 mit der Programmiersprache C++ genutzt.

Das genutzte Dateninterface wurde bereitgestellt. Mit der Nutzung des Interfaces und der Verwendung von C++ als Programmiersprache soll eine spätere Einbindung des im Rahmen dieser Arbeit entwickelten Programms in andere am Leibniz-Institut für Neurobiologie Magdeburg entwickelte Tools erleichtert werden. Weiterhin senkt die Nutzung des Dateninterfaces den späteren Wartungsaufwand, da eventuelle Formatänderungen im eingelesenen Format nicht innerhalb des Codes für das Tool geändert werden müssen.

Aus Übersichtsgründen wird im Folgenden nur ein grober Überblick über die Implementation des Tools gegeben. Detailliertere Informationen können der dem Tool beigefügten Quellcodedokumentation entnommen werden. Ebenfalls ist dem Tool eine Anleitung beigefügt, die die Verwendung erklärt und auf die benötigten Ein- und Ausgabeformate eingeht.

Für das Einlesen der statistischen Karten wird die Klasse *DataF* genutzt. Sie enthält eine Anbindung an das oben genannte Dateninterface. *DataF* liest mittels des Dateninterfaces die Dateien ein und extrahiert die für das Tool relevanten Informationen. Anschließend werden die Daten in der Klasse *ProcessingData* gespeichert.

Die Klasse *voiHandler* dient zum Speichern der Ergebnisse des Tools. Ihr werden die durch den Algorithmus erzeugten Regionskarten übergeben. Anschließend wandelt sie diese in „.voi“ Dateien um und speichert sie. Zurzeit arbeitet diese Klasse noch mit einer eigenen Methode zum Erzeugen der „.voi“ Dateien und nicht mit den Methoden des Dateninterfaces. Der Grund hierfür ist, dass zum Zeitpunkt der Erstellung der Klasse im Dateninterface noch keine Methoden zum Speichern der „.voi“ Dateien vorhanden war.

Die Klasse *ProcessingData* speichert alle für die Bearbeitung benötigten Daten und Informationen. Über diese Klasse können die für die Algorithmen benötigten Daten in dem für sie benötigten Datenformat ausgelesen werden. Die t-Karte wird dabei als Float Array gespeichert, da es sich bei den Daten um Fließkommazahlen handelt.

Für die Umsetzung der Verfahren RG1 und RG2 dienen die Klassen *Maxima* und *RegionTest*. Der Klasse *Maxima* wird die t-Karte und die benötigten Parameter wie

Schwellenwert s , Variation zwischen den Maxima var und maximale Distanz zum regionenbildenden Maximum übergeben. Anschließend berechnet diese Klasse, wie im Kapitel 3.2.1 beschrieben, die lokalen Maxima. Mit diesem Ergebnis und den übergebenen Parametern ruft sie die Klasse *RegionTest* auf. Diese ermittelt mittels des in 3.2.1 aufgeführten Region-Growing Algorithmus die Einteilung der t-Karte in Regionen. Als Ergebnis liefert *RegionTest* ein Integer Array, welches für jedes Voxel aus der t-Karte die jeweils zugeordnete Region speichert. Mittels der vorher beschriebenen Klasse *voiHandler* kann die Regionskarte als „.voi“ Datei gespeichert werden.

Um die in 3.2.2 genannten Unterschiede der Algorithmen RG1 und RG2 umzusetzen, werden für RG2 die Klassen *Maxima2* und *RegionTest2* genutzt. Diese Klassen sind von *Maxima* oder *RegionTest* abgeleitet. Dadurch mussten sie nur um die unter 3.2.2 beschriebenen Modifikationen erweitert werden. So konnte ein erneutes Implementieren der Einlese- und Speichermethoden vermieden werden.

Die Klasse *WST* dient zur Umsetzung der Wasserscheidentransformation. Sie wird mit der Klasse *ProcessingData* und einem Parameter zur Auswahl, ob die 6er oder die 26er Nachbarschaft verwendet werden soll, aufgerufen. Anschließend führt die Klasse die Wasserscheidentransformation aus und speichert das Ergebnis in der Hilfsklasse *MosaikGraph*. Diese Klasse dient dazu ein Mosaikbild der Regionskarten zu erstellen.

Die Datenstruktur des MSRTree ist durch die gleichnamige Klasse *MSRTree* umgesetzt. *MSRTree* enthält Methoden zum Zugriff auf alle Knoten der erzeugten Ebenen und kann für jede Ebene Regionskarten erzeugen, die mithilfe der Klasse *voiHandler* gespeichert werden können. Weiterhin ermöglicht es, die Klasse auf die Eigenschaften der einzelnen Knoten zuzugreifen, um die statistischen Werte der Regionen auszulesen. Die verschiedenen Knotentypen des MSRTree, die unter 3.2.4 genannt sind, werden durch die Klassen *MSRLeafNode* (Blattknoten), *MSRBranchNode* (Astknoten) und *MSRBoleNode* (Stammknoten) repräsentiert.

Als Ergebnis sind die beiden Programme ClusterV2.exe und ClusterV3.exe entstanden. ClusterV2.exe ermittelt die Regionen mit den Verfahren RG1, RG2 und WST mit 6er und 26er Nachbarschaft. ClusterV3.exe liefert die Ergebnisse des MSRTree. Zu Testzwecken wurde die Anzahl der möglichen Ebenen allerdings auf 2 beschränkt.

4. Experimentelle Validierung

Das Ergebnis dieser Arbeit sind die beiden Programme ClusterV2.exe und ClusterV3.exe, die die beschriebenen Verfahren umsetzen. Die beiden entstandenen Programme sind einfache Konsolenanwendungen. Als Eingabe dienen jeweils die t-Karten und eine Datei, in der die Parameter und die Dateipfade für die t-Karten gespeichert werden. Zu diesen Programmen gibt es eine Anleitung und eine Dokumentation des zugrundeliegenden Quelltextes.

Die Auswertung der Testdatensätze hat im Schnitt nur wenige Sekunden benötigt. Das Speichern der Ergebnisse als „.voi“ Datei ist die Funktion, die die meiste Zeit in Anspruch nimmt. Je nach Anzahl der gefundenen Regionen kann es sich beim Speichern um einen Zeitaufwand von dreißig Sekunden bis hin zu ein bis zwei Minuten handeln.

4.1. Experimente

4.1.1. Ziel

Ein erstes Ziel der durchgeführten Experimente ist, zu demonstrieren, dass die Verfahren den unter 1.3 genannten Zielstellungen entsprechen, d.h., dass sie automatisiert, robust und wiederholbar sind und intuitive Parameter besitzen. Ein zweites Ziel soll die Überprüfung der Plausibilität der Ergebnisse sein.

Um die genannten Ziele zu erreichen werden ein Datensatz mit allen Methoden und verschiedenen Parametereinstellungen ausgewertet und die erzeugten Ergebnisse verglichen. Da es keinen „Goldstandard“ für den Vergleich gibt, wird sich auf die Aussage von Experten verlassen, was die Plausibilität der Ergebnisse angeht.

