

Verarbeitung von GIS-Daten im Browser

Lennart Karsten

Hochschule für Angewandte Wissenschaften

Department Informatik

Berliner Tor 7

20099 Hamburg, Germany

Email: lennart.karsten@haw-hamburg.de

I. EINLEITUNG

Die Verarbeitung von Geodaten in klassischen Desktop Anwendungen ist seit einigen Jahren gängige Praxis. Zunehmend ist es gewünscht kollaborativ an Daten zu arbeiten. Um den wachsenden Anforderungen gerecht zu werden und Geodaten im Browser besser handhaben zu können bedarf es neuer Techniken.

Die MARS Websuite stellt Werkzeuge bereit um diesen Anforderungen gerecht zu werden. Die Fokussierung liegt auf der afrikanischen Savanne, da diese zu den, am stärksten bedrohten Ökosystemen unserer Erde gehört [1].

II. DIE MARS ARBEITSGRUPPE

Die MARS (Multi-Agent Research & Simulation) Arbeitsgruppe um Prof. Dr. Thiel-Clemen befasst sich mit der Erforschung und Entwicklung von Simulationen der afrikanischen Savanne unter Zuhilfenahme von Multi-Agenten Systemen (MAS). Ziel ist es, ein besseres Verständnis, der dortigen Umwelt zu erlangen.

Das MARS Framework besteht aus verschiedenen Komponenten, von denen für diese Arbeit nur die Websuite von Bedeutung ist. Sie ist der eigentlichen Simulation in MARS LIFE vorangestellt und ist für Import, Datenaufbereitung und dem Zusammenfügen von Simulationsmodell und Daten zuständig.

Der Fokus dieser Arbeit liegt auf DEIMOS, welches Teil der Websuite ist, welche die in den folgenden Kapiteln genannten Arbeiten ausführt. Die Websuite besteht aus den folgenden Komponenten.

A. MARS ROCK

ROCK basiert auf dem Konzept eines Datawarehouses zur Speicherung von relationalen Daten und wird in der Masterarbeit von Busch[3] beschrieben.

Dieser Ansatz hat sich in der Praxis als unflexibel und langsam herausgestellt. Aus diesem Grund wird ROCK durch MariaDB und InfluxDB ersetzt.

MariaDB bildet tabellarische Daten in einer Tabelle ab. InfluxDB ist eine Datenbank, die für Zeitreihendaten konzipiert wurde. Zeitreihendaten werden hier entsprechen abgelegt.

B. MARS Import

MARS Import soll PHOBOS als Importkomponente für GROUND bzw. ROCK ersetzen und befindet sich derzeit in

der Entwicklung.

Der Import besteht aus je einer Ausprägung für die Art, der zu importierenden Daten. Diese sind Tabellenimport, Zeitreihenimport und GIS Import.

Zu Beginn des Imports werden die Metadaten in einen entsprechenden Service geschrieben und der Dateiupload wird begonnen. Wenn der Upload abgeschlossen ist wird dies im Metadaten Service entsprechend vermerkt.

C. MARS DEIMOS

DEIMOS arbeitet auf Basis der Daten, die durch den Import importiert werden. Daher ist es wichtig, dass Anforderungen an die Verarbeitung, beim Import berücksichtigt werden. Dies erfordert eine enge Zusammenarbeit zwischen DEIMOS und dem Import.

Die Aufgabe von DEIMOS ist es, eine Webanwendung bereit zu stellen, die es dem Endnutzer und Simulationsentwickler erlaubt, Datenbestände zu sichten, validieren und zu manipulieren.

Hierbei liegt der Fokus auf der Visualisierung unterschiedlicher Datentypen, sowie der Validierung und Plausibilitätsprüfung der Eingabedaten.

Der Benutzer wählt Daten für die Simulation aus und referenziert die Datensätze in einer Beschreibungsdatei. Diese wird an SHUTTLE übergeben.

D. MARS SHUTTLE

SHUTTLE liefert die Projektzusammenstellung von DEIMOS aus und stellt eine DSL bereit. Diese ermöglicht es, dem Benutzer das Simulationsmodell mit den Daten zu verbinden. Aus diesem Prozess entsteht die „Scenario Configuration“, welche in der Simulation verwendet wird.

III. STAND DER WISSENSCHAFT

Für die Verarbeitung in Geoinformationssystemen (GIS) sind eine Vielzahl von Datentypen und Datenrepräsentationen notwendig. Die Folgenden Werkzeuge sind für diese Arbeit von zentraler Bedeutung.

