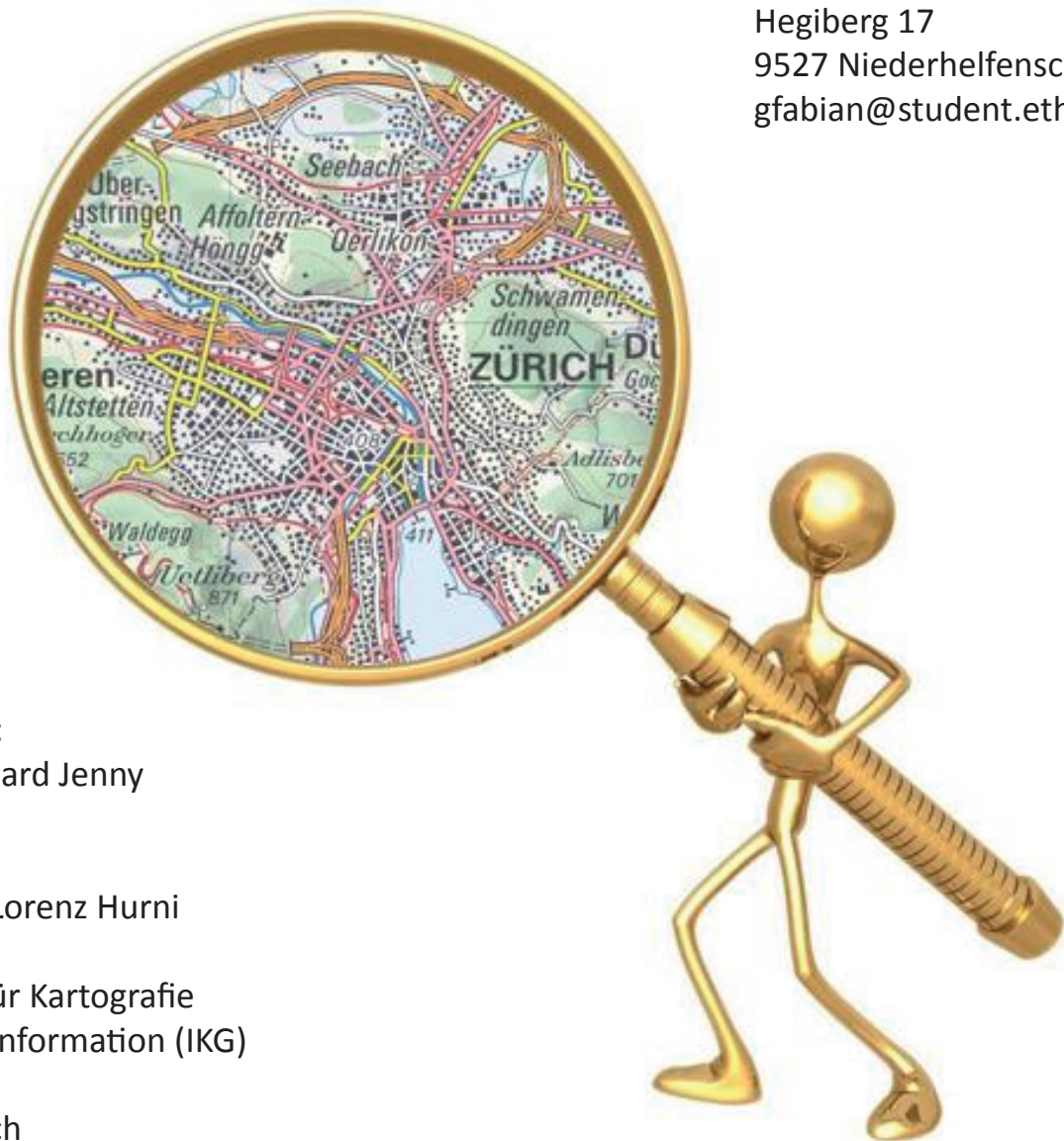


Adaptive Zooming

Autor:

Fabian Gschwend
Hegiberg 17
9527 Niederhelfenschwil
gfabian@student.ethz.ch



Betreuer:
Dr. Bernhard Jenny

Leitung:
Dr. Prof. Lorenz Hurni

Institut für Kartografie
und Geoinformation (IKG)

ETH Zürich

Quellen Titelbild

Lupe: <http://www.denkerfolg.eu/images/lupe.jpg>

Karte: <http://map.schweizmobil.ch/?lang=de>

I Vorwort

Der vorliegende Bericht dokumentiert meine Bachelorarbeit zum Thema „Adaptive Zooming“. Sie ist Teil des Bachelorstudiums Geomatik und Planung. Ausgeschrieben wurde die Arbeit vom Institut für Kartografie und Geoinformation (IKG), das zum Departements Bau, Umwelt und Geomatik (D-BAUG) der ETH Zürich gehört.

Für die Wahl des Themas „Adaptive Zooming“ gab es zwei Hauptgründe. Einerseits ist die Bedeutung der digitalen Karten stark gestiegen. Mit der rasanten Entwicklung von Computeranwendungen hat sich auch die Funktionalität von digitalen Karten in den letzten Jahren stark erhöht. Ausserdem ist die Verbreitung durch das Internet angewachsen. Webkarten wie Google Maps sind mittlerweile nicht nur für Kartografen kaum mehr wegzudenken. Andererseits ergab sich die Möglichkeit diese Arbeit als Programm umzusetzen. Aus der Erweiterung meiner Programmierkenntnisse werde ich auch nach dieser Arbeit einen Nutzen ziehen können.

An dieser Stelle möchte ich mich auch bei meinem Betreuer Bernhard Jenny für die Unterstützung bedanken. Er konnte mir vor allem zu Beginn der Arbeit wichtige Inputs zur Vorgehensweise geben. Ausserdem stand er jederzeit für Fragen zur Verfügung.

Fabian Gschwend

II Zusammenfassung

Unter „Adaptive Zooming“ versteht man das kontinuierliche Zoomen in digitalen Karten. Bei den meisten Webkarten, wie beispielsweise Google Maps, ist das kontinuierliche Zoomen nicht möglich. Es lassen sich nur eine bestimmte Anzahl Zoomstufen auswählen. Dabei ist für jede Zoomstufe die Karte als Bild gespeichert. Dadurch muss der Kartenausschnitt nicht bei jedem Aufruf neu berechnet, sondern nur geladen werden.

Das kontinuierliche Zoomen ist nur mit Vektordaten möglich. Aus diesen Geodaten wird die Karte fortlaufend neu berechnet. Das Hauptproblem liegt dabei, dass parallel zum Zoomwert auch die Generalisierung angepasst werden soll. Dadurch muss jeder Generalisierungszustand aus den Ausgangsdaten berechenbar sein. Als Ausgangsdaten für diese Bachelor-Arbeit wurden zwei Flusssdatensätze mit unterschiedlichem Detailgrad von Natural Earth verwendet. Das Ziel war, einen kontinuierlichen Übergang der Generalisierung zwischen den beiden Ausgangszuständen zu generieren.

Die Umsetzung erfolgte in Rahmen eines JavaScript-Programms. Es wurde jedoch kein neues Programm geschrieben, sondern ein bestehendes weiterentwickelt. Im Basisprogramm waren die Grundfunktionen zum Einlesen und Darstellen von Shapefiles bereits enthalten. Dadurch konnte man sich auf die Modellierung der kontinuierlichen Generalisierung konzentrieren.

Bei der Programmierung der neuen Funktionen war die Grundidee, den Aufbau der beiden Geometrien zu vereinheitlichen. Dazu mussten die Geometrien der beiden Datensätze umstrukturiert werden. Ausserdem wurden einige neue Punkte hinzugefügt. Das Resultat wurde als Webkarte dargestellt. Bei dieser Karte ist für den Anwender sowohl der Zoom, als auch die Generalisierung frei wählbar. Ausserdem können die beiden Werte miteinander verknüpft werden.

III Inhaltsverzeichnis

I	Vorwort	3
II	Zusammenfassung.....	4
III	Inhaltsverzeichnis.....	5
IV	Abbildungsverzeichnis	7
1	Einleitung.....	8
1.1	Einführung.....	8
1.2	Ausgangslage.....	8
1.3	Zielsetzung.....	9
1.4	Übersicht	10
2	Daten.....	11
3	Programm	13
3.1	Einführung.....	13
3.2	Basisprogramm.....	13
3.3	Programmablauf.....	14
3.3.1	Programmaufruf.....	16
3.3.2	Daten einlesen.....	17
3.3.3	Daten formatieren.....	17
3.3.4	Verarbeitung der Flussdaten	17
3.3.5	Karte rendern	18
4	Generierung der neuen Geometrie.....	20
4.1	Einführung.....	20
4.2	Flüsse verknüpfen	20
4.3	Aufbau der Geometrien vereinheitlichen	21
4.3.1	Datenstruktur	21
4.3.2	Vorbereitende Massnahmen	22
4.3.3	Abschnitte zuordnen	22
4.3.4	Orientierung untersuchen.....	24
4.3.5	Punkte zuordnen	26
4.3.6	Punkte positionieren: Detaillierter Datensatz.....	28

4.4	Geometrie interpolieren	29
5	Benutzeroberfläche	31
6	Resultatbeurteilung.....	32
6.1	Einführung.....	32
6.2	Beispiel Ganges	32
6.3	Fehleranalyse Flussdaten	33
6.4	Kompatibilitätstest mit Landesgrenzen	34
7	Ausblick	36
7.1	Zielerreichung.....	36
7.2	Weiterführende Ideen.....	36
7.3	Fazit	37
8	Literaturverzeichnis.....	38

IV Abbildungsverzeichnis

Abbildung 1: Relief (Quelle: Natural Earth, Mai 2011)	11
Abbildung 2: Vektordaten in drei Masstäben (Quelle: Natural Earth, Mai 2011)	11
Abbildung 3: Ablaufdiagramm	15
Abbildung 4: HTML-Code, Programmaufruf	16
Abbildung 5: Datenstruktur.....	21
Abbildung 6: Abschnitte zuordnen.....	23
Abbildung 7: Beispiel Nigerdelta	24
Abbildung 8: Orientierung, Fall 1	25
Abbildung 9: Orientierung, Fall 2	25
Abbildung 10: Orientierung, Fall 3	25
Abbildung 11: Orientierung, Spezialfall.....	26
Abbildung 12: Punkte zuordnen.....	27
Abbildung 13: Punkte zuordnen, Spezialfall splitten	28
Abbildung 14: Punkte positionieren 1.....	28
Abbildung 15: Punkte positionieren 2.....	29
Abbildung 16: Interpolation der neuen Geometrie	30
Abbildung 17: Screen-Shot, Programm.....	31
Abbildung 18: Resultatbeispiel Ganges, 3 Generalisierungsstufen	33
Abbildung 19: Fehler, Kreis	33
Abbildung 20: Fehler, Abschnitt ohne Zuordnung.....	33
Abbildung 22: Analysebeispiel Schweiz, 2 Generalisierungsstufen	34
Abbildung 21: Fehler, Abschnittszuordnung.....	34
Abbildung 23: Analysebeispiel Norwegen, 2 Generalisierungsstufen	35
Abbildung 24: Analysebeispiel Island, 2 Generalisierungsstufen.....	35

1 Einleitung

1.1 Einführung

Unter „Adaptive Zooming“ versteht man das kontinuierliche Zoomen in digitalen Karten. Beim Zoomen soll sich das Kartenbild nicht nur vergrössern, sondern auch detaillierter dargestellt werden. Bei vielen Webkarten, wie beispielsweise Google Maps, ist das Zoomen mit der Scroll-Funktion möglich und damit einfach zu handhaben. Dabei wird dem Anwender näherungsweise das Gefühl des kontinuierlichen Zooms vermittelt.