4.1.2. Verwendete Daten

Bei dem Datensatz, der dem Experiment für die Vergleiche der Verfahren zugrunde liegt handelt es sich um eine t-Karte, die den konjunktiven Kontrast bei Menschenstimmen aktiv und bei Menschenstimmen stärker aktiv als bei Musikinstrumenten oder Tierlauten abbildet. Der Datensatz ist im Rahmen eines Experimentes zur genaueren Parzellierung des auditorischen Kortex entstanden (Bethmann et al. 2009). Weiterhin wurde der zu untersuchende Bereich auf den Temporallappen der linken Hirnhemisphäre beschränkt.

Für die Plausibilitätsprüfung wurde eine t-Karte mit demselben Kontrast (bei Menschenstimmen aktiv und bei Menschenstimmen stärker aktiv als bei Musikinstrumenten oder Tierlauten) verwendet, allerdings wurde auf dieser Karte der zu untersuchende Bereich nicht eingeschränkt.

4.1.3. Durchführung

Die dem Experiment zugrunde liegende t-Karte wurde mit dem Programm BrainVoyagerTM von einem Experten erzeugt.

Für den Vergleich der Verfahren wurden die unter 4.1 genannten Programme ClusterV2.exe und ClusterV3.exe eingesetzt.

Die t-Karte diente als Eingabe für die Programme. Für die Wasserscheidentransformation und für den MSRTree wurde jeweils nur ein Versuch durchgeführt, da diese beiden Methoden, bis auf die anzugebende minimale Clustergröße g_{\min} und den Schwellenwert s , parameterfrei sind. Für die Region-Growing Verfahren wurden mehrere Testläufe mit verschiedenen Parameterwerten durchgeführt. Dabei wurde als minimale Clustergröße g_{\min} bei allen Experimenten $g_{\min} = 4$ angenommen und der untere Schwellenwert s mit $s = 4$. Beide Werte sind von Experten als Richtwerte vorgegeben worden und stellen Auswertungsstandards dar. Die nachfolgende Tabelle 1 gibt die Parameterwerte, die für die Durchführung des Experimentes mit den Region-Growing Verfahren verwendet wurden, an.

Versuch	Maxima Variation (var_{max})	Regionenvariation (var_{reg})
1	10 %	10 %
2	10 %	50 %
3	10 %	90 %
4	50 %	10 %
5	50 %	50 %
6	50 %	90 %
7	90 %	10 %
8	90 %	50 %
9	90 %	90 %

Tabelle 1, die Tabelle zeigt die Einstellungen der Parameterwerte für verschiedene Testläufe.

Hierbei steht var_{max} für die von RG1 verwendete Variation der Maxima bei der Rauschbehandlung wie im Algorithmus 2 Schritt 2-4 erwähnt. Die Regionenvariation var_{reg} ist bei RG1 der maximale Unterschied benachbarter Voxel und bei RG2 dient sie zusätzlich zum überprüfen der Abweichung vom Mittelwert. Die angegebenen Prozentwerte geben an, um wie viel Prozent ein getesteter Wert von seinem Vergleichswert abweichen darf. Unterscheiden sich die Vergleichswerte um mehr als den angegebenen Prozentwert, werden sie nicht als ähnlich betrachtet.

Als Ergebnis des Experiments sind für jeden Testlauf „.voi“ Dateien erzeugt worden, in denen die gefundenen Regionen gespeichert sind. Für die Analyse der Ergebnisse wurde das Programm BrainVoyagerTM genutzt.

Für den Vergleich der Verfahren wurden die Anzahl der erzeugten Regionen sowie deren Lage und durchschnittliche Größe gegenübergestellt.

Für die Plausibilitätsprüfung wurden die Zeitverläufe aus dem der genutzten t-Karte zugrundeliegenden fMRT-Datensatz für einige ausgewählte Regionen mit einem von den Experten vorhergesagten Verhalten für die Zeitverläufe verglichen.

4.1.4. Ergebnisse

4.1.4.1. Vergleich der Verfahren

Die nachfolgenden Abbildungen zeigen die Ergebnisse der einzelnen Verfahren. In Abbildung 8 sind alle Verfahren gegenübergestellt. Die Abbildungen 9 und 10 zeigen die Ergebnisse der Region-Growing Verfahren mit den verschiedenen angewendeten Parameterwerten. Bei allen Abbildungen handelt es sich um Screenshots aus BrainVoyager™, die die Schicht $x = -56$ in der sagittalen Ansicht darstellen.