A. Tabellendaten

Tabellendaten in GIS sind im allgemeinen händisch oder automatisch erstellte Tabellen mit Messdaten unterschiedlichster Art.

Als Spezialfall sind Zeitreihendaten zu nennen. Hierbei handelt es sich um Daten mit einem zeitlichem Ablauf, wie beispielsweise die Entwicklung von Temperatur über einen bestimmten Zeitraum.

Für die Speicherung solcher Daten sind relationale Datenbanken am besten geeignet. Der MARS Import speichert Zeitreihendaten daher in einer InfluxDB¹, welche speziell für diesen Zweck konzipiert wurde.

B. Vektordaten

Vektordaten basieren auf mathematischen Funktionen und sind somit insbesondere für trigonometrische Formen und lineare Merkmale im Koordinatensystem geeignet. Sie können verlustfrei und beliebig im Raum skaliert werden.

1) *Shapefile*: Shapefiles², entwickelt von der Firma Esri, ist ein Format bestehend aus mindestens drei Dateien (.shp, .shx, .dbf), weitere Dateien können optional hinzugefügt werden. Shapefiles sind das, am häufigsten verwendete Vektorformat. Nahezu alle gängigen GIS Anwendungen unterstützen dieses Format. Programme wie ArcMap und QGIS nutzen es standardmäßig beim exportieren von Vektorebenen.

2) *GeoJSON*: GeoJSON³ ist ein offenes Format zur Speicherung von Geodaten. Entwickelt wird es von der „IETF Geographic JSON Working Group“. GeoJSON hat gegenüber Shapefiles den Vorteil, dass es einen minimalen Overhead besitzt und Menschen lesbar ist. GeoJSON lässt sich in jeder Sprache parsen, was insbesondere die Handhabung im Web vereinfacht.

C. Rasterdaten

Rasterdaten werden aus einer Matrix von quadratischen Zellen gleicher Größe gebildet. Jede Zelle erhält einen Farbwert und abhängig vom Format einen Alphawert für die Transparenz. Siehe Abb. 1.

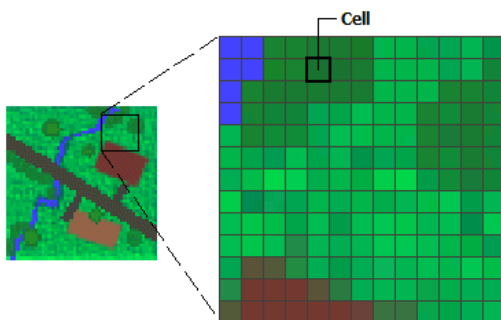


Abbildung 1. Rasterdaten. Quelle ArcGis Dokumentation⁴

1) *GeoTIFF*: TIFF(.tiff) ist ein Bildformat, welches Rasterdaten verlustfrei speichert. GeoTIFF erweitert dieses Format um Metadaten, wie der Projektion und den Geo-Koordinaten

des Bildausschnittes. Diese Eigenschaften machen das Format attraktiv für die Speicherung großer Gebietskarten. Diese könne in Abhängigkeit zum erfassten Gebiet und dessen Auflösung mehrere Gigabyte umfassen.

Für die Darstellung von großen Rasterdaten, insbesondere in einer Client & Server Architektur sind GeoTIFF Dateien daher ungeeignet und werden zu diesem Zweck in eine Mercator Pyramide (siehe III-E) umgewandelt.

D. Quadtree

Quadrees[4] sind ein Referenzierungsmodell, zur Aufteilung von zweidimensionalen Flächen in einer Baumstruktur. Jede Fläche wird in genau vier Unterbereiche (Kindknoten) unterteilt. Dies geschieht rekursiv bis zur gewünschten Tiefe. Diese Form der Flächenaufteilung wird u.a. für Mercator Pyramiden (siehe III-E) verwendet.

E. Mercator Pyramide

Raster Pyramiden bestehen aus gleichgroßen Teilabschnitten einer größeren Rasterdatei in der Mercator Projektion, bei der jede Ebene eine Detailstufe (Zoomstufe) bildet. Sie sind eine Implementation der Quadrees.

Die Mercator Projektion ist die Abbildung der Erdoberfläche auf eine quadratische Fläche. Für jede Detailstufe werden quadratische Kacheln gleicher Größe generiert.