Früher oder später wird man sich aber über diese Zoomfunktion ärgern. Wenn man nämlich beim Scrollen nur noch zwischen einem zu grossen und einem zu kleinen Detailgrad wechseln kann, wird man feststellen, dass die Stärke der Generalisierung nicht frei gewählt werden kann. Das Zoomen ist nicht kontinuierlich, sondern nur stufenweise möglich. Der Grund dafür ist, dass der Kartenausschnitt beim Aufruf nicht neu aus den Geodaten berechnet wird. Stattdessen ist für jede Zoomstufe die ganze Karte als Bild gespeichert. Deshalb muss beim Verändern des Kartenausschnitts nur ein neuer Bildausschnitt geladen werden.

Es gibt auch Beispiele für Webkarten mit gespeicherten Bildern, bei welchen der Zoomwert frei gewählt werden kann. Beim GIS-Browser des Kantons Zürich kann der Anwender dafür den Kartenmassstab eingeben. Der angezeigte Kartenausschnitt ist dann eine massstäbliche Vergrösserung oder Verkleinerung eines der gespeicherten Kartenbilder. Damit erzielt man zwar eine kontinuierliche Vergrösserung, der Detailgrad bleibt aber trotzdem stufenartig.

1.2 Ausgangslage

Um ein kontinuierliches Zoomen zu ermöglichen, muss die Karte fortlaufend aus den Geodaten neu berechnet werden. Dafür braucht man Vektordaten. Betrachtet man einen Vektordatensatz in einem GIS-Programm wie ArcGis von ESRI, lässt sich dieser beliebig vergrössern und verkleinern. Die Linienbreiten und Punktgrössen werden dabei stetig dem gewählten Massstab angepasst. Der Detailgrad der Karte verändert sich jedoch nicht.

Die eigentliche Aufgabe zur Realisierung des kontinuierlichen Zooms ist also die Modellierung der kontinuierlichen Generalisierung aus einem oder wenigen Zuständen in den Vektordaten. Man beachte, dass nur Zustände generiert werden können, die stärker generalisiert sind als der detaillierteste Vektordatensatz.

Grundsätzlich reicht ein Vektordatensatz aus, um jeden ungenaueren Generalisierungszustand daraus generieren zu können. Dies geschieht durch die Anwendung von analytischen Algorithmen auf die Ausgangsdaten. Diese Methode wurde von A. Cecconi und M. Galanda beim Projekt

„Adaptive Zooming in Web Cartography“ verwendet. Dieses Projekt wurde im Bereich GIS im Departement Geografie der Universität Zürich durchgeführt.

Beim Vorgang der Generalisierung kombiniert man zwei Methoden. Zum Einen müssen mit zunehmendem Generalisierungsgrad Objekte weggelassen oder zusammengefasst werden, welche keinen Platz mehr haben. Das Entfernen eines Objektes selbst lässt sich natürlich nicht kontinuierlich darstellen. Dadurch, dass der Entfernungszeitpunkt für jedes Objekt einzeln berechnet wird, verschwinden jedoch nie viele Objekte auf einmal. Dies kann im gewissermassen auch als kontinuierlich bezeichnet werden. Die zweite Methode ist das Vereinfachen von Objekten mit zunehmendem Generalisierungsgrad. Das Vereinfachen wird durch das kontinuierliche Weglassen von Stützpunkten einer Linie realisiert. Abhängig vom Objekttyp werden unterschiedliche Algorithmen zur Umsetzung der einen oder beiden Methoden auf die Vektordaten angewendet. (Cecconi/Galanda, 2002)

Wie sich im Projekt „Adaptive Zooming in Web Cartography“ von Cecconi und Galanda gezeigt hat, liegt das Hauptproblem in der Berechnungsdauer des Kartenausschnitts. Je komplexer die Generalisierungsalgorithmen werden, desto länger wartet schlussendlich der Anwender vor dem Computer, bis er den gewünschten Kartenausschnitt zu sehen bekommt. Auf der anderen Seite sind aber gewisse Operationen zwingend notwendig, um die Anforderungen der Generalisierung zu erfüllen. (Cecconi/Galanda, 2002)

Ausserdem sei an dieser Stelle erwähnt, dass es auch Geodaten gibt, die nicht für die kontinuierliche Generalisierung geeignet sind. Bei einem Strassennetz beispielsweise, folgt der Aufbau einer Hierarchie. Dabei ist es absolut sinnvoll alle Strassen der gleichen Kategorie beim Selben Generalisierungszustand auszublenden. Deshalb wurde auch von Cecconi und Galanda bei einigen Objekten auf die stetige Neuberechnung verzichtet. Um diese Objekte trotzdem darstellen zu können, wurde ein ähnliches Prinzip angewendet wie bei den Karten mit abgestufter Zoomfunktion. Anstelle der ganzen Kartenbilder wurden einfach nur die Datenobjekte in einer gewissen Anzahl Detailstufen gespeichert. Wie beim GIS-Browser des Kantons Zürich werden je nach gewählter Generalisierungsstärke Vergrösserungen oder Verkleinerungen dieser Detailstufen angezeigt. (Cecconi/Galanda, 2002)

1.3 Zielsetzung

In dieser Arbeit wurde ein etwas anderer Ansatz für die Modellierung der kontinuierlichen Generalisierung gewählt. Anstatt die Generalisierungszustände aus einem Datensatz zu berechnen, wurden zwei Datensätze derselben Objektklasse, aber mit unterschiedlichem Detailgrad, als Ausgangsdaten gewählt.

Als Ziel wurde formuliert, den Bereich der Generalisierungsstärke zwischen den beiden Ausgangszuständen, kontinuierlich zu modellieren. Dazu sollen zwei Vektordatensätze mit Flussdaten verwendet werden. Die Umsetzung erfolgt durch die Programmierung einer Anwendung. Das Programm soll am Ende in der Lage sein, Vektordaten einzulesen und daraus eine Karte zu zeichnen. Damit die Karte die Anforderungen des kontinuierlichen Zooms erfüllt, soll die Zoomstufe frei wählbar sein und sich die Generalisierung der Flüsse stetig anpassen.

Abhängig von den Erkenntnissen, die dabei erlangt werden, soll das Programm auch auf andere, ähnliche Objektklassen angepasst werden. In Betracht gezogen wurden unter anderem Seen und Staatsgrenzen. Zusätzlich soll die Berechnung eines Vektorfeldes geprüft werden, das sich aus den Verschiebungen der Flusstützpunkte während des Generalisierungsvorgangs ergibt.

1.4 Übersicht

Dieser Abschnitt gibt einen kurzen Überblick über die Struktur dieses Berichts. Im Anschluss an diese Einleitung folgt im zweiten Kapitel die Beschreibung der gewählten Geodaten. Das dritte Kapitel enthält die Dokumentation des Programmablaufs beim ersten Aufruf. Dabei wird unter anderem auch aufgezeigt, welche Funktionen neu programmiert wurden. Der detaillierten Beschreibung dieser Funktionen wurde ein eigenes Kapitel gewidmet (Kap. 4). Der fünfte Abschnitt beschreibt die graphische Benutzeroberfläche. Dabei werden die Möglichkeiten des Anwenders zur Interaktion mit dem Kartenprogramm aufgezeigt.

Im sechsten Kapitel folgt die Beurteilung der erzielten Resultate. Es wird geprüft, ob die Anforderungen des kontinuierlichen Zooms erfüllt werden konnten. Zum Abschluss wird im siebten Abschnitt ein Fazit zu dieser Arbeit gezogen. Erwähnt werden mögliche Weiterentwicklungen des bearbeiteten Programms, aber auch des Projekts „Adaptive Zooming“ als Ganzes.

2 Daten

Als Grundlagedaten für die Umsetzung dieser Arbeit dienen Kartendaten von Natural Earth. Dabei handelt es sich um eine frei zugängliche Sammlung von Kartendaten der ganzen Welt. Diese können kostenlos und ohne Gebrauchseinschränkungen heruntergeladen werden. Natural Earth wurde ausschliesslich von Freiwilligen aufgebaut, wird aber unter anderem von der North American Cartographic Information Society (NACIS) unterstützt. Die Idee dahinter ist, jedem Kartografen kostenlose Grundlagedaten zur Verfügung zu stellen. (Natural Earth, 2011)



Abbildung 1: Relief (Quelle: Natural Earth, Mai 2011)

Die Auswahl an Kartendaten ist gross. Mehrheitlich sind es Vektordaten, die als Shapefile bereitgestellt werden. Angeboten werden Länder, Städte, Strassen, aber auch Seen, Küsten und Inseln, um nur einige Beispiele zu nennen. Zusätzlich stehen einige Reliefs als Rasterdaten (Abb. 1) zur Verfügung. Die meisten Geodaten decken die ganze Welt ab und können wie in der Abbildung Vektordaten (Abb. 2) beliebig zu digitalen Karten zusammengesetzt werden. Entsprechend ist auch der grösste verfügbare Kartenmassstab der Datensätze mit 1:10 Millionen relativ klein. Die beiden weiteren angebotenen Kartenmassstäbe sind 1:50 Millionen und 1:110 Millionen. (Abb. Vektordaten). Der Unterschied liegt dabei im Detailgrad der Datenobjekte. (Natural Earth, 2011)



Abbildung 2: Vektordaten in drei Massstäben (Quelle: Natural Earth, Mai 2011)

Dieser Unterschied wird bei der Modellierung des kontinuierlichen Übergangs ausgenutzt. Für die Programmierung werden die Flussdatensätze der Massstäbe 1:10 Millionen und 1:50 Millionen verwendet. Ausserdem kommen in dieser Arbeit die Länder-Datensätze in den verschiedenen Massstäben zum Einsatz. Die darin enthaltenen Staatsgrenzen dienen allerdings ausschliesslich zur Analyse der Übertragbarkeit des Programms auf andere Datensätze. Die Funktionen zur Modellierung des kontinuierlichen Übergangs werden nicht auf die Staatsgrenzen angepasst. Die entsprechenden Resultate findet man im Kapitel Resultatbeurteilung (Kap. 6).