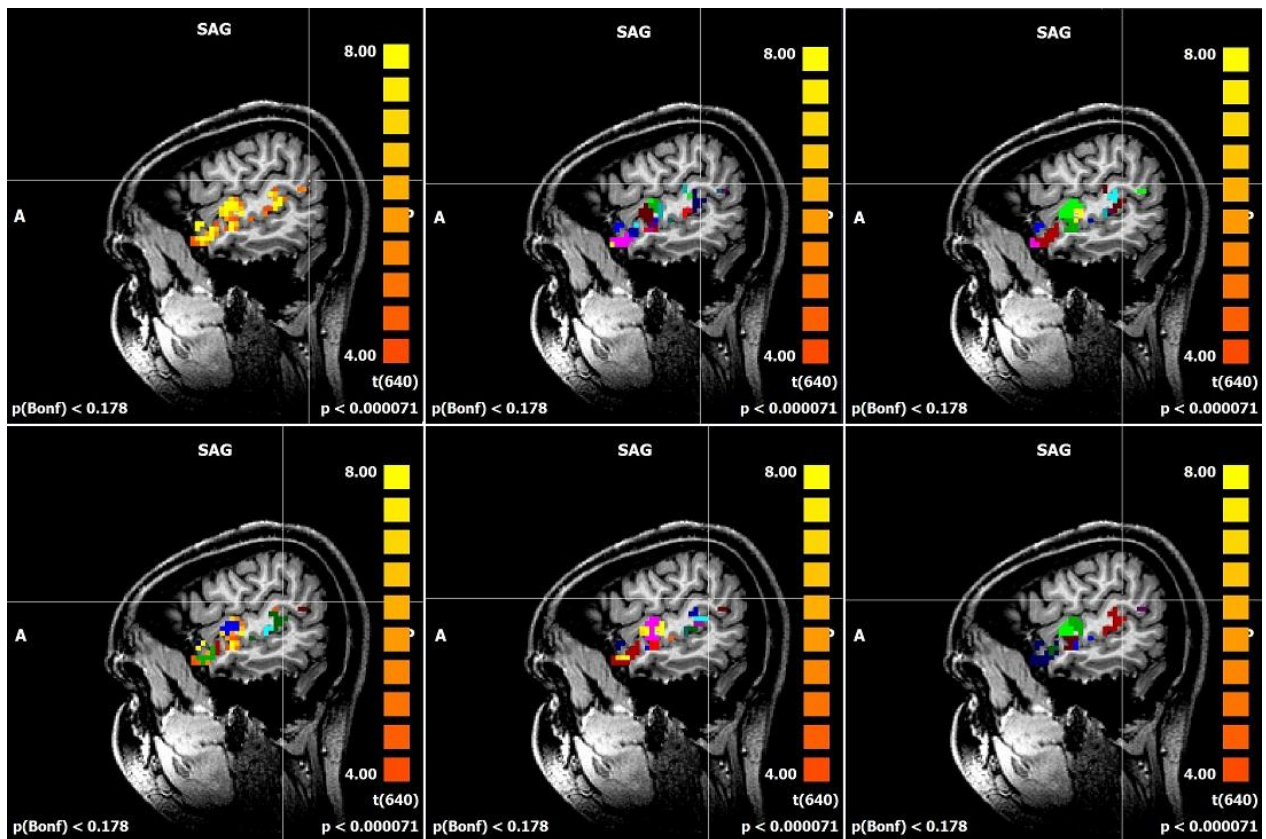


Abbildung 8: Auf der Abbildung sind die Ergebnisse aller Verfahren gegenübergestellt. In der oberen Reihe sieht man von links nach rechts: Die originale statistische Karte, die WST mit 6er Nachbarschaft und die WST mit 26er Nachbarschaft. In der unteren Reihe sind von links nach rechts die Ergebnisse von RG1 ($var_{max} = 10$, $var_{rg} = 50$), RG2 ($var_{rg} = 50$) und der 2. Ebene des MSRTrees abgebildet.

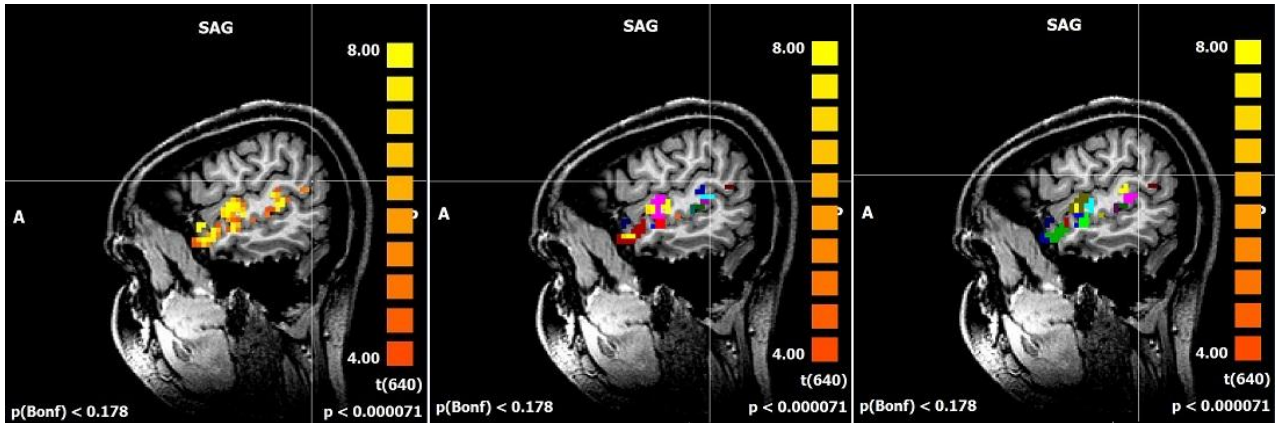


Abbildung 9: Auf dieser Abbildung sind die Ergebnisse des Verfahrens RG2 gegenübergestellt. Von links nach rechts sieht man: Die originale statistische Karte, RG2 mit $\text{var}_{rg} = 50$ und RG2 mit $\text{var}_{rg} = 90$. Auf eine Abbildung von RG2 mit $\text{var}_{rg} = 10$ wurde verzichtet, da in der angezeigten Schicht keine Regionen vorhanden waren.