Beginnend mit einer Kachel auf Detailstufe 0 vervierfachen sich die Anzahl der Kacheln auf jeder Ebene, bis zur gewünschten Tiefe. Die Verzweigung entspricht einem Quad-Tree, also einem Baum mit jeweils vier Kindelementen. Üblich sind Kachelgrößen von 256x256 Pixel, 19 Stufen und das Dateiformat .png.

Dieses Verfahren erlaubt es, dem Benutzer einer Webanwendung beliebig große Dateien ohne sichtbaren Qualitätsverlust zu präsentieren.

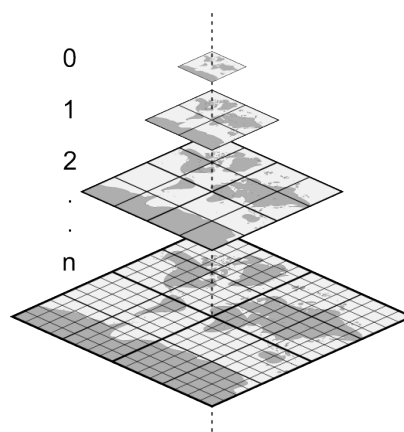


Abbildung 2. Mercator Kachel Pyramide

F. Quadkeys

Die Bing Quadkeys benennen die Flächen einer Mercator Pyramide mit einem Wert. Gewöhnlich werden Mercator Pyramiden mit Z/X/Y bezeichnet, wobei Z die Detailstufe, X

¹<https://influxdata.com/>

²<https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

³<http://geojson.org/>

die X-Achse und Y die Y-Achse ausmachen. Quadkeys codieren die Position der Fläche, indem für jede Detailstufe eine Ziffer zwischen 0 und 3 hinzugefügt wird. Siehe hierfür Abb. 3.

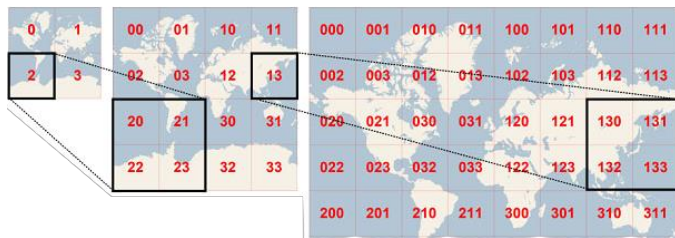


Abbildung 3. Bing Quadkeys[5]

IV. GRUNDPROJEKT

Der Begriff Geoinformationssystem(GIS) umfasst die Speicherung, Manipulation, Organisation und Darstellung von räumliche Daten. Diese Daten werden im MARS Framework als Basis für Agentensimulationen genutzt. MARS DEIMOS stellt diese Daten dar und ermöglicht es dem Benutzer Daten für Simulationen auszuwählen. Um eine bessere Benutzbarkeit zu erreichen sollen die Daten auf Korrektheit geprüft werden.

Eine unmittelbare Korrektur der erkannten Fehler soll nur in geringem Maße durchgeführt werden. Grund hierfür ist die Existenz von bekannten und etablierten GIS-Werkzeugen wie ArcGIS, QGIS und GRASS, die für derartige Aufgaben konzipiert wurden.

Voraussetzung für ein zukunftssicheres Design sind, stabile und dokumentierte Softwarelösungen für die Analyse und Manipulation, sowie eine dauerhafte Speicherung der GEO-Daten.

Ziel des Grundprojektes ist es, die aktuelle Situation zu dokumentieren und Verbesserungsvorschläge zu erarbeiten.

A. Technologievergleich

Für die Verarbeitung von Geodaten gibt es zahlreiche Bibliotheken mit unterschiedlichen Einsatzgebieten. Folgende kommen für die Analyse und Manipulation von GIS-Daten in Betracht.

1) *Geospatial Data Abstraction Library (GDAL) [C/C++]*: GDAL ist eine, in C/C++ geschriebene Geo-Bibliothek für Raster und Vektordaten. Es wird in vielen Werkzeugen als Grundlage verwendet. Werkzeuge, die GDAL verwenden sind u.a. ArcGIS, QGIS, GRASS, Google Earth, die Programmiersprache „R“, sowie die im folgenden genannten Werkzeuge mit Ausnahme GeoTools und dem GeoServer.

GDAL über den Aufruf der Konsolenanwendung direkt zu nutzen ist möglich, jedoch nicht zu empfehlen. Insbesondere die Auswertung von Rückgabewerten ist mit der unmittelbaren Verwendung umständlich, da die Konsolenausgaben interpretiert werden müssen.