Nur in einem Generalisierungszustand wird in der Karte zusätzlich die Landfläche angezeigt. Dazu wird der entsprechende Datensatz im Massstab 1:50 Millionen verwendet. Kombiniert mit dem blauen Hintergrund sorgt dieser dafür, dass eine Kartenansicht vermittelt wird. Gleichzeitig wird auch die Übersichtlichkeit verbessert.

Vorgängig zur Bearbeitung mit dem Programm, werden die Flussdaten einer kleiner Manipulation unterzogen. In ArcGis von ESRI werden die Flüsse mit gleichem Namen durch die „Dissolve“-Funktion zusammengefügt.

3 Programm

3.1 Einführung

Die Umsetzung dieser Arbeit erfolgt im Rahmen eines Programmes. Die Basisvoraussetzung an das Programm ist, aus Vektordaten eine Karte berechnen und darstellen zu können. Dies lässt bezüglich der Wahl der Programmiersprache natürlich eine Vielzahl von Möglichkeiten offen. Der Mangel an Programmierkenntnissen schränkt die Auswahl jedoch stark ein. Es werden deshalb nur zwei Varianten gegeneinander abgewogen.

Eine Möglichkeit ist die Umsetzung des Programms mit Java. Diese wäre aufgrund der vorhandenen Programmierkenntnisse zu favorisieren. Aus folgendem Grund fällt die Wahl jedoch auf die Alternative JavaScript: Für das Einlesen und Darstellen von Shapefiles gibt es ein frei verfügbares Programm in dieser Sprache. Für die Umsetzung wird also kein eigenes Programm geschrieben, sondern ein bestehendes weiterentwickelt. Die Grundvoraussetzungen, das Einlesen und Darstellen einer Karte aus Vektordaten, ist damit bereits durch das Basisprogramm erfüllt. Dadurch steht mehr Zeit für das eigentliche Problem, die Modellierung der kontinuierlichen Generalisierung zur Verfügung. Allerdings nimmt auch die Analyse eines fremden Programms gewisse Zeit in Anspruch.

JavaScript unterscheidet sich im Aufbau zwar stark von Java, hat aber zumindest den gleichen Syntax. Die Sprache wird vor allem für die Webprogrammierung eingesetzt und dient als Erweiterung zu HTML. JavaScript-Programme können direkt in dem HTML-Code eingegliedert werden. Abhängig von Benutzereingaben kann man damit eine Webseite dynamisch verändern. Auf Grund dieser Voraussetzungen ist es naheliegend das Resultat als Webkarte darzustellen. Für die Programmierung werden der Texteditor Notepad++ sowie der Firefox Debugger eingesetzt.

In diesem Kapitel wird die Entwicklung und der Ablauf des Programmes beim ersten Aufruf beschrieben. Die weiteren Anwendungsmöglichkeiten nach dem Laden der Karte, werden im Kapitel Benutzeroberfläche (Kap. 5) erklärt. Ausserdem wird den Funktionen zur Generierung der neuen Geometrie aus den beiden Flussdatensätzen ein eigenes Kapitel (Kap. 4) gewidmet.

3.2 Basisprogramm

Das Basisprogramm stammt von der Softwareplattform GitHub. Dabei handelt es sich um eine Plattform für den Austausch von Programmcode im Internet. Sie dient dem Weiterentwickeln von Softwareprojekten durch verschiedene Programmierer. Der Programmcode zahlreicher öffentlicher Projekte kann kostenlos von der Website heruntergeladen werden. Das zugehörige Versionsverwaltungssystem Git braucht man nur, wenn man den eigenen Programmcode in die Projektentwicklung mit einbringen möchte. Darauf wird im Rahmen dieser Arbeit aber verzichtet. (GitHub, 2011)

Das Programm findet man auf der Internetseite von GitHub unter dem Namen „RandomEtc/shapefile-js“. Es setzt sich zusammen aus sieben JavaScript-Files. Einige dieser Files sind nicht neu erstellt worden, sondern stammen aus anderen JavaScript-Bibliotheken. Das Basisprogramm gleicht dadurch einem Puzzle. Die auf GitHub neu programmierten Files dienen unter anderem dazu, die diversen Funktionen aus den Bibliotheken miteinander zu verknüpfen.

Im Grunde führt das Programm zwei Abläufe aus. Es können Shapefiles eingelesen und anschliessend die darin enthaltenen Objekte dargestellt werden. Allerdings werden nicht alle Shapetyps unterstützt. Die Funktionalität beschränkt sich auf die gewöhnlichen Punkt-, Linien- und Flächenobjekte in 2D. Dies stellt jedoch für diese Arbeit keine Einschränkung dar.

3.3 Programmablauf

Im Folgenden wird der Ablauf des bearbeiteten Programms beschrieben. Dieser unterscheidet sich natürlich an einigen Stellen vom Basisprogramm. Teilweise wurden bestehende Funktionen abgeändert. Es kommen aber auch einige neue Funktionen hinzu. Diese werden allesamt in die bereits existierenden JavaScript-Files integriert. Die neuen Funktionen könnten jedoch ohne Unterschied auch in einem neuen File abgelegt werden. Einen Überblick über den ganzen Ablauf gibt das Ablaufdiagramm für den ersten Programmaufruf (Abb. 3).

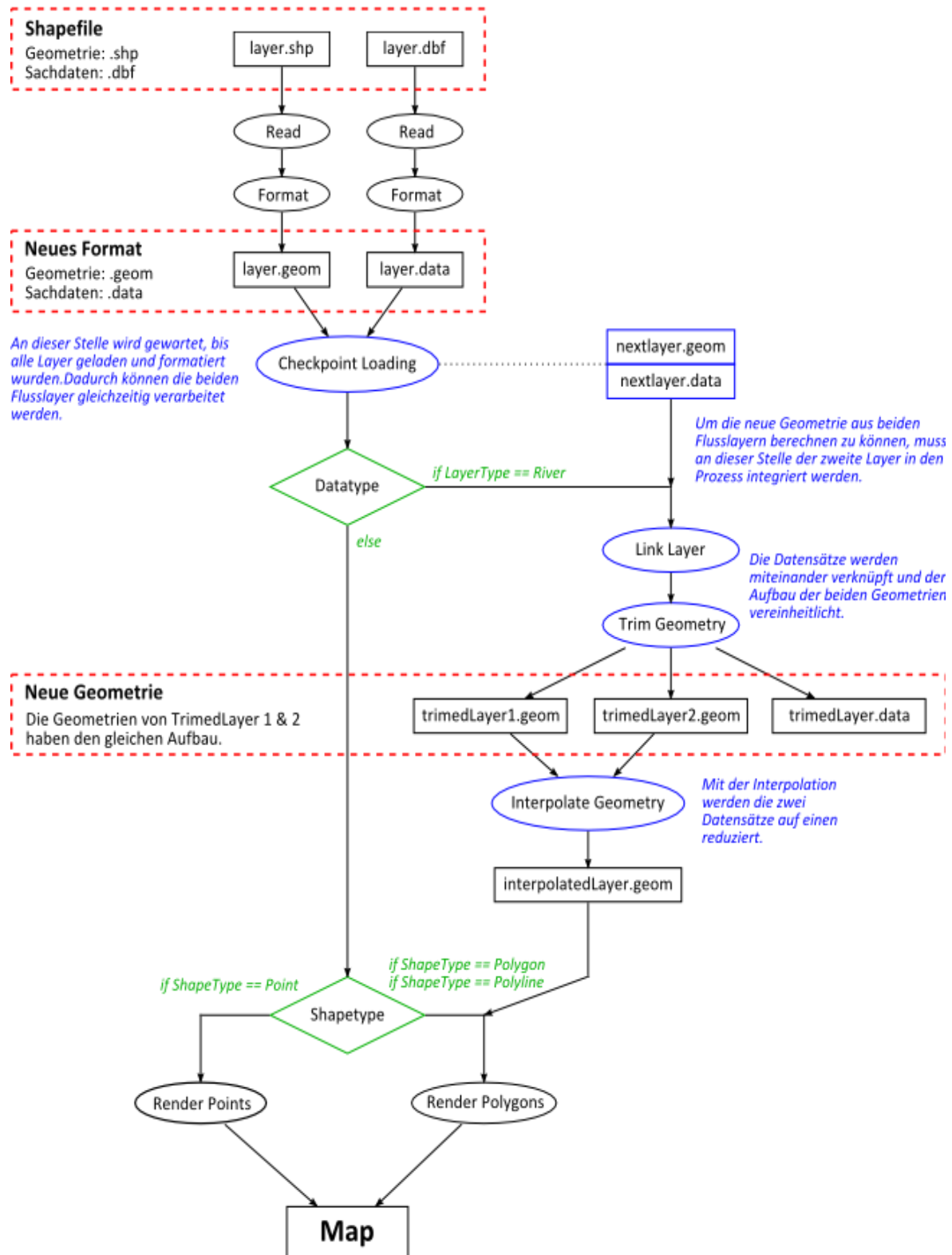


Abbildung 3: Ablaufdiagramm

3.3.1 Programmaufruf

Der Aufrufbefehl für die Berechnung der Karte erfolgt im HTML-Code der Webseite, auf der die Karte erscheinen soll. Einen Ausschnitt davon sieht man in Abbildung 4. Der Aufruf wird mit einem sogenannten „onLoad-Event“ verknüpft. Dieser sorgt dafür, dass das Programm erst ausgeführt wird, wenn man die HTML-Seite komplett geladen wurde. Andernfalls könnten dem Programm Input-Werte, wie beispielsweise die Position der Karte, fehlen. (PHP Performance, 2008)

Nachdem das Laden der Webseite abgeschlossen ist, wird die Funktion „Map“ aufgerufen. Damit das Programm weiss, mit welchen Daten es überhaupt arbeiten muss, müssen beim Aufruf der Funktion „Map“ Informationen zu den Datenlayern an das Programm übergeben werden. Wie man in Abbildung 4 erkennen kann, werden dazu sowohl der Dateipfad der Layer, sowie einige Angaben zur Darstellung der Objekte benötigt.

```
<script type="text/javascript">
window.onload = function() {
  var map = new Map('map', [
    // Layer Land
    new Layer('naturalearthdata/cultural/50m_land',
      {fillStyle:'#fadfa8',strokeStyle:'#5c3d1f',lineWidth:
        '0.5'},
      'NoRiver'),
    // Layer Rivers 1:50m
    new Layer('naturalearthdata/river/50m_r_d2',
      {strokeStyle:'blue',lineWidth:'0.5',textProp:'NAME'},
      'River'),
    // Layer River2 1:10m
    new Layer('naturalearthdata/river/10m_r_d',
      {strokeStyle:'blue',lineWidth:'0.5',textProp:'Name'},
      'River')
  ]);
}
</script>
```

Abbildung 4: HTML-Code, Programmaufruf

Die Information „textProp“ wird nur gebraucht, falls in der Karte, und damit auch im Programm, Sachdaten des Layers benötigt werden. Als Wert ist der Name des gewünschten Attributs anzugeben. Diesen findet man, indem man das Shapefile mit einem GIS-Programm öffnet. Im vorliegenden Fall benötigt man die Namen der beiden Flussdatensätze, um sie miteinander verknüpfen zu können. Darauf wird im vierten Kapitel genauer eingegangen. Ein anderer Verwendungszweck von Sachdaten ist beispielsweise die Beschriftung von Punktobjekten in der Karte.