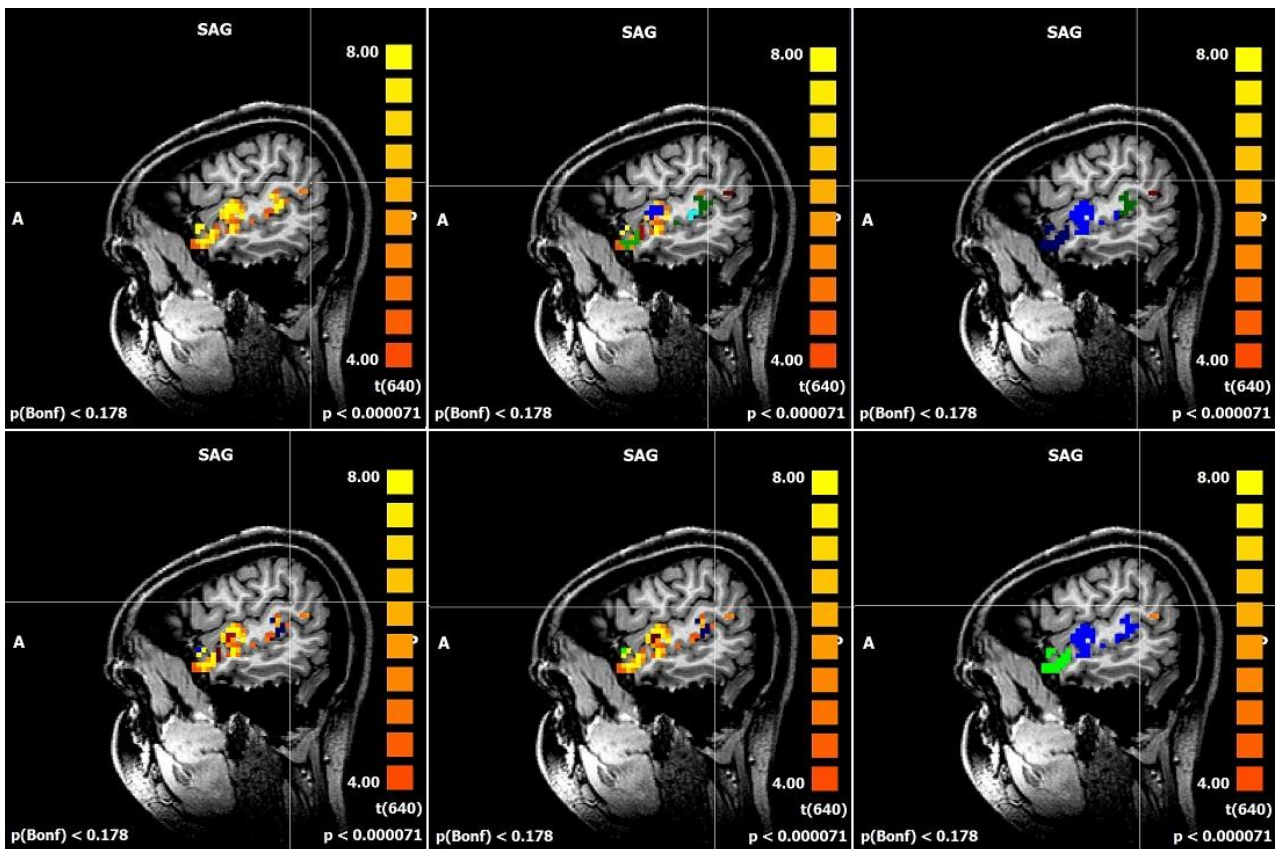


Abbildung 10: Auf dieser Abbildung sind die Ergebnisse des Verfahrens RG1 gegenübergestellt. In der ersten Reihe sieht man von links nach rechts: die originale statistische Karte, RG1 mit $\text{var}_{max} = 10$ und $\text{var}_{rg} = 50$ sowie RG1 mit $\text{var}_{max} = 10$ und $\text{var}_{rg} = 90$. In der zweiten Reihe sind von rechts nach links RG1 mit $\text{var}_{max} = 50$ und $\text{var}_{rg} = 50$, RG1 mit $\text{var}_{max} = 90$ und $\text{var}_{rg} = 50$ sowie RG1 mit $\text{var}_{max} = 90$ und $\text{var}_{rg} = 90$ abgebildet.

Die nachfolgende Tabelle 2 gibt für die Wasserscheidentransformation und den MSRTree die Anzahl der gefundenen Regionen und die durchschnittliche Regionengröße an. Tabelle 3 listet die Anzahl der gefundenen Regionen und die durchschnittliche Regionsgröße für die beiden Region-Growing Verfahren auf.

Verfahren	Anzahl Regionen	Durchschnittliche Regionengröße
WST 6er Nachbarschaft	75	4.1
WST 26er Nachbarschaft	26	11.7
MSRTree 1. Ebene	57	5.4
MSRTree 2. Ebene	16	18.6

Tabelle 2, Anzahl und durchschnittliche Größe der entstandenen Regionen aus der WST und dem MSRTree.

Verfahren	var _{max}	var _{reg}	Anzahl Regionen	Durchschnittliche Regionengröße
RG1	10	10	1	4
RG1	10	50	11	19.3
RG1	10	90	7	41.1
RG1	50	10	0	0
RG1	50	50	7	12.9
RG1	50	90	2	40.5
RG1	90	10	2	4
RG1	90	50	5	19.6
RG1	90	90	2	124
RG2	-	10	3	4.3
RG2	-	50	18	13.7
RG2	-	90	19	14.6

Tabelle 3, Anzahl und durchschnittliche Größe der entstandenen Regionen aus RG1 und RG2 mit verschiedenen Parameterwerten.

4.1.4.2. Plausibilitätsprüfung

Die Abbildungen zeigen die Zeitverläufe für die Verfahren RG1 (Abbildung 11), RG2 (Abbildung 12), WST mit 26er Nachbarschaft (Abbildung 13).

Durch die Experten wurde für die Zeitverläufe das folgende Verhalten vorhergesagt: Je weiter frontal eine Region im temporalen Lappen liegt, desto stärker reagiert sie auf Stimmen und desto schwächer reagiert sie auf Musikinstrumente und Tierstimmen. Wie auf den abgebildeten Diagrammen zu erkennen ist, verhalten sich die Zeitverläufe der erzeugten Regionen entsprechend der von den Experten getroffenen Vorhersage.

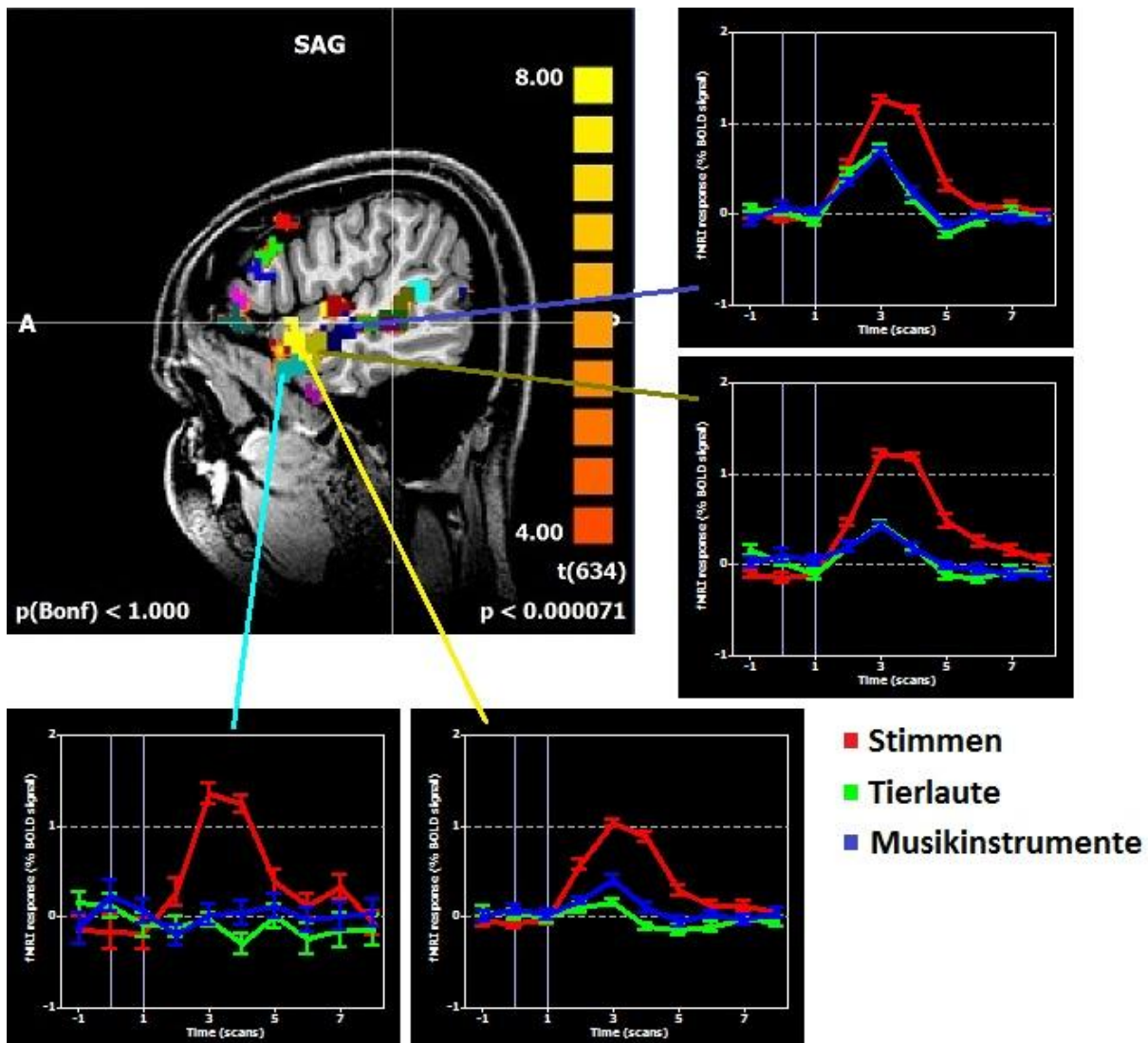


Abbildung 11: Die Abbildung zeigt einen Vergleich von Zeitverläufen ausgewählter Regionen. Die Regionen sind mit dem Verfahren RG1 ($var_{max} = 10$, $var_{rg} = 50$) entstanden.