Abhilfe können eine Vielzahl von GDAL Wrapper Bibliotheken schaffen. Diese ermöglichen eine bessere Handhabung.

2) *SharpMap [C#]*: SharpMap ist ein Geospatiale Bibliothek für C#, welche bereits in MARS LIVE verwendet wird. SharpMap unterstützt verschiedene Vektorformate, darunter Shapefiles. Die Unterstützung von Rasterdaten geschieht durch GDAL. Für die Darstellung von Kartendaten kann der Web Map Service (WMS) der The Open Source Geospatial Foundation (OSGeo) verwendet werden.

Die Dokumentation von SharpMap beschränkt sich auf 5 kurze Anwendungsbeispiele, eine komplette Klassendokumentation ist nicht vorhanden.

3) *DotSpatial [C#]*: DotSpatial ist neben SharpMap eine weitere Möglichkeit Geospatiale Operation auf Geodaten in C# durchzuführen. Im Vergleich zu SharpMap stehen dem Entwickler neben Tutorials und Code Ausschnitten auch ein Getting started Guide und eine SDK Dokumentation einiger Komponenten zur Verfügung.

4) *GeoTools [Java]*: GeoTools unterscheidet sich von den bereits genannten Werkzeugen insofern, als dass es in Java geschrieben wurde, welches bis jetzt nicht Teil der Websuite ist. Java als Sprache wird in der MARS Arbeitsgruppe bis jetzt nur im ConfigService eingesetzt. Durch den Einsatz von Docker ist dies allerdings nicht als Problem anzusehen, da eine heterogen gestaltete Softwarearchitektur dem Microservices Ansatz entspricht.

Die Bibliothek basiert als einzige der genannten Lösungen nicht auf GDAL als Basis. GeoTools ist Teil der Open Source Geospatial Foundation (OSGeo).

Es verfügt neben einem Quickstart Guide und einer API Dokumentation, über einen User Guide und zahlreiche Programmbeispiele. Die Dokumentationen sind ausgesprochen ausführlich und sehr aktuell. Das Projekt ist aktueller als jedes der anderen genannten Bibliotheken.

5) *GeoServer*: Der GeoServer ist eine Verwaltungssoftware für Geodaten, basierend auf Java und PostGIS. Unterstützt werden eine Vielzahl von Raster-/ und Vektorformaten. Der GeoServer erlaubt neben der Verwaltung auch die Bearbeitung von Geodaten, sowie eine Darstellung mit Hilfe von OpenLayers. All diese Funktionalitäten lassen sich über eine Weboberfläche bedienen.

Mittels einer ausführlichen REST Schnittstelle lassen sich die Geoserver Abfragen in andere Anwendungen integrieren.

6) *GeoNode [Python]*: GeoNode ist keine Bibliothek, es handelt sich hierbei um ein Content Management System (CMS) für Geodaten, basierend auf dem Python Webframework Django.

Nach der Installation bietet GeoNode eine ansprechende Benutzeroberfläche, über die sich Raster-/ und Vektordaten importieren lassen. Auf Basis dieser Layer lassen sich Karten zusammenstellen.

GeoNode speichert die importierten Daten nicht direkt in eine Datenbank, sondern nutzt eine Instanz des GeoServers (siehe Abs. IV-A5). Obwohl der GeoServer ein Vielzahl von Formaten unterstützt erlaubt GeoNode derzeit (v2.3), nur den Import von Esri Shapefiles und GeoTIFF Dateien.

B. Ergebnis

Im Grundprojekt wurde eine Anforderungsanalyse durchgeführt. Auf Basis der ermittelten Anforderung wurde entschieden, dass der GeoServer zukünftig wieder für die Verwaltung von Geodaten verwendet werden soll. Des weiteren wird

für die Berechnungen von Geooperationen ein Microservice erstellt, der alle nötigen Geooperation durchführt.

1) *Datenspeicherung*: In der Vergangenheit wurde vom GeoServer Abstand genommen, Performance Messungen im Rahmen des Grundprojektes haben jedoch gezeigt, dass ein entsprechend optimierter Geoserver durchaus in der Lage ist große Mengen an Geodaten zu verwalten. Getestet wurde die Importgeschwindigkeit mit jeweils einer kleinen und einer großen Esri Ascii(.asc) Datei und einem Shapefile(.shp). Die Kleinen Dateien haben, komprimiert in einer .zip Datei eine ungefähre Größe von 400kB und die großen Dateien eine Größe von 500MB. Es ist zu beachten, dass die Importzeiten unter localhost durchgeführt wurden und Die Ergebnisse Mittelwerte sind.