Grundsätzlich können beliebig viele Datenlayer hinzugefügt werden. Bei den beiden Flussdatensätzen, welche mit dem Programm in einen Datensatz zusammengefasst werden sollen, ist zusätzlich als dritter Parameter das Schlüsselwort „River“ anzugeben. Bei den übrigen Layern kann ganz auf den dritten Parameter verzichtet werden. Der Übersicht halber wird in Abbildung 4 das Wort „NoRiver“ eingefügt. Um diese zusätzliche Information auch nutzen zu können, muss man im Basisprogramm die Funktion Layer anpassen.

Mit dem Aufruf der Funktion „Map“ setzt man den Ablauf zu Kartenberechnung in Gang gesetzt. Ausserdem werden sogenannte Eventlistener initialisiert. Damit ermöglicht man dem Anwender die Interaktivität. Dazu gehören beispielsweise die Möglichkeit die Zoomstufe zu verändern oder den Kartenausschnitt zu verschieben. Eine ausführliche Beschreibung dieser Anwendungen folgt in Kapitel Benutzeroberfläche (Kap. 5).

3.3.2 Daten einlesen

Um eine Karte berechnen zu können, werden zuerst die entsprechenden Daten benötigt. Die Datenlayer sind als Shapefiles gespeichert. Ein Shapefile besteht aus mehreren Dateien. Dazu gehören zwingend die Dateien mit den Endungen shp und dbf. Die shp-Datei dient zur Speicherung der Geometrie, während in der dbf-Datei die Sachdaten zu jedem Objekt hinterlegt sind. (ESRI, 1998)

Die shp- und dbf-Dateien werden separat für jeden Layer eingelesen. Der Zugriff auf die Daten erfolgt durch einen XMLHttpRequest. Dieser setzt das http-Protokoll zur Kommunikation mit den Shapefiles ein. Im folgenden wird dieser Ablauf kurz beschrieben.

Zuerst versendet das Programm eine leere Anfrage an die Datei, um zu überprüfen, ob diese auch wirklich existiert. Der Dateipfad ist aus dem Programmaufruf bekannt. Bekommt das Programm eine positive Antwort in Form einer Statusmeldung, wird die Hauptanfrage versendet. Dabei handelt es sich um eine „GET“-Anfrage. Als Antwort darauf erhält das Programm den Inhalt des angesprochenen Files. Eine positive Eigenschaft des XMLHttpRequest ist, dass das Programm nicht die Antwort auf eine Anfrage abwarten muss, um weiterarbeiten zu können. Dadurch können auch mehrere Anfragen gleichzeitig unterwegs sein. (Webmasterpro)

3.3.3 Daten formatieren

Nun sind die Daten im Programm gespeichert. Allerdings haben sie immer noch das gleiche Format wie im Shapefile. Um die Karte im Anschluss berechnen zu können, nimmt man an dieser Stelle eine Formatierung der binären Shapefiles vor. Diese erfolgt durch einige Funktionen, welche gegenüber dem Basisprogramm nicht verändert wurden. Im Anschluss sind die Koordinaten der Punkte als Dezimalzahlen gespeichert und die Sachdaten sind als gewöhnliche Zeichenketten verfügbar. Die Punkte ihrerseits sind in einer Arraydatenstruktur abgelegt. An der Geometrie selbst ändert die Formatierung nichts.

3.3.4 Verarbeitung der Flussdaten

Im Basisprogramm wird das Rendern für jeden Layer einzeln in Gang gesetzt, sobald die Formatierung abgeschlossen wird. Die Flussdaten können jedoch erst gerendert werden, nachdem sie in einem Datensatz zusammengefasst sind. Dafür muss eine Möglichkeit geschaffen werden um zwei Layer gleichzeitig verarbeiten zu können.

Zuerst muss sichergestellt werden, dass auch beide Flussdatensätze bereits zur Verfügung stehen. Dafür wird eine neue Funktion eingefügt welche überprüft, ob die Formatierung aller Layer abgeschlossen ist. Diese Funktion ist im Ablaufdiagramm (Abb. 3) mit dem Ausdruck „Checkpoint

Loading“ vermerkt. Eine weitere Funktion ermöglicht den gleichzeitigen Zugriff auf die Daten zweier Layer, was im Basisprogramm ebenfalls nicht vorgesehen ist.

Weil von diesen Massnahmen nur die Flussdatensätze betroffen sind, müssen sie von den übrigen Daten separiert werden. Der erste Flussdatensatz wird durch den Parameter Datatype (vgl. Abb. 3), der beim Programmaufruf mit dem Schlüsselwort „River“ vermerkt ist. Anstatt den Flussdatensatz selbst zu rendern, wird in diesem Fall der zweite Flussdatensatz aufgerufen.

Im Anschluss wird der Aufbau der Geometrie in den beiden Datensätzen vereinheitlicht. Konkret heisst das, dass für jeden Punkt des einen Datensatzes, ein neuer Punkt im anderen Datensatz hinzugefügt wird. Die Schwierigkeit besteht vor allem darin, diese Punkte in beiden Datensätzen in die gleiche Reihenfolge zu bringen, um sie nachher zu einem Datensatz interpolieren zu können. Die Funktionen zur Verknüpfung, Vereinheitlichung und Interpolation der Flussdaten werden in Kapitel 4 erläutert.

3.3.5 Karte rendern

Der weitere Ablauf kann am einfachsten mit dem aus dem Englischen kommenden Wort „rendern“ beschrieben werden. In Bezug auf ein Bild oder eine Karte heisst es so viel wie berechnen und darstellen. Berechnet wird beim Rendern die Lage der Punkte im abgebildeten Kartenausschnitt, welche am Ende für die Darstellung entscheidend ist. Der Kartenausschnitt kann später vom Anwender frei gewählt werden. Die Darstellung erfolgt gemäss den Angaben beim Programmaufruf (Abb. 4). Bei Linien kann man beispielsweise die Strichdicke und die Linienfarbe angeben.

Das Rendering-Prozess ist nicht für alle Objekttypen gleich. Für jeden Datensatz wird unverändert gegenüber dem Basisprogramm geprüft, welchen Shape-Type die Datenobjekte haben. Dabei wird nur zwischen Punkte- und Linienobjekten unterschieden. Die Flächenobjekte werden gleich gerendert wie die Linienobjekte, mit dem Unterschied, dass sie im Anschluss noch mit Farbe ausgefüllt werden können. Beim neu generierten Flussdatensatz kann man auf die Überprüfung des Shape-Typs verzichten. Für die Flüsse wird direkt die Render-Funktion für Polylines aufgerufen.

Das Beschriften ist nur bei Punktobjekten möglich. Grundsätzlich können die Punkte mit allem beschriftet werden, was mit den Sachdaten eingelesen wurde. Bei Städten kann beispielsweise die Einwohnerzahl angegeben werden, falls diese in der eingelesenen dbf-Datei enthalten ist. Weil sich die Ausgabe aber auf ein Attribut beschränkt, ist in den meisten Fällen nur die Beschriftung mit dem Namen sinnvoll.

Alle Sachdaten, die nicht auf der Karte erscheinen sollen, müssten eigentlich auch nicht geladen werden. Nur die Objektdaten des gewünschten Attributs zu laden, anstelle der ganzen dbf-Datei, dürfte allerdings schwierig umzusetzen sein. Mit beträchtlich geringerem Aufwand umzusetzen wäre, ganz auf das Laden der dbf-Datei eines Layers zu verzichten, falls keine Beschriftung erfolgen soll. Insbesondere bei allen Datensätzen die keine Punktobjekte enthalten und dadurch nicht beschriftet werden können. Eine Ausnahme bilden dabei die Flussdatensätze, welche mit Hilfe der Sachdaten aus der dbf-Datei miteinander verknüpft werden.

Für das Zeichnen der Geometrie wird das HTML-Element Canvas verwendet. Das Canvas-Element ist Teil von HTML5, der neusten HTML-Version. Entsprechend ist die Darstellungstechnik noch relativ neu. Es wird jedoch mittlerweile von den aktuellen Versionen aller gängigen Browser unterstützt. Kombiniert mit JavaScript können damit Bilder generiert und dynamisch verändert werden. Für das Beispiel der Webkarte heisst das konkret, dass man den Kartenausschnitt aus den Vektordaten fortlaufend neu gezeichnet werden kann. (Flanagan, 2011)

Eine Erweiterung des Canvas-Elements ist WebGL. Während sich das Basis-Canvas auf den zweidimensionalen Raum beschränkt, kann man mit WebGL auch 3D-Graphiken rendern. Dadurch wird die Technik auch für die Game-Industrie interessant. Eine Alternative zu Canvas ist das bereits etwas ältere Scalable Vector Graphics (SVG). SVG kann man wie Canvas mit Hilfe von JavaScript Vektorgraphiken in den neuen Browserversionen darstellen und animieren. (Braun, 2011)

4 Generierung der neuen Geometrie

4.1 Einführung

Die Funktionen zur Generierung der neuen Flussgeometrie sind als Hauptteil dieser Arbeit zu betrachten. Deshalb wird ihnen hiermit ein eigenes Kapitel gewidmet. Sie modellieren den kontinuierlichen Übergang zwischen den Generalisierungsstufen der beiden Flussdatensätze. Der Ablauf ist grob in die folgenden drei Abschnitte unterteilt. Zuerst werden die gleichnamigen Flüsse miteinander verknüpft. Danach wird, unter anderem durch das Einfügen von zusätzlichen Punkten, der Aufbau der beiden Geometrien vereinheitlicht. Abschliessend fasst das Programm die beiden einheitlich aufgebauten Geometrien zu einem Datensatz zusammen. Bei der Beschreibung dieser Funktionen wird auf die Verwendung von Programm-Code verzichtet.

Die Idee hinter diesem Ablauf ist vor allem die Minimierung der Wartezeiten für den Benutzer. Denn sowohl die Verknüpfung der Flüsse, als auch die Vereinheitlichung des geometrischen Aufbaus der beiden Datensätze, muss nur beim ersten Programmaufruf durchgeführt werden. Dadurch hat das Programm nur beim ersten Aufruf einen grossen Berechnungsaufwand.