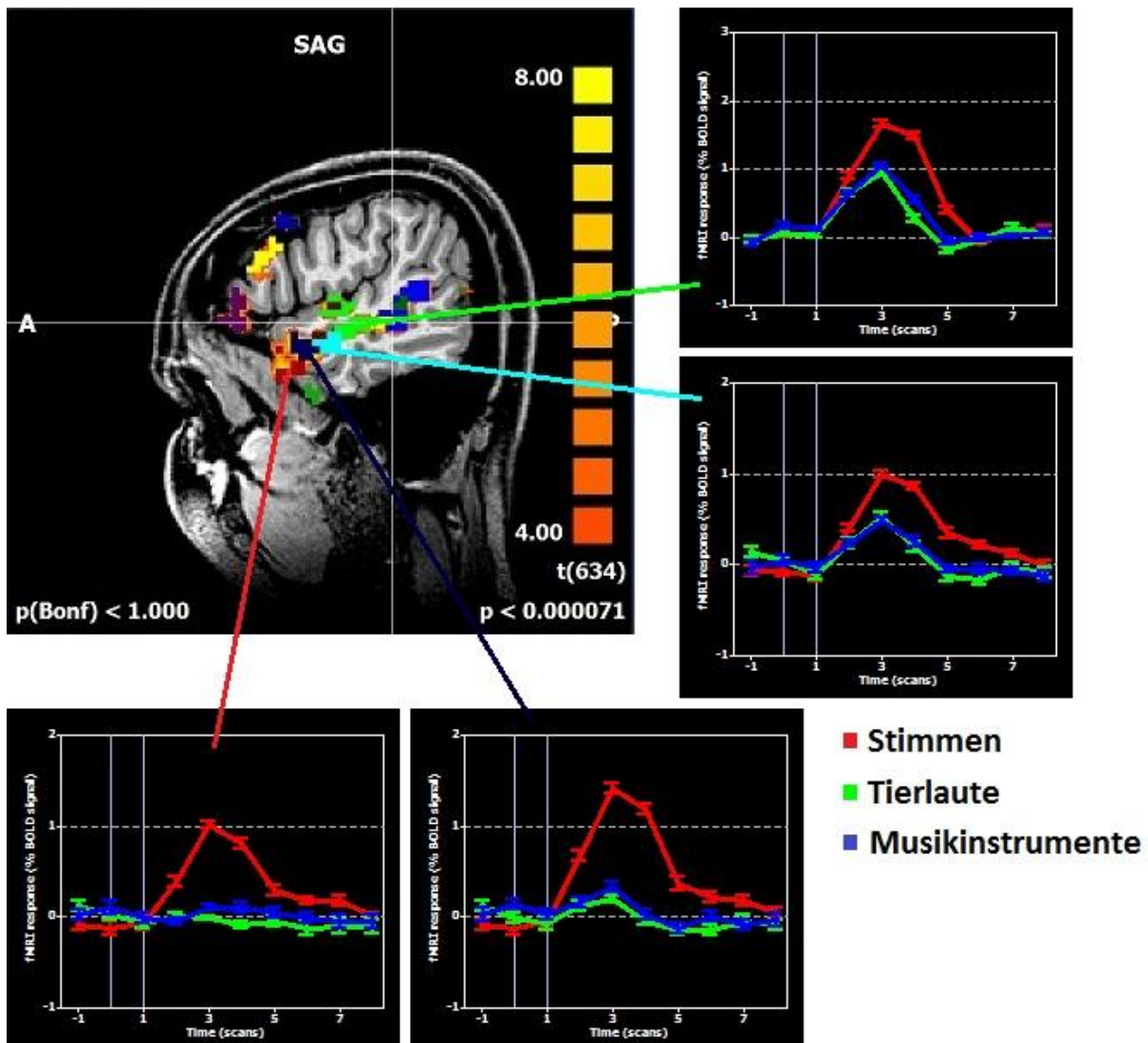


Abbildung 12: Die Abbildung zeigt einen Vergleich von Zeitverläufen ausgewählter Regionen. Die Regionen sind mit dem Verfahren RG2 ($var_{rg} = 50$) entstanden.