	Shapefile	.asc Datei
kleine Datei	200 ms	70 ms
große Datei	8.800 ms	3.500 ms

Tabelle I. GeoServer Import Performance

2) *Datenverarbeitung*: Viele Arbeiten, wie die Konvertierung von Geodaten und die Generierung von Bilderpyramiden sind direkt über den GeoServer möglich. Für die weiterführende Verarbeitung und die inhaltliche Validitätsprüfung wird eine zentrale Komponente geschaffen, die lose an die Websuite gekoppelt ist. Diese Komponente besteht aus einer erweiterbaren Bibliothek auf Basis von GeoTools (siehe IV-A4).

V. HAUPTPROJEKT

Ziel des Hauptprojektes ist es, die spatiale Eingrenzung für alle Geodaten zu ermöglichen und die reduzierten Datensätze zu persistieren. Außerdem soll der GIS-Import für den GeoServer entwickelt werden. Des weiteren wird der Quad Algorithmus von Busch[3] in die Websuite eingebaut.

A. Spatiale Eingrenzung

Die Anforderungsanalyse im Grundprojekt hat ergeben, dass GeoTools für die Verarbeitung von GIS-Daten am besten geeignet ist. Im Hauptprojekt soll ein entsprechender Service erarbeitet werden, der die notwendigen Operationen durchführt.

B. GIS-Import

Der GIS-Import ist Teil des allgemeinen Imports (siehe II-B). Er ist dafür zuständig, Raster- und Vektordaten im GeoServer und der darunterliegenden PostGIS Datenbank zu speichern.

C. Quad Algorithmus

Bei der Speicherung von Rasterdaten wird jeder Datensatz als einzelne Datei abgelegt und es wird für den schnellen Zugriff im Web eine Mercatur Pyramide (siehe III-E) generiert. In bestimmten Anwendungsfällen ist es notwendig, dass Teilbereiche einer Rasterkarte referenziert werden und für den schnellen Zugriff abgelegt werden. Ein Anwendungsfall hierfür ist, dem Benutzer auswählbare Bereiche zur Verfügung zu stellen, welche die Ausdehnung der Simulation definieren.

Eine Möglichkeit ist, den gewünschten Ausschnitt aus dem Quellmaterial in höchster Auflösung heraus zu schneiden und eine neue Mercatur Pyramide zu generieren. Dies hat allerdings redundante Datenhaltung zur Folge. Besser ist es, QuadTiles aus der vorhandenen Bilder Pyramide zu referenzieren.

Der sog. Quad Algorithmus[3] berechnet auf Basis einer geometrischen Form (z.B. Shapefile oder GeoJSON) die minimale Menge an Quadtiles, die nötig sind, um die Fläche komplett darstellen zu können und dabei den minimalen Overhead zu erreichen. Siehe Abb. 4. Die zu speichernden Daten beschränken sich bei diesem Verfahren, auf ein Array von Quadkeys (siehe III-F).

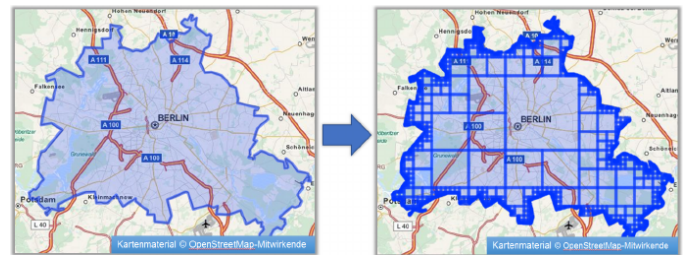


Abbildung 4. Umwandlung Shape in Quads am Beispiel Berlins. Quelle: [3]

Zur Reduktion des Rechenaufwandes wird zunächst die niedrigste Detailstufe errechnet, auf der die Berechnung begonnen werden kann. Hierzu wird von der obersten Ebene an, auf jeder Ebene überprüft, ob die gewählte Geometrie größer als ein Quadtile ist. Ist dies der Fall, so ist die aktuelle Ebene der Beginn des eigentlichen Algorithmus. In parallelen Threads werden alle Tiles der aktuellen Ebene in drei Kategorien eingeteilt:

- **Outside:** Quadtile liegt vollständig außerhalb
- **Within:** Quadtile liegt vollständig innerhalb
- **Intersects:** Quadtile liegt teilweise innerhalb / außerhalb

Tiles außerhalb des zu durchsuchenden Bereichs sind nicht teil der Ergebnismenge. Sie werden somit nicht weiter verarbeitet und die Rekursion wird abgebrochen.