Damit die beiden Geometrien mit vereinheitlichtem Aufbau nur einmal berechnet werden müssen, haben sie auch einige Anforderungen zu erfüllen. Aus diesen beiden Datensätzen soll mit der Interpolation jede beliebige Generalisierungsstufe zwischen den beiden Ausgangszuständen modelliert werden können. Dies muss bei der Funktion zur Vereinheitlichung der Geometrie berücksichtigt werden.

4.2 Flüsse verknüpfen

Um die gleichen Objekte in beiden Datensätzen miteinander verknüpfen zu können, müssen sie bei einem Attribut der Sachdaten den gleichen Wert aufweisen. Weil bei den gewählten Flussdatensätzen von Natural Earth jeweils der Name gespeichert ist, wird dieser als Verknüpfungsattribut gewählt. Bei anderen Ausgangsdaten eignen sich oftmals auch eine ID oder eine Nummer. Weil die Beschriftung nur bei Punktobjekten möglich ist, werden die Sachdaten nach dem Verknüpfen der Flüsse nicht mehr gebraucht.

Im detaillierteren Datensatz sind deutlich mehr Flüsse enthalten als im anderen. Flüsse, die nur in einem Datensatz vorkommen, können für die neue Geometrie nicht genutzt werden. Der Grund ist, dass nur eine Generalisierungsstufe bekannt ist. Deshalb werden sie an dieser Stelle gelöscht. Denkbar ist auch die Verschiebung dieser Daten in einen neuen Datensatz. Dadurch sind sie, im Gegensatz zu den anderen Flüssen, aber nur in einem statischen Generalisierungszustand gespeichert. Allerdings könnte man sie beispielsweise bei der höchsten Generalisierungsstufe zusätzlich einblenden.

4.3 Aufbau der Geometrien vereinheitlichen

Um die gewünschte Generalisierungsstufe im Anschluss interpolieren zu können, muss jeder Stützpunkt der Flussgeometrie in beiden Datensätzen enthalten sein. Darum müssen in beiden Datensätzen zusätzliche Punkte eingefügt werden. Dies ist jedoch nur möglich, wenn der Aufbau der beiden Datensätze identisch ist.

Die Funktion zum Aufbau der Geometrien beinhaltet einige berechnungsintensive Vorgänge. Sie muss jedoch nur einmalig, beim ersten Programmaufruf, ausgeführt werden. Anschliessend werden die neu generierten Geometrien gespeichert und können wiederholt für die Interpolation der verschiedenen Generalisierungsstufen aufgerufen werden. Zum besseren Verständnis dieser Funktion wird zuerst die Datenstruktur erläutert.

4.3.1 Datenstruktur

Die Struktur entspricht der gängigen Art, ein Netzwerk zu modellieren. Jeder Fluss besteht aus einem oder mehreren Abschnitten. Ein Flussabschnitt seinerseits setzt sich aus mindestens zwei Stützpunkten zusammen. Soweit trifft dies auf beide Datensätze zu. Anhand eines kleinen Beispiels (Abb. 5) kann man aber erkennen, dass die beiden Datensätze trotzdem einige strukturelle Unterschiede haben.

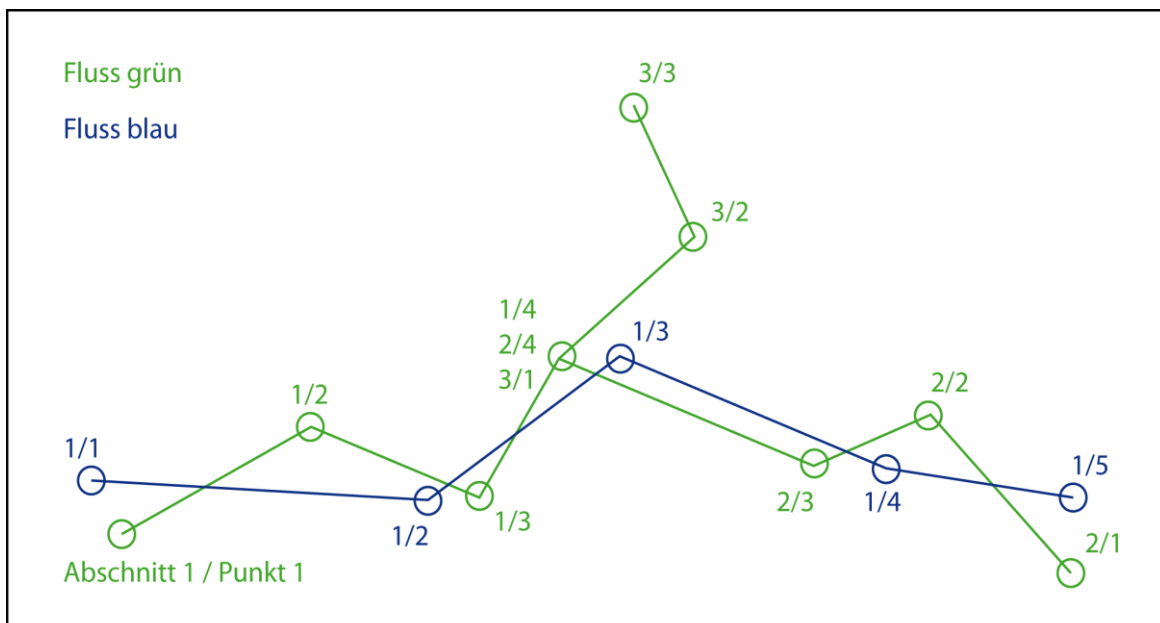


Abbildung 5: Datenstruktur

Der detailliertere Fluss ist grün dargestellt, der ungenauere blau. Dies wird in der weiteren Beschreibung der Funktionen beibehalten. Die Punkte sind jeweils mit ihrer Abschnitt- und Punktnummer beschriftet. Wie man sieht, besteht der grüne Fluss aus drei Abschnitten, während die blauen Punkte alle im gleichen Abschnitt gespeichert sind. Der Punkt in der Mitte ist im grünen Datensatz in allen drei Abschnitten enthalten. Ausserdem ist im rechten Teil des Beispiels die Punktfolgenfolge gegenläufig. Ein weiteres Problem ist, dass der dritte Abschnitt des grünen Flusses im blauen gar nicht vorhanden ist. Um einen einheitlichen Aufbau der beiden Datensätze zu ermöglichen, müssen diese Unterschiede behoben werden. Dazu müssen unter anderem

Abschnitte neu unterteilt, sowie bei einigen die Punktreihenfolge umgekehrt werden. Diese Massnahmen betreffen hauptsächlich den grünen Flussdatensatz. Das heisst, dass die Vereinheitlichung des Aufbaus durch die Anpassung des grünen Datensatzes an den blauen realisiert wird.

An dieser Stelle sei noch erwähnt, dass das Beispiel in Abbildung 5 einen eher komplizierten Fall darstellt. Ein grosser Teil der Flüsse hat keine Verzweigungen oder besteht sogar in beiden Datensätzen jeweils nur aus einem Abschnitt. Grundsätzlich tritt das Problem von Verzweigungen nur bei Flussdeltas auf. Beim Zusammenfluss von zwei Flüssen haben diese in der Regel nicht die gleichen Namen. Trotzdem muss das Programm natürlich auch mit den aussergewöhnlichen Fällen zurechtkommen.

4.3.2 Vorbereitende Massnahmen

Bevor mit der Zuordnung der Flussabschnitte begonnen wird, werden alle Abschnitte welche nur aus zwei Stützpunkten bestehen, bearbeitet. Diese Massnahmen betreffen beide Datensätze. Für die Bearbeitung nimmt das Programm eine Unterteilung in zwei Kategorien vor. Abschnitte mit einer Ausdehnung von weniger als 0.005 Grad werden für eine sinnvolle Darstellung als zu klein betrachtet. Deshalb werden sie gelöscht.

Bei den längeren Abschnitten aus zwei Stützpunkten werden zwei zusätzliche Punkte auf der Verbindung der beiden ursprünglichen Punkte eingefügt. Diese teilen den Abschnitt in drei gleich lange Teile. Durch die zusätzlichen Punkte wird die nachfolgende Abschnittszuordnung vereinfacht. Auf die Darstellung des Abschnitts haben diese Punkte keinen Einfluss.

4.3.3 Abschnitte zuordnen

Bei der Zuordnung wird für jeden Flussabschnitt des detaillierteren grünen Datensatzes der passende Abschnitt im blauen gesucht. Dafür werden als erstes im grünen Flussabschnitt zwei Zuordnungsknoten definiert. Die Wahl der Punkte ist abhängig von der Anzahl der Stützpunkte im betreffenden Abschnitt. Besteht ein Abschnitt aus 30 oder mehr Punkten, markiert man die Punkte zehn und zwanzig als Zuordnungsknoten. Bei kürzeren Abschnitten wird der erste Zuordnungsknoten nach einem Drittel, der Zweite nach zwei Dritteln der Anzahl Stützpunkte, gewählt. Anschliessend werden vom ersten Zuordnungsknoten die Distanzen zu den blauen Punkten des gleichen Flusses berechnet. Dadurch lassen sich die drei nächstliegenden blauen Punkte ermitteln.

Wie man in Abbildung 5 gesehen hat, gibt es aber nicht immer zwei Datensätze, die komplett zueinander passen. In diesem Fall wählt das Programm den blauen Abschnitt aus, der am besten zum Anfang des grünen passt. Deshalb ist es sinnvoll den wichtigeren ersten Zuordnungsknoten nahe am Anfang des Abschnitts zu wählen. Die Wahl des ersten Punktes als Zuordnungsknoten ist jedoch nicht geeignet. Der Grund ist, dass bei Flüssen mit mehreren Abschnitten teilweise Verzweigungen auftreten. Bei Verzweigungen liegen Punkte dreier Abschnitte in unmittelbarer Nähe zueinander oder sogar am gleichen Ort. Deshalb kann der Fall auftreten, dass drei Punkte von drei verschiedenen Abschnitten als nächste Punkte ermittelt werden. Mit dieser Information alleine ist das Programm nicht in der Lage den grünen Abschnitt eindeutig zuzuordnen.

Dadurch, dass nicht der erste Punkt als Zuordnungsknoten gewählt wird, lassen sich die meisten dieser Fälle umgehen. Bei den meisten Abschnitten gehören die drei nächsten blauen Punkte sogar allesamt zum gleichen Abschnitt. Ein solches Beispiel sieht man in Abbildung 6. Dabei werden die drei orange markierten Punkte X1-X3 als nächste Punkte gefunden. Sobald die nächsten zwei Punkte zum gleichen blauen Abschnitt gehören, ordnet das Programm diesen blauen Abschnitt definitiv dem grünen zu. In Beispiel Abschnitte zuordnen wird somit der grüne Abschnitt 1 dem blauen Abschnitt zwei zugeordnet, ohne dass der zweite Zuordnungsknoten zum Einsatz kommt.