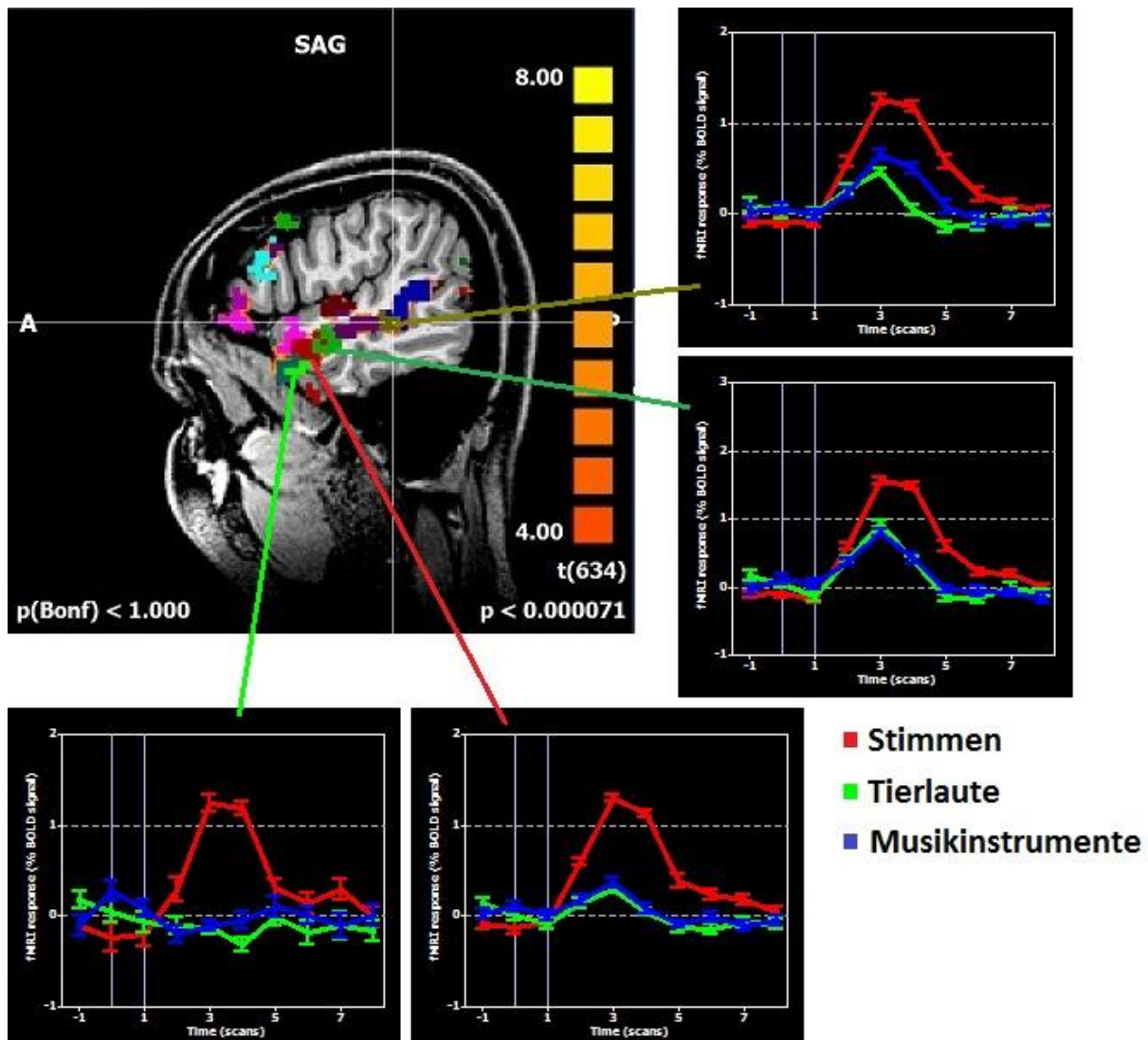


Abbildung 13: Die Abbildung zeigt einen Vergleich von Zeitverläufen ausgewählter Regionen. Die Regionen sind mit dem Verfahren WST (26er Nachbarschaft) entstanden.

4.1.5. Auswertung

Die gezeigten Ergebnisse lassen den Schluss zu, dass eine Segmentierung auf t-Karten zu plausiblen Regionen wie im Abschnitt 4.2.4.2 gezeigt führen kann. Die gezeigten Zeitverläufe (Abbildungen 11-13) bestätigt die Annahme der Experten, daraus lässt sich schließen, dass die gefundenen Regionen als Grundlage für eine umfangreichere Datenanalyse dienen können.

Beim Vergleich der Verfahren kann man, wie in Abbildung 8 gezeigt, sehen, dass die einzelnen Verfahren Regionen liefern, die eine ähnliche räumliche Lage besitzen. Weiterhin ist zu sehen, dass sich die Regionen zum größten Teil auf dem Kortex befinden, was den Annahmen über die Anatomie des Gehirns entspricht (siehe 3.1.2).

Der Vergleich der Ergebnisse von RG1 in Abbildung 10 zeigt deutlich, wie anfällig dieses Verfahren gegenüber den Parameterwerten ist. Wird var_{\max} niedrig gewählt, so werden im Vorverarbeitungsschritt nur wenige lokale Maxima aus der Eingabemenge für den Algorithmus entfernt, was dazu führt, dass mehr Saatpunkte vorhanden sind und somit mehr Regionen entstehen können. Wird var_{\max} hoch gewählt, werden bei der Rauschbehandlung im Vorverarbeitungsschritt fast alle lokalen Maxima entfernt, wodurch nur noch wenige Regionen entstehen können. Es ist sinnvoll, den Parameter var_{\max} niedrig (10%) zu wählen, da somit weniger lokale Maxima, die potenzielle Regionenzentren sind, im Vorverarbeitungsschritt entfernt werden.

Weiterhin hat der Parameter var_{rg} einen starken Einfluss auf die Größe der entstandenen Regionen, wie es Tabelle 3 zu entnehmen ist. Ein zu gering gewählter Wert (10%) führt dazu, dass nur wenige Voxel das in Abschnitt 3.2.1 beschriebene Homogenitätskriterium erfüllen. Hierdurch werden die Regionen relativ klein ($|\bar{r}_i| \leq g_{\min}$). Da Regionen kleiner der Mindestregionsgröße g_{\min} bei RG1 entfernt werden, enthält die entstehende Regionskarte nur sehr wenige Regionen. Wird var_{rg} hoch (90%) gewählt, so erfüllt fast jeder Voxel das Homogenitätskriterium, was zu sehr großen Regionen ($|\bar{r}_i| > 40$) führt. Plausible Ergebnisse erhält man mit einem Wert zwischen 40 – 60% für var_{rg} . Aus der starken Anfälligkeit von RG1 gegenüber Veränderungen der Parameterwerte lässt sich schließen, dass dieses Verfahren nicht robust ist.

Im Gegensatz dazu ist RG2 nicht von dem Parameter var_{\max} abhängig, sondern nur von var_{rg} . Zusätzlich lässt sich der Tabelle 3 entnehmen, dass RG2 nicht so anfällig für Veränderungen des Parameterwertes ist (Abbildung 9). Wie bei RG1 führt ein zu niedrig gewählter Wert dazu, dass das Homogenitätskriterium nur von wenigen Voxeln erfüllt wird. Allerdings ist zu sehen, dass auch ein hoher Wert (90%) von var_{rg} zu ähnlichen Ergebnissen wie ein moderat gewählter Wert (50%) führt. Auch für RG2 liegt ein sinnvoller Wert für var_{rg} zwischen 40 – 60%. Die Unabhängigkeit von dem Parameter var_{\max} und die geringere Anfälligkeit für Veränderungen des Parameters var_{rg} zeigen, dass RG2 ein robusteres Verfahren als RG1 ist.

Bei der Wasserscheidentransformation lässt sich in Tabelle 2 erkennen, dass die Wahl der Nachbarschaft einen großen Einfluss auf die Anzahl der entstandenen Regionen sowie deren Größe hat. Der Unterschied in Anzahl und Größe der Regionen lässt sich damit erklären, dass bei einer größeren Nachbarschaft mehr Voxel als an eine Region angrenzend gesehen werden, was dazu führt, dass auch mehr Voxel bei der Flutung (siehe Algorithmus 3

Schritte 5-12) überprüft und der Region hinzugefügt werden können. Wie im Abschnitt 3.2.3 beschrieben ist der Einsatz der 6er Nachbarschaft im Hinblick auf die Anatomie sinnvoller.

Der erkennbare Unterschied in der Anzahl der Regionen und der durchschnittlichen Regionsgröße zwischen der WST in 6er Nachbarschaft und der initialen Ebene (Astknoten) des MSRTree lässt sich damit erklären, dass beim MSRTree, im Gegensatz zur WST, Regionen, die kleiner g_{\min} sind, entfernt werden.

5. Fazit

Das Ziel dieser Arbeit war es, Methoden vorzuschlagen, welche die aus den fMRT-Daten entstandenen statistischen Karten auf mögliche funktionell abgrenzbare Gehirnareale untersuchen. Dabei sollten die gewählten Methoden datengetrieben, robust und automatisiert sein. Weiterhin sollte eine Datenstruktur zum Verwalten der Ergebnisse implementiert und die Ergebnisse der vorgeschlagenen Methoden auf Plausibilität durch Experten geprüft werden.