Liegt ein Geometrisches Objekt vollständig in einer Kachel, so ist eine Tiefensuche ebenfalls nicht nötig. Das Tile wird der Ergebnismenge hinzugefügt.

Bei Kacheln der dritten Klasse, wird für die 4 tiefer gelegenen Kacheln jeweils in eigenen Threads rekursiv die gleiche Überprüfung durchgeführt.

VI. MASTERARBEIT - SPATIALER INDIKATOR

Ökosysteme sind sehr komplexe Gebilde mit einer Vielzahl von Einflussfaktoren. Diese Konstrukte zu Verstehen ist die Voraussetzung dafür, Folgeabschätzungen durchführen zu können und somit, den stetig voranschreitenden Klimawandel zu begrenzen.

Spatiale Indikatoren können bei der räumlichen Analyse eingesetzt werden, um den klimatische Veränderungen besser zu verstehen ([6] [7] [8]).

Die Arbeitsgruppe um Hagenlocher et al. [9] befasst sich mit den Ländern Westafrikas, da diese zu den am stärksten betroffenen, durch den Klimawandel gehören [10]. Um besonders bedrohte Gebiete, sog. Hotspots zu extrahieren und diese Bereiche mit anderen vergleichbar zu machen wurde der spatial gemischte Indikator (SGI) entwickelt. Dieser stellt eine Weiterentwicklung des SI dar und wird mit Hilfe von arithmetischen Berechnungen aus verschiedenen homogenisierten Subindikatoren berechnet.

Der SGI gibt einen räumlich abhängigen Wert an, der geographische Gebiete in Bezug auf eine bestimmte Kennzahl vergleichbar macht [11]. Zur Berechnung des SGI müssen die Datensätze zunächst wert-normalisiert und anschließend segmentiert(regionalisiert) und klassifiziert werden. Abb. 5 zeigt den allgemeinen Ablauf.



Abbildung 5. Spatialer Indikator. Quelle: [12]

A. Voraussetzungen

Für die Errechnung des SGI gelten bestimmte Prämissen, denen die Daten genügen müssen.

Der gemischte Indikator über ein Bereich wird aus verschiedenen Datensätzen errechnet. Eine korrekte Aussage kann nur getroffen werden, wenn die Daten den gleichen geographischen Bereich referenzieren.

Die Auflösung der Datensätze ist häufig unterschiedlich. Beispielsweise können Temperaturdaten stündlich aufgenommen sein und Niederschlagsdaten täglich. Eine entsprechende Inter- bzw. extrapolieren ist zwingend erforderlich.

Die einzelnen Daten müssen in GIS Formaten vorliegen. Hierbei handelt es sich um Formate, die neben den eigentlichen

Daten Metadaten abspeichern können. Gängige Formate sind: *.shp, *.asc oder GeoJSON.

Die Layer, auf denen sich die Informationen befinden müssen georeferenziert sein, es muss also in den erwähnten Metadaten hinterlegt sein, für welchen Bereich und ggf. Zeitraum die Daten gültig sind.

B. Vorbereitung

Es muss mittels GIS-Operationen eine gemeinsame Schnittmenge aus den, zur Verfügung stehenden Daten gebildet werden.

Diese Fläche wird in verschiedene, gleichgroße Quadranten aufgeteilt um später eine gleich gewichtete Aussage zu erhalten. Die maximale Feingranularität der Quadranten hängt von der Auflösung der Daten ab, eine zu feine Unterteilung führt nicht zur Verbesserung des Ergebnis, jedoch zu höherem Rechenaufwand.

Die Werte, der einzelnen Eingabeparameter müssen Wert-normalisiert werden. Üblich ist die Bildung des SGI über Farbwerte eines Graustufenbildes. Eine Normalisierung von 0 bis 255 ist hier der Regelfall. Abbildung 6 zeigt die Formel zur Normalisierung.

$$v' = \frac{v - \min}{\max - \min} (\max_{\text{norm}} - \min_{\text{norm}}) + \min_{\text{norm}}$$

Abbildung 6. Wertnormalisierung. Quelle: [9]

In Abhängig der Eingabewerte zueinander, ist es gegebenenfalls notwendig eine Gewichtung einfließen zu lassen. Diese ist Domain abhängig und kann nicht generell festgelegt werden.