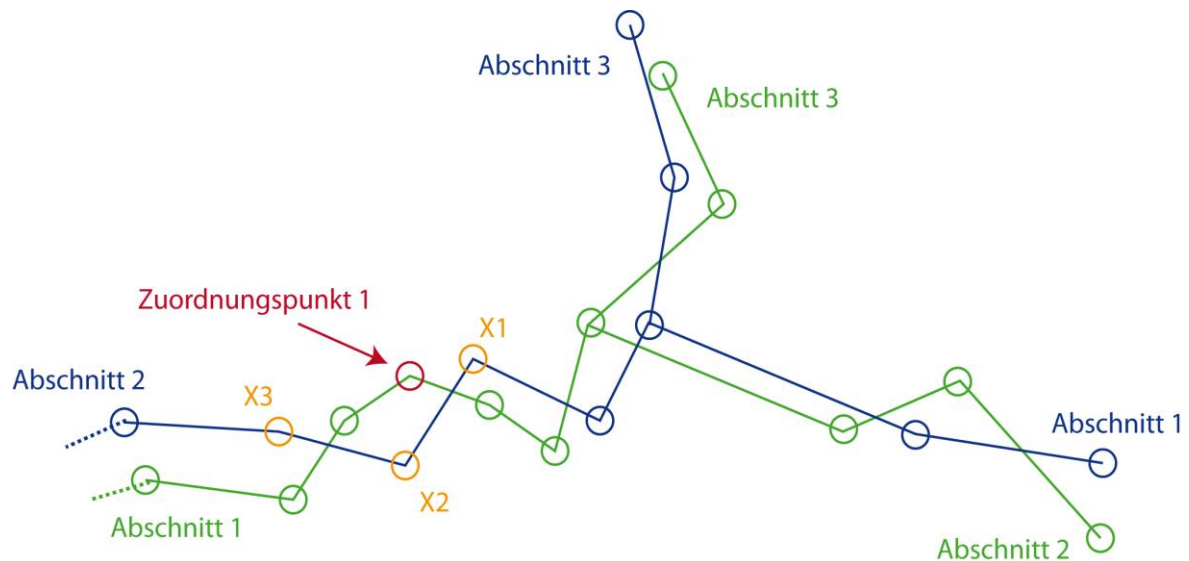


Abbildung 6: Abschnitte zuordnen

Der zweite Zuordnungsknoten wird nur verwendet, wenn die beiden nächsten Punkte nicht zum gleichen Abschnitt gehören. Dabei wird wieder mit der Berechnung von Distanzen die Nähe der Punkte zum Zuordnungsknoten untersucht. Infrage kommen diesmal allerdings nur die Punkte jener Abschnitte, zu welchen die nächsten drei Punkte des ersten Zuordnungsknotens gehören. Ausserdem reicht diesmal die Ermittlung des allernächsten Punktes aus. Der Abschnitt auf dem dieser Punkt liegt, wird ausgewählt.

Ein Beispiel, das den Einsatz des zweiten Zuordnungsknoten erfordert, sieht man in Abbildung 7. Das Problem ist, dass die Verzweigung beim grünen Datensatz viel weiter südlich liegt als beim blauen. Dadurch tritt bei diesem Beispiel genau der Fall ein, dass die drei Startknoten der blauen Abschnitte am nächsten beim ersten Zuordnungspunkt liegen. Dank der Registrierung aller drei Punkte, kann man wenigstens sicher sein, dass auch der gesuchte dabei ist. Der richtige Abschnitt wird anschliessend wie oben beschrieben mit dem zweiten Zuordnungsknoten ermittelt.

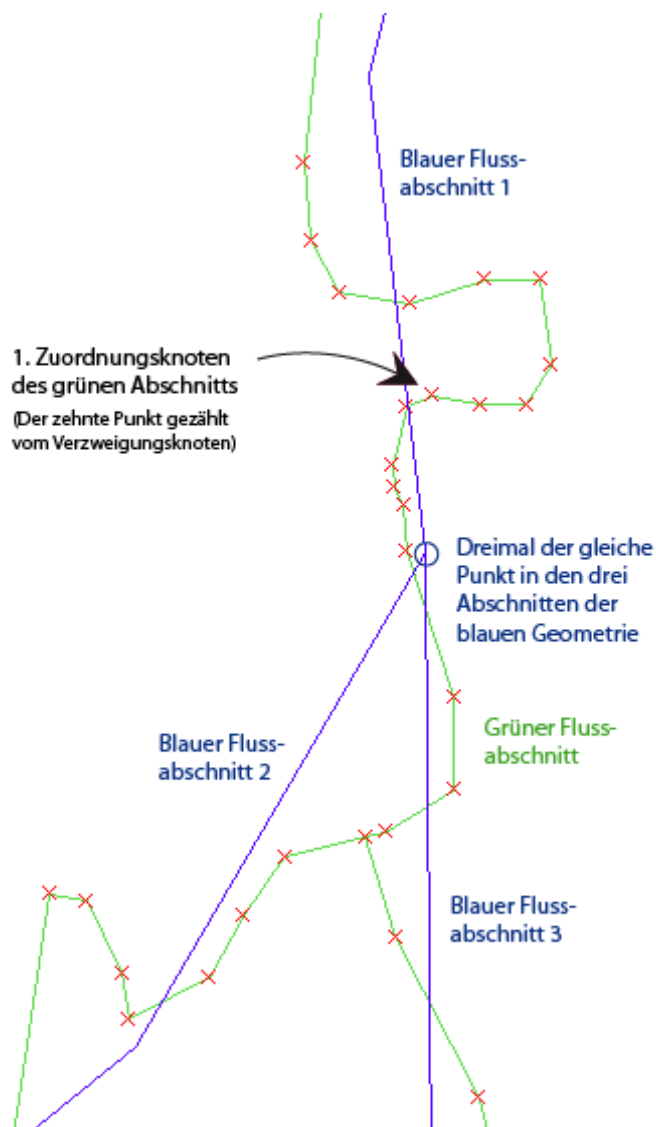


Abbildung 7: Beispiel Nigerdelta

4.3.4 Orientierung untersuchen

Bevor mit der Zuordnung der einzelnen Punkte begonnen werden kann, ist die Überprüfung der Orientierung notwendig. Mit der Orientierung ist in diesem Fall die Nummerierung der Punkte gemeint. Diese ist entweder in oder gegen die Fließrichtung definiert. Die Fließrichtung selbst spielt keine Rolle. Es ist nur wichtig, dass die Orientierung bei beiden gleich ist. Dazu geht man wie folgt vor.

Zuerst sucht man für den ersten Punkt des grünen Abschnitts den nächsten Punkt im zuvor zugeordneten blauen Abschnitt. Dieser Punkt wird mit X markiert. Anschliessend werden für den zweiten Punkt des grünen Abschnitts die Distanzen zum Punkten X, sowie zu dessen Nachbarpunkten X-1 und X+1, berechnet. Die Fortsetzung ist abhängig davon, welcher dieser drei Punkte am nächsten beim Punkt 2 liegt.

In Abbildung 8 sieht man ein Beispiel, bei dem der Punkt X+1 am nächsten bei P2 liegt. In diesem Fall sind die beiden Abschnitte gleich orientiert. Somit kann mit der Punktzuordnung weitergefahren werden.

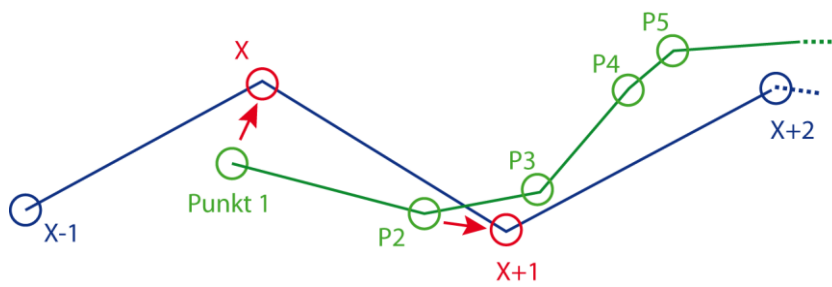


Abbildung 8: Orientierung, Fall 1

Ist die Distanz von X-1 zum Punkt 2 am kürzesten, ist die Orientierung gegenläufig (Abb. 9). In diesem Fall wird die Reihenfolge der Punkte des grünen Abschnitts umgekehrt. Anschliessend wird die Abschnittszuordnung wiederholt. Wie bereits erwähnt ist bei einer nicht eindeutigen Zuordnung der Abschnittsanfang entscheidend. Dieser liegt nun aber auf der anderen Seite des Abschnitts.

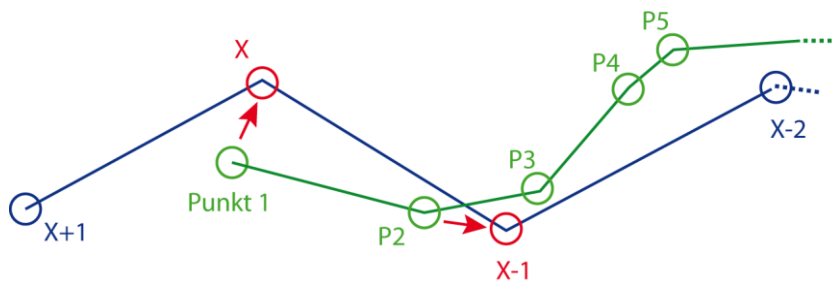


Abbildung 9: Orientierung, Fall 2

Die dritte Möglichkeit ist, dass wie in Abbildung 10 der Punkt X nicht nur am nächsten bei Punkt 1, sondern auch beim zweiten Punkt liegt. Tritt dieser Fall ein, ist die Orientierung weiterhin unbekannt. Entsprechend fährt das Programm mit dem nächsten Punkt des grünen Abschnitts fort. Erneut werden die Distanzen zu den drei Punkten des blauen Abschnitts berechnet und miteinander verglichen. Gleichermassen geht es weiter, bis die Frage nach der Orientierung geklärt ist. Dieser Fall tritt in Abbildung 10 beim Punkt 4 ein. An dieser Stelle liegt erstmals der X+1 näher beim aktuell bearbeiteten grünen Punkt als X. Damit sind die beiden Abschnitte gleich orientiert und die Punktzuordnung kann beginnen.

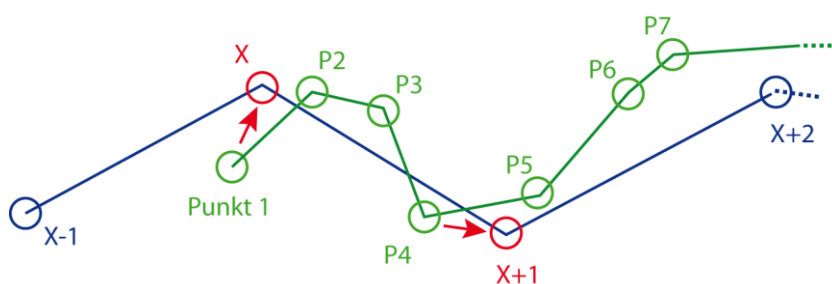


Abbildung 10: Orientierung, Fall 3

Wie bei der Abschnittszuordnung, muss auch bei der Untersuchung der Orientierung das Programm zusätzlich erweitert werden um einige Spezialfälle abzufangen. Es kann zum Beispiel eintreffen, dass alle Punkte eines grünen Abschnitts am nächsten beim blauen Startpunkt X liegen. Dieser Fall wird beim dritten grünen Abschnitt des Einführungsbeispiels (Abb. 5) eintreten. Es ist offensichtlich, dass kein blauer Abschnitt im Datensatz enthalten ist, der diesem grünen Abschnitt entspricht. Deshalb werden diese grünen Abschnitte gelöscht.