Als Ergebnis dieser Arbeit wurden verschiedene Verfahren (RG1, RG2, WST) erläutert, die der Zielstellung entsprechen.

Das Verfahren RG1 diene dazu, zu ermitteln, ob eine Segmentierung auf t-Karten überhaupt sinnvoll ist. Hierzu bedient sich das Verfahren einem einfachen Region-Growing, welches anhand der gefundenen lokalen Maxima eine erste Abschätzung der möglichen Regionen liefert. Diese Abschätzung liefert eine Einteilung in Regionen, bei denen man davon ausgehen kann, dass sie möglicherweise unabhängig voneinander sind. Damit wurde bestätigt, dass eine Segmentierung auf t-Karten sinnvoll sein kann. Ein Problem des Verfahrens ist, dass man die Variation für das Hinzufügen der Voxel var_{rg} und die Variation für die Rauschbehandlung var_{\max} angeben muss. Durch ungünstig gewählte Parameter kann das Ergebnis negativ beeinflusst werden.

Nachdem durch RG1 nachgewiesen wurde, dass eine Segmentierung möglich ist, diene RG2 dazu, die erhaltenen Ergebnisse zu verbessern. Die mit diesem Verfahren erzeugten Regionen sind etwas größer als mit RG1. Weiterhin sind die Regionszentren teilweise etwas verschoben, da keine vorherige Rauschberücksichtigung erfolgt, sondern kleine Regionen im späteren Verlauf des Algorithmus zusammengefasst werden. Dadurch

wurde verhindert, dass eventuelle Regionszentren schon vor dem Region-Growing entfernt werden. RG2 liefert ein ähnliches Ergebnis wie RG1. Auch hier lässt sich feststellen, dass das Verfahren möglicherweise getrennte Regionen liefert. Im Gegensatz zu RG1 ist RG2 allerdings nicht so anfällig für Störungen durch eine ungünstige Wahl der Parameterwerte.

Ein Problem bei beiden Region-Growing Verfahren ist, dass die Regionen teilweise sehr klein ausfielen. Laut Aussagen der Experten, die die Methoden getestet haben, wären größere Regionen sinnvoll. Weiterhin kann es bei zu gering gewählter Schrittweite und zu streng gewählter Variation zu Lücken zwischen den Regionen kommen, da es Voxel gibt, die nach den gegebenen Kriterien zu keiner Region hinzugefügt werden können. Aufgrund der Parameterabhängigkeit wurde versucht das Problem der Segmentierung anstatt mit dem Top-Down-Ansatz Region-Growing mit dem Bottom-Up-Ansatz der Wasserscheidentransformation zu lösen.

Da die gegebenen Daten in ihrer Form eine Segmentierung mittels Inhomogenitätskriterium erlauben, wurde getestet, ob die Wasserscheidentransformation ähnliche oder bessere Ergebnisse bei der Segmentierung liefert. Ein großer Vorteil ist, dass die Daten vor der Segmentierung nicht extra bearbeitet werden müssen, da sie schon in einer geeigneten Form für die Transformation vorliegen. Die WST mit der 26er Nachbarschaft liefert recht große zusammenhängende Regionen und wurde bei einem Vergleich mit RG1 und RG2 von den Experten als besser bewertet, was bedeutet, dass die Region an sich plausibler erschienen. Ein Problem bei der 26er Nachbarschaft ist allerdings, dass es zu Überläufen zwischen verschiedenen Regionen, die eigentlich getrennt sein müssten, kommen kann. Zum Beispiel kann es vorkommen, dass sich eine Region von einen auf den nächsten Gyrus fortgesetzt hat, obwohl sich die Gyri gegenüber lagen. Um diesem Problem entgegenzuwirken, wurde die Nachbarschaft eingeschränkt. Mit einer 6er Nachbarschaft wird die Anatomie des Gehirns besser abgebildet und die Überläufe werden eingeschränkt. Ein Nachteil bei der eingeschränkten Nachbarschaft ist allerdings, dass die Regionen sehr viel kleiner und damit zahlreicher werden. Da die zugrundeliegenden Daten verrauscht sind, war allerdings mit einer Übersegmentierung durch die Wasserscheidentransformation zu rechnen, da diese erwiesenermaßen sehr empfindlich gegenüber lokalen Variationen ist.

Alle bisher genannten Verfahren erzeugen Regionen die nach Meinung der Experten zwar plausibel aber teilweise zu klein waren. Daraus folgt, dass eine weitere Zusammenfassung nötig ist. Um diese weitere Zusammenfassung zu unterstützen ist im Rahmen dieser Arbeit der MSRTree entstanden der mithilfe der

Wasserscheidentransformation eine Baumstruktur mit einer immer weitergehenden Zusammenfassung der Regionen liefern kann. Der MSRTree repräsentiert die, in der Zielstellung geforderte, Datenstruktur zum Speichern und Verwalten der Ergebnisse. Die erste Ebene des MSRTree liefert ähnliche Ergebnisse wie die WST mit 6er Nachbarschaft. Die Unterschiede begründen sich in dem Entfernen der Regionen, die der Mindestregionsgröße g_{\min} nicht entsprechen. Diese wurden in der WST nicht entfernt. Die weitere Zusammenfassung der Regionen sollte das Problem der Übersegmentierung und der daraus resultierenden zu kleinen Regionen lösen. Allerdings liefert der MSRTree schon bei dem zweiten Zusammenfassungsschritt ein paar Regionen, die unverhältnismäßig groß sind. Aus diesem Grund ist auf die Erzeugung weitere Ebenen verzichtet worden, da die erzeugten Regionen mit jeder weiteren Ebene noch größer werden. Diese Funktionalität unterstützt der MSRTree dennoch, da sie für eine interaktive Analyse sinnvoll sind, um größere zusammenhängende Gebiete zu lokalisieren.

Die Ergebnisse der verschiedenen Verfahren lassen darauf schließen, dass eine erste Abschätzung von möglichen Regionen mittels einer Segmentierung auf den t-Karten möglich ist. Allerdings ist diese Abschätzung durch teils zu kleine Regionen gekennzeichnet, was eine weitere Zusammenfassung dieser Regionen nötig macht. Der für diesen Zweck entworfene MSRTree unterstützt die weitere Zusammenfassung. Allerdings führt die Zusammenfassung zu zum Teil nicht plausiblen Regionen. Der Grund hierfür ist, dass die durchgeführte Wasserscheidentransformation auf der zweiten Ebene des MSRTree nur den Mittelwert der Regionen der ersten Ebene zur Verfügung hat. Hierbei handelt es sich um Mittelwerte von zu Regionen zusammengefassten statistischen Wahrscheinlichkeitswerten. Daraus folgt, dass diese Werte alleine für eine weitere Zusammenfassung nicht geeignet sind. Das Einbringen weiterer Kriterien in die Zusammenfassung könnte dieses Problem beheben. Solche Kriterien könnten unter anderem der Abstand zwischen regionsbildenden Extrema oder auch ein Unterschiedsmaß zwischen kleinstem und größtem Wert einer Region sein.