Abschließend wird aus jedem Teilquadranten der SGI gebildet.

C. Durchführung

Die Bildung des SGI, mit den vorbereiteten Daten geschieht in drei Schritten: Segmentierung, Klassifizierung und Verschneidung.

1) *Segmentierung*: Nach Stepinski et al. [13] werden bei der Segmentierung kleine Teile eines Bildes auf Basis von homogenen Eigenschaften unterteilt. Für die Durchführung wird das Verfahren der objekt-basierte Bildanalyse (OBIA) [14] eingesetzt. Hierbei werden anhand von Farbwerten einzelne Objekte aus einem Bild herausgearbeitet, welche für die Klassifizierung verwendet werden.

2) *Klassifizierung*: Bei der Klassifizierung werden Grundinformationen der einzelnen Bereiche entfernt, um die Unterschiede von Bereichen besser heraus arbeiten zu können und Muster zu erkennen. Anhand von domainspezifischen Regeln werden Informationen herausgearbeitet.

Es kommen entsprechend des gewünschten Ergebnis andere Algorithmen zum Einsatz. Gängig sind die Klassifizierung durch Bestimmung der Häufigkeiten oder Durchschnittswerte der Farbwerte.

3) *Verschneidung*: Bei der Verschneidung wird aus den einzelnen Subindikatoren der Segmentierung und Klassifizierung, der SGI für einen Bereich errechnet. Hagenlocher et al. [9] nutzt hierfür den Cumulative climate change impact (siehe Abb. 7).

$$|CCCI| = \sqrt{SI_1^2 + SI_2^2 + SI_3^2}$$

Abbildung 7. CCI Formel. Quelle: [9]

Die Verschneidung von Rasterdaten erfolgt über die Normalisierung verschiedener Zellen. Hierbei wird, ähnlich wie beim Downsampling von Bildern in eine geringere Auflösung, jeweils der, am häufigsten vorkommende Wert benachbarter Zellen genommen. Dieser Wert wird klassifiziert und auf dessen Ergebnis wird die, oben genannten CCI Formel angewendet. Abb. 8 zeigt diesen Ablauf.

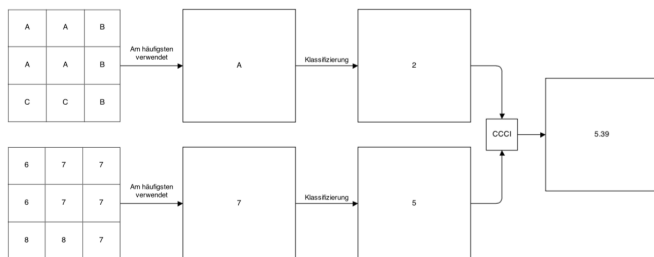


Abbildung 8. CCI Verschneidung. Quelle: [12]

4) *Ergebnis*: Mit dem oben beschriebenen Verfahren hat die Arbeitsgruppe um Hagenlocher [9] eine Karte zur Visualisierung von Klimarisiken in West-Afrika erstellt. Die SGIs der einzelnen Bereiche wurden entsprechend ihrer Werte zwischen 0=weiß über gelb als Zwischenwert bis 1=rot dargestellt. Zusätzlich wurde an den roten Hotspots die Art der Gefährdung in einem Tortendiagramm dargestellt. Siehe Abb. 9.

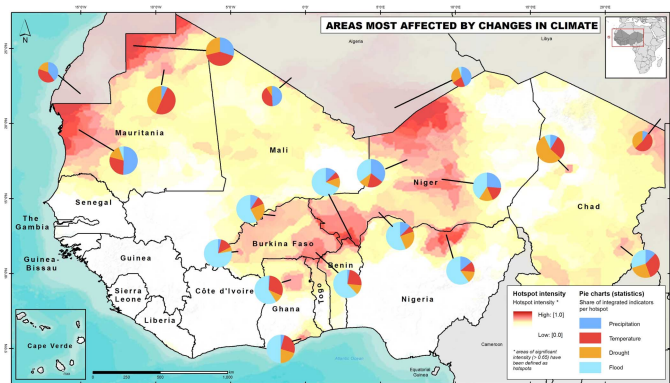


Abbildung 9. Klimahotspots. Quelle: [9]

VII. ZUSAMMENFASSUNG

In dieser Arbeit wurde der aktuelle Stand zur Forschung an webbasierte GIS Datenverarbeitung beschrieben.