Ein weiterer Spezialfall ist bei den zu drehenden Abschnitten zu betrachten. Weil das Programm den grünen Abschnitt in der Abbildung 11 dem ersten blauen Abschnitt zuordnet (vgl. Kap 4.3.3), ist die Orientierung gegenläufig. Wie bereits erwähnt, wird in diesem Fall die Punktreihenfolge umgekehrt. Dieser Zustand ist in Abbildung 11 sichtbar.

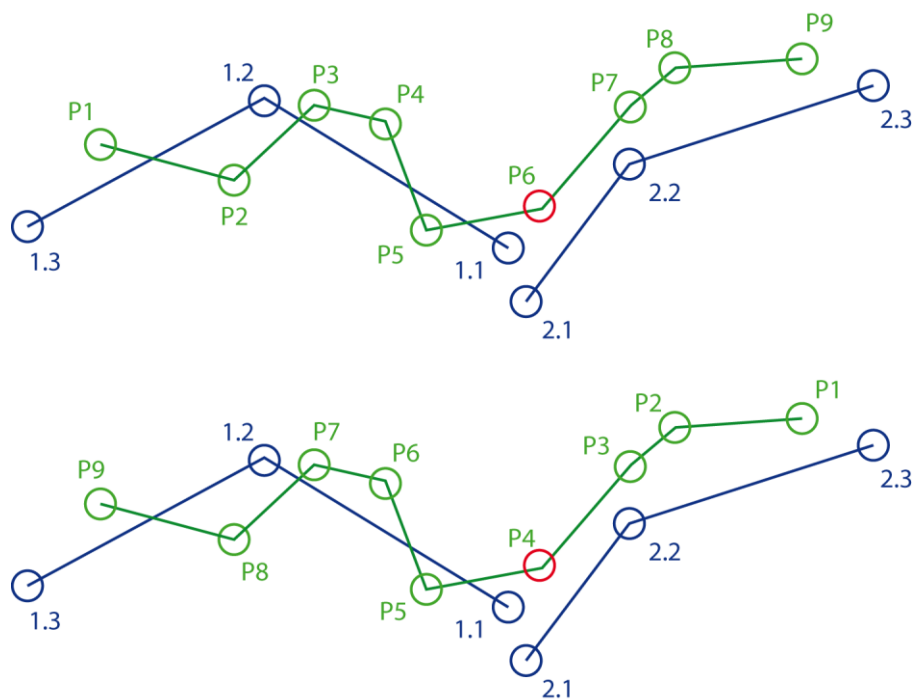


Abbildung 11: Orientierung, Spezialfall

Das Problem ist, dass beim zweiten Durchlauf der grüne Abschnitt dem zweiten blauen Abschnitt zugeordnet wird und damit die Orientierung wieder gegenläufig ist. Durch das erneute Umkehren würde allerdings nur wieder der Anfangszustand hergestellt und damit eine Endlosschleife generiert werden. Deshalb muss in diesem Fall der grüne Abschnitt an der rot markierten Stelle gesplittet werden. Die beiden neuen grünen Abschnitte werden anschliessend einzeln neu zugeordnet.

4.3.5 Punkte zuordnen

Nachdem der grüne Abschnitt korrekt orientiert dem blauen Abschnitt zugeordnet ist, kann mit der Zuordnung der Punkte begonnen werden. Dazu wird das Verfahren zur Untersuchung der Orientierung leicht abgeändert fortgesetzt. Der zuletzt ausgewählte blaue Punkt ist jeweils als X bezeichnet. Neu werden aber nur noch die Distanzen vom grünen Punkt zu den Punkten X und X+1 betrachtet. Der Punkt X-1 darf nicht mehr ausgewählt werden, weil ansonsten bei der

Einordnung der Punkte, deren Reihenfolge verändert würde. Dadurch wird dem fünften, ebenso wie dem vierten grünen Punkt, der blaue P4 zugeordnet.

Liegt der Punkt $X+1$ näher als X , wird zusätzlich die Distanz zum Punkt $X+2$ berechnet. Weist dieser wiederum die kürzere Distanz zum grünen Punkt auf als der Punkt $X+1$, wird der nächste Punkt getestet und so weiter. Dadurch wird im Beispiel in Abbildung 12 für den grünen Punkt 4 nicht nur die blauen P2 und P3, sondern auch P4 und P5 getestet. Korrekterweise wird am Ende der vierte blaue Punkt ausgewählt. Die Zuordnung aller grünen Punkte ist in der Tabelle festgehalten.

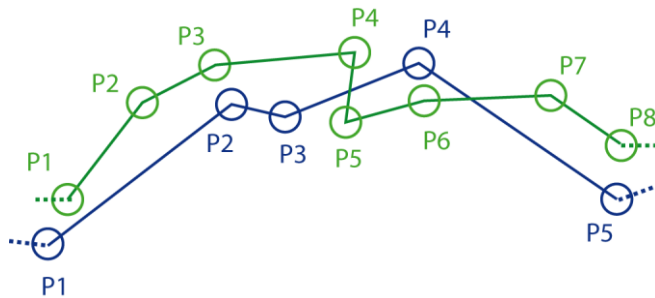


Abbildung 12: Punkte zuordnen

Grüner Punkt	Getestete blaue Punkte	Zugeordneter blauer Punkt	Umgekehrte Zuordnung
P1	P1,P2	P1	Ja
P2	P1,P2,P3	P2	Ja
P3	P2,P3	P2	Ja
P4	P2,P3,P4,P5	P4	Ja
P5	P4,P5	P4	Nein
P6	P4,P5	P4	Ja

Mit der Zuordnung eines blauen Punktes zum grünen wird auch jeweils die umgekehrte Zuweisung geprüft. Diese erfolgt allerdings nur, wenn der aktuelle grüne Punkt näher beim blauen liegt als alle zuvor zugewiesenen. Dadurch wird in Abbildung 12 zuerst der grüne P4 dem vierten blauen Punkt zugeordnet. Bei der Bearbeitung des sechsten grünen Punktes wird aber festgestellt, dass dieser die kürzeste Distanz zum blauen P4 aufweist.

Mit dieser Methode ist die Zuordnung für die blauen Punkte jedoch nicht komplett. Im Beispiel Punkte zuordnen wird dem blauen P3 kein grüner zugewiesen. Darum wird nach der Zuordnung aller grünen Punkte für alle blauen Punkte ohne Zuordnung der nächstliegende grüne gesucht. In Falle des blauen P3 wäre dies der fünfte grüne Punkt.

Die Punktzuordnung wird grundsätzlich so lange fortgesetzt, bis jedem grünen Punkt ein blauer zugeordnet wurde. Wenn jedoch das Ende des blauen Abschnitts erreicht ist, bevor dies der Fall ist, muss eine zusätzliche Massnahme getroffen werden.

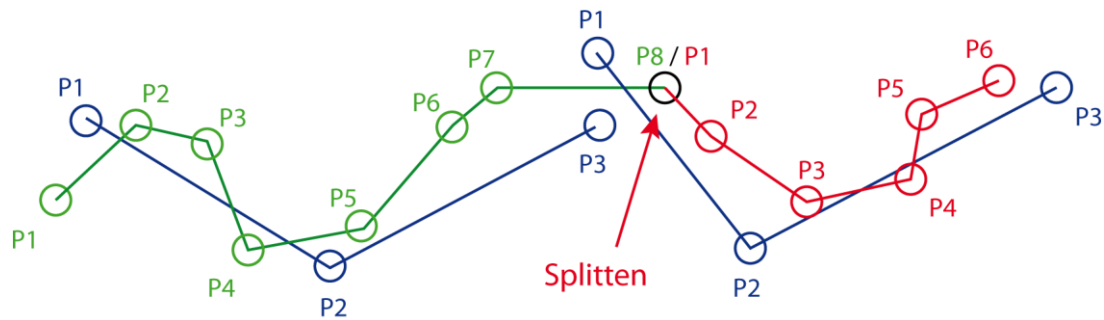


Abbildung 13: Punkte zuordnen, Spezialfall splitten

Wie in den Abbildungen 13 dargestellt, muss dafür der grüne Abschnitt gesplittet werden. Der Grund ist, dass offenbar nicht der ganze grüne Abschnitt zum gleichen blauen passt. Für den rot markierten neuen Abschnitt beginnt der Ablauf mit der Abschnittszuordnung von vorne. Für den gekürzten ursprünglichen Abschnitt ist die Punktzuordnung abgeschlossen. Der Punkt, an dessen Stelle gesplittet wird, ist in beiden Abschnitten enthalten.

An dieser Stelle sei noch kurz erwähnt, wozu die Zuordnung der Punkte gebraucht wird. Wie man in einigen Beispielen gesehen hat, werden teilweise mehrere grüne Abschnitte einem blauen zugewiesen. Dadurch werden die betreffenden grünen Abschnitte quasi zu einem Abschnitt zusammengesetzt. Die Reihenfolge in welcher die grünen Abschnitte zusammengefügt werden, ist durch die Zuordnung deren Anfangs- und Endpunkte bestimmt.

Die Zuordnung der restlichen grünen Punkte war eigentlich nur in einer früheren Version zwingend notwendig. Weil aber damals die Zuweisung der blauen Knoten darauf aufgebaut wurde, ist sie trotzdem beibehalten worden. Ausserdem hat man gesehen, dass die Zuordnung auch mit der Erkennung der zu splittenden Abschnitte verknüpft ist. Die Zuweisung der blauen Knoten, wird im Anschluss für die Positionierung der neuen Punkte benötigt.

4.3.6 Punkte positionieren: Detaillierter Datensatz

Nach der Zuordnung der Punkte wird für jeden blauen Punkt ein neuer im grünen Abschnitt generiert. In der Abbildung 14 wurde dem rot markierten Punkt im blauen Abschnitt der grüne Punkt X zugewiesen. Dieser ist der Ausgangspunkt für die Berechnung der Position des neuen Punktes.