Zusammenfassend lässt sich sagen, dass die gewählten Methoden dazu geeignet sind, eine erste Abschätzung von möglichen getrennten Arealen im Gehirn zu ermitteln, die als eine Grundlage für eine genauere Datenanalyse dienen kann. Eine genauere Zusammenfassung der erzeugten Regionen zu größeren zusammenhängenden Gebieten lässt sich allerdings nicht ohne das Einbringen von zusätzlichen Informationen erreichen.

6. Ausblick

Wie im vorherigen Abschnitt erwähnt, können die Verfahren eine erste Abschätzung für die gesuchten Regionen auf Basis der statistischen Karten liefern. Um das Ergebnis der Verfahren zu verbessern ist das Einbringen weiterer Informationen (Anatomie, Zeitverläufe aus den fMRT-Daten) nötig. Mithilfe von zusätzlichen Informationen könnten bessere Kriterien zum Bilden der Regionen gefunden werden. Sinnvoll wäre es Veränderungen des mittleren Zeitverlaufs einer Region zu betrachten, wenn ein neues Voxel der Region hinzugefügt wird.

Der Aufbau des MSRTree gestattet es, das Verfahren um weitere Eingabeparameter zu erweitern oder bei alternativen Methoden einzusetzen. So könnten beispielsweise statistische Karten für verschiedene Fragestellungen erstellt werden (zum Beispiel im Bezug auf die im Rahmen dieser Arbeit verwendeten Daten: Ein Vergleich der Ergebnisse der genutzten Daten mit Ergebnissen auf Basis einer statistischen Karte die den Kontrast Menschenstimmen gegen Ruhe anzeigt) und die jeweils auf ihnen gefundenen Regionen verglichen und kombiniert werden. Eine quantitative Analyse der Regionengrößen von mehreren statistischen Karten könnte helfen, die Frage zu klären, ob die von den Verfahren gefundenen Regionen wirklich zu klein sind.

Eine sinnvolle und wichtige Erweiterung für den MSRTree ist die Anbindung an ein interaktives Interface. Durch diese Erweiterung könnten zum einen die erzeugten Ergebnisse besser dargestellt und ausgewertet werden und zum anderen wäre es möglich dem Nutzer Eingriffsmöglichkeiten für die Zusammenfassung der Regionen zu geben (Beispielsweise das Setzen von Grenzen, um das Zusammenfügen von zwei Regionen zu verhindern). Durch die Erweiterung des MSRTrees um ein interaktives Interface könnte somit den Experten ein einfach zu handhabendes Werkzeug zur Verfügung gestellt werden, mit dessen Hilfe sie schnell eine erste Abschätzung der zu analysierenden Daten erhalten können, was den Zeitaufwand der vollständigen Datenanalyse drastisch verkürzen könnte.

7. Bibliographie

Adams R. and Bischof L.; “Seeded region growing”; IEEE Trans. PAMI, Vol. 16, No. 6, pp 641-647; 1994.

Bethmann Anja, Scheich Henning and Brechmann André; "The role of the temporal lobe for extracting proper names from sound vision"; 3rd International Conference on Auditory Cortex; August 29 - September 2, 2009.

Chen, S., C.A. Bouman, and M.J. Lowe; “Clustered component analysis for functional MRI.”; IEEE Transactions on Medical Imaging, 23(1): p. 85-98; 2004.

Defeng Wang, Lin Shi, Daniel S. Yeung, Pheng-Ann Heng, Tien-Tsin Wong, and Eric C. C. Tsang; “Support Vector Clustering for Brain Activation Detection”; J. Duncan and G. Gerig (Eds.): MICCAI 2005, LNCS 3749, pp. 572–579; 2005.

Deleus Filip and Van Hulle Marc M.; “A Connectivity-Based Method for Defining Regions-of-Interest in fMRI Data”; IEEE Transactions on Image Processing, VOL.18, NO.8; August 2009.

Formisano, E., Esposito, F., Di Salle, F. & Goebel R.; “Cortex-based independent component analysis of fMRI time-series.”; Magnetic Resonance Imaging, 22, 1493-1504; 2004.

Friston, K.J., Holmes A.P., Worsley K.J., Poline J.P., Frith C.D., and Frackowiak R.S.J., “Statistical parametric maps in functional imaging: A general linear approach,”; Human Brain Mapping, vol.2, pp.189–210; 1995.

Gang Lin, Umesh Adiga, Kathy Olson, John F. Guzowski, Carol A. Barnes, Badrinath Roysam; “A Hybrid 3-D Watershed Algorithm Incorporating Gradient Cues & Object Models for Automatic Segmentation of Nuclei in Confocal Image Stacks”; Cytometry, 56A:23-36; 2003.

Kostas Haris, Serafim N. Efstratiadis, Nicos Maglaveras, Aggelos K. Katsaggelos; „Hybrid Image Segmentation Using Watersheds and Fast Region Merging”; IEEE Transactions on Image Processing, Vol. 7, No. 12, December 1998.

Kruggel F. and von Cramon D.Y.; “Temporal properties of the hemodynamic response in functional MRI,”; Human Brain Mapping, vol.8, pp.259–271, 1999.

Vincent L. and Soille P.; "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations"; IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 6, June 1991.

Wegner S., Stalling D., Hege H.C., Oswald H., Fleck E.; „Die 3D-Wasserscheidentransformation auf Graphenebene – Eine Anwendung für die Hypertermieplanung“; 1997.

Worsley, K.J.; "Detecting activation in fMRI data."; Statistical Methods in Medical Research, 12(5):401-418; 2003.

Zilles K.; Palomero-Gallagher N.; "Cyto-, Myelo-, and Receptor Architectonics of the Human Parietal Cortex"; NeuroImage 14: S8-S20; 2001.

Zilles K., Schleicher A., Palomero-Gallagher N., Amunts K.; "Quantitative Analysis of Cyto- and Receptor Architecture of the Human Brain"; In: Toga, A. W., Mazziotta, J. C. (Eds.), Brain Mapping: The Methods. Elsevier Science, pp. 573-602; 2002.

Zilles K.; "Architecture of the Human Cerebral Cortex."; In: Paxinos, G. (Ed.), The Human Nervous System Amsterdam, pp. 997-1055; 2004.

Abbildungsverzeichnis

Abbildung 1, Darstellung von definierten Regionen des Temporallappens. Seite 7. Quelle: Poster: Bethmann Anja, Scheich Henning and Brechmann André; "The role of the temporal lobe for extracting proper names from sound vision"; 3rd International Conference on Auditory Cortex; August 29 - September 2, 2009.

Abbildung 2, Grafische Darstellung des GLM. Seite 13. Quelle: BrainVoyagerTM-Hilfe.

Alle weiteren Abbildungen wurden selbst erstellt.