Im Grundprojekt wurden die technischen Grundlagen erarbeitet und Designentscheidungen getroffen, welche für die zukünftige Arbeit als Grundlage dienen soll.

Das Hauptprojekt befasst sich mit der spatialen Problematik, insbesondere der Handhabung von räumlicher Eingrenzung unterschiedlich gearteter Formen. Hierzu ist der Quad Algorithmus ein Vielversprechendes Werkzeug.

Das Kapitel zur Masterarbeit behandelt den SGI, welcher eine Methode zur Evaluierung von klimatischen Ereignissen darstellt.

VIII. AUSBLICK

Das Grundprojekt ist nahezu abgeschlossen, an dessen Themen werden daher keine großen Veränderungen mehr vorgenommen.

Die Themen des Hauptprojektes und der Masterarbeit stehen noch nicht endgültig fest. Im speziellen das Thema der Masterarbeit erfordert weitergehende Recherche, bei der die wissenschaftliche Relevanz für die Informatik zu prüfen ist.

LITERATUR

- [1] I. P. on Climate Change, "Climate change 2007 - impacts, adaptation and vulnerability: Working group ii contribution to the fourth assessment report of the ipcc," 2008. [Online]. Available: http://www.ipcc.ch/pdf/assessment-report/ar4/wg2/ar4_wg2_full_report.pdf
- [2] Q. B. Le, S. J. Park, P. L. Vlek, and A. B. Cremers, "Land-use dynamic simulator (ludas): A multi-agent system model for simulating spatio-temporal dynamics of coupled human-landscape system. i. structure and theoretical specification," *Ecological Informatics*, vol. 3, no. 2, pp. 135 – 153, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1574954108000228>
- [3] J. Busch, "Ein spatio-temporales data warehouse für multiskalige, heterogene ökologische daten," Master's thesis, Hochschule für Angewandte Wissenschaften, 3 2015.
- [4] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1–9. [Online]. Available: <http://dx.doi.org/10.1007/BF00288933>
- [5] J. Schwartz, "Bing maps tile system," <https://msdn.microsoft.com/en-us/library/bb259689.aspx>, zuletzt besucht: 25.02.2016.
- [6] T. J. Cline, D. A. Seekell, S. R. Carpenter, M. L. Pace, J. R. Hodgson, J. F. Kitchell, and B. C. Weidel, "Early warnings of regime shifts: evaluation of spatial indicators from a whole-ecosystem experiment," *Ecosphere*, vol. 5, no. 8, pp. 1–13, 2014.
- [7] A. Costa, M. Madeira, J. L. Santos, T. Plieninger, and J. Seixas, "Fragmentation patterns of evergreen oak woodlands in southwestern iberia: Identifying key spatial indicators," *Journal of environmental management*, vol. 133, pp. 18–26, 2014.
- [8] K. Mattingly and J. Morrissey, "Housing and transport expenditure: socio-spatial indicators of affordability in auckland," *Cities*, vol. 38, pp. 69–83, 2014.
- [9] M. Hagenlocher, S. Lang, D. Holbling, D. Tiede, and S. Kienberger, "Modeling hotspots of climate change in the sahel using object-based regionalization of multidimensional gridded datasets," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 7, no. 1, pp. 229–234, 2014.
- [10] D. Hamro-Drotz, "Livelihood security: Climate change, migration and conflict in the sahel," *Conflict-sensitive adaptation to climate change in Africa*, p. 195, 2014.
- [11] L. Anselin, "Local indicators of spatial association—lisa," *Geographical analysis*, vol. 27, no. 2, pp. 93–115, 1995.
- [12] M. Baldowski, "Räumliche analyse mit spatialen indikatoren," Master's thesis, Hochschule für Angewandte Wissenschaften, 8 2014.

- [13] T. F. Stepinski, S. Ghosh, and R. Vilalta, "Automatic recognition of landforms on mars using terrain segmentation and classification," in *Discovery Science*. Springer, 2006, pp. 255–266.
- [14] G. Hay and G. Castilla, "Object-based image analysis: strengths, weaknesses, opportunities and threats (swot)," in *Proc. 1st Int. Conf. OBIA*, 2006, pp. 4–5.