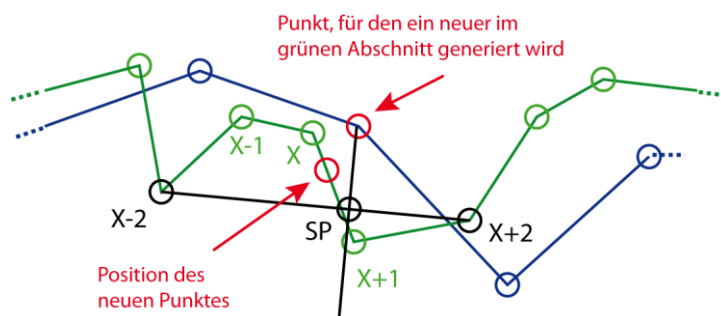


Abbildung 14: Punkte positionieren 1

Um einen repräsentativen Richtungsvektor für den Teil des grünen Abschnitts zu erhalten, werden die Punkte X-2 und X+2 miteinander verbunden. Anschliessend wird diese Linie vom bearbeiteten Punkt aus orthogonal geschnitten. Der Schnittpunkt SP definiert die Position des neuen Punktes auf der grünen Geometrie. Der neue Punkt teilt die grüne Linie von X-2 nach X+2 im gleichen Verhältnis wie der Schnittpunkt die schwarze Linie.

Liegt der Schnittpunkt nicht zwischen den Punkten X-2 und X+2 wird der neue Punkt mit den gleichen Koordinaten wie X-2 beziehungsweise X+2 eingefügt. Ausserdem verhindert das Programm das Vertauschen der Reihenfolge. Für das Beispiel in Abbildung 14 würde das konkret heissen, dass der nächste Punkt weiter rechts eingefügt werden muss als der markierte neue Punkt.

Nachdem für jeden blauen Punkt ein neuer generiert wurde, wird umgekehrt auch für den grünen ein neuer Punkt auf dem blauen Abschnitt eingefügt. Wie man in Abbildung 15 sieht, wird dazu die Lage der neuen Punkte des grünen Abschnitts ausgenutzt. Die Positionen der neuen Punkte auf dem blauen Abschnitt werden so gewählt, dass die Streckenverhältnisse zwischen den Punkten B1 und B2 auf beiden Linien gleich sind.

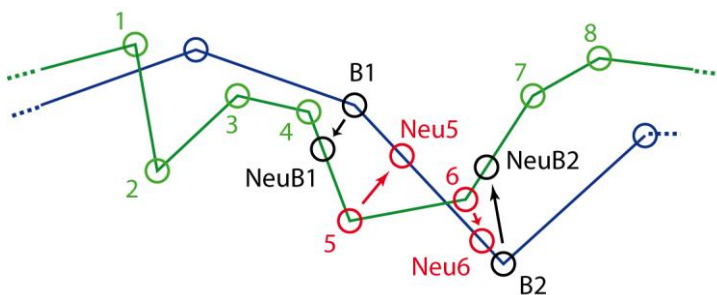


Abbildung 15: Punkte positionieren 2

Anschliessend ist die Vereinheitlichung des Aufbaus der beiden Geometrien abgeschlossen. Aufgrund der Generierung der neuen Punkte hat sich die Punktmenge in den beiden Datensätzen insgesamt verdoppelt. Entscheidend ist jedoch, dass beide Datensätze neu genau gleich viele Punkte enthalten. Jeder Punkt des einen Datensatzes hat seinen Gegenpunkt im anderen. Aus diesen beiden Datensätzen kann nun jeder Generalisierungszustand zwischen den beiden Ausgangszuständen einfach interpoliert werden. Deshalb muss die strukturelle Vereinheitlichung nur einmalig beim ersten Programmaufruf durchgeführt werden.

4.4 Geometrie interpolieren

Mit dieser Funktion werden die definitiven Stützpunkte der Flüsse berechnet. Dazu werden die Punktepaare der beiden Datensätze jeweils mit einer Geraden verbunden. Abhängig von der gewünschten Generalisierungsstufe werden alle diese Geraden im selben Verhältnis geteilt. Die dadurch generierten Punkte werden wie in Abbildung 16 zu Flussabschnitten verbunden und in der Karte dargestellt.

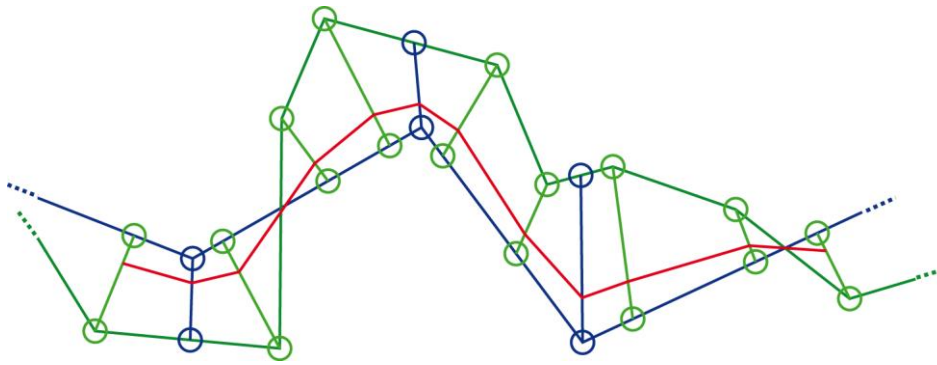


Abbildung 16: Interpolation der neuen Geometrie

5 Benutzeroberfläche

Die graphische Benutzeroberfläche des Programms sieht man in Abbildung 17. Neben dem Titel und der Karte enthält sie auch zwei Slider sowie eine Checkbox. Die Slider sind Teil von JQuery, einer JavaScript-Klassenbibliothek (jQuery, 2011). Diese kann kostenlos heruntergeladen werden. Um die Slider auch nutzen zu können, müssen sie in einem JavaScript-File initialisiert werden. Die Platzierung der Elemente erfolgt im HTML-Dokument. Zusätzlich zu HTML wird dafür auch CSS verwendet.

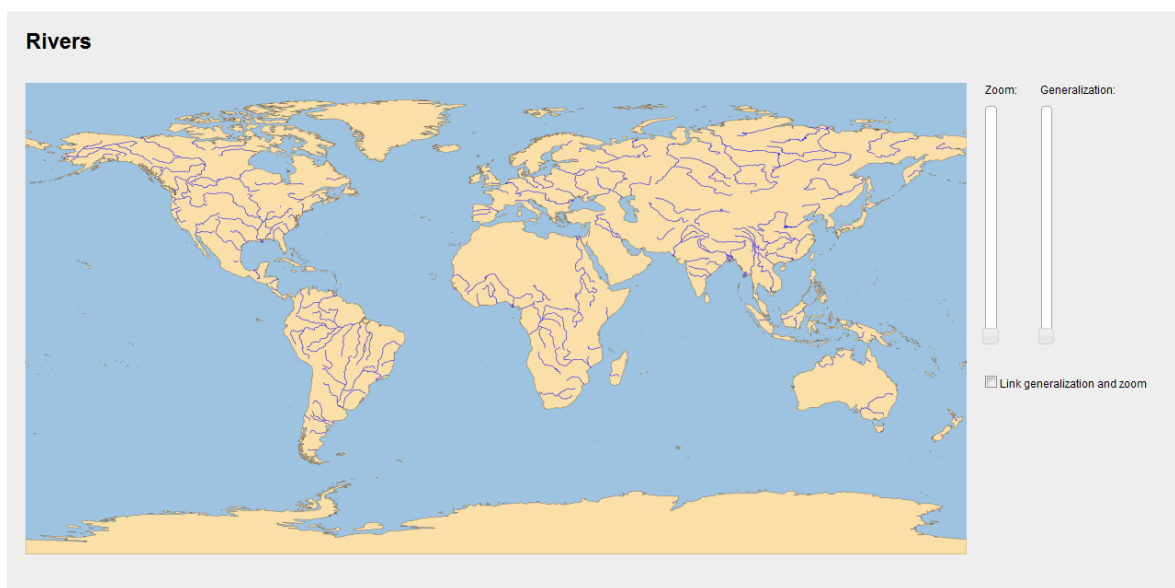


Abbildung 17: Screen-Shot, Programm

Beim ersten Slider kann der Zoom eingestellt werden. Im Gegensatz zu Google Maps, ist der Zoomwert frei wählbar. Das Zoomen ist zusätzlich auch durch scrollen mit der Maus im Kartenausschnitt möglich. Ausserdem kann mit der Maus der Kartenausschnitt verschoben werden. Wird ein neuer Zoomwert gewählt oder der Kartenausschnitt verschoben, wird dieser neu berechnet und angezeigt. Der zweite Slider dient zur Auswahl der gewünschten Generalisierungsstärke. Verschiebt man den Generalisierungsslider, wird der angezeigte Flussdatensatz neu interpoliert und dargestellt.

Mit der Checkbox können die beiden Slider miteinander verknüpft werden. Ist sie aktiviert, Passt sich die Generalisierung dem gewählten Zoomwert an und umgekehrt. Aufgrund des kleinen Masstabs passen die modellierten Generalisierungszustände jedoch nur zu den grössten 20 Prozent der Zoomwerte. Entsprechend passt sich die Generalisierung auch nur in diesem Zoombereich an.

6 Resultatbeurteilung

6.1 Einführung

In diesem Kapitel werden die Resultat des Programms anhand von den generierten Kartenausschnitten beurteilt. Gezeigt wird ein gelungenes Beispiel, aber auch Kartenausschnitte bei welchen das Programm versagt hat. In diesen Fällen wird der Fehlerursache eruiert. Zum Testen der Kompatibilität wurde das Programm zusätzlich mit einigen Landesgrenzen getestet.

6.2 Beispiel Ganges

Im ersten Beispiel (Abb. 18) ist das Gangesdelta in drei verschiedenen Generalisierungszuständen abgebildet. Dabei handelt es sich um einen der komplizierten Fälle. Trotzdem hat die Generierung der neuen Geometrie sehr gut funktioniert. Alle Abschnitte werden korrekt zugeordnet. Ebenso ist in den Abschnitten die Punktreihenfolge erhalten geblieben. Wenn man die drei Bilder miteinander vergleicht, kann man ausserdem feststellen, dass die Flusstrukturen mit abnehmender Generalisierungsstärke wie gewünscht feiner werden.

Beim Beispiel Ganges ist zudem sehr gut sichtbar, dass die Landfläche nicht mitgeneralisiert wird. Die Landfläche passt exakt zu den Flussmündungen des Datensatz 1:50 Millionen (Abb. 18, Bild 1). Der Datensatz 1:10 Millionen (Abb. 18, Bild 3) hat jedoch ein stark verschobene Uferlinie. Um diesen Umstand zu korrigieren, müsste die Uferlinie ebenfalls kontinuierlich generalisiert werden.



Ähnliche Resultate sind auch den meisten anderen Flüssen festzustellen. Die meisten Aufgetretenen Fehler konnten im Verlaufe der Programmentwicklung nach und nach korrigiert werden. Einige wenige sind jedoch bis zu diesem Zeitpunkt noch nicht gelöst worden. Diese Fehler werden in den folgenden Absätzen analysiert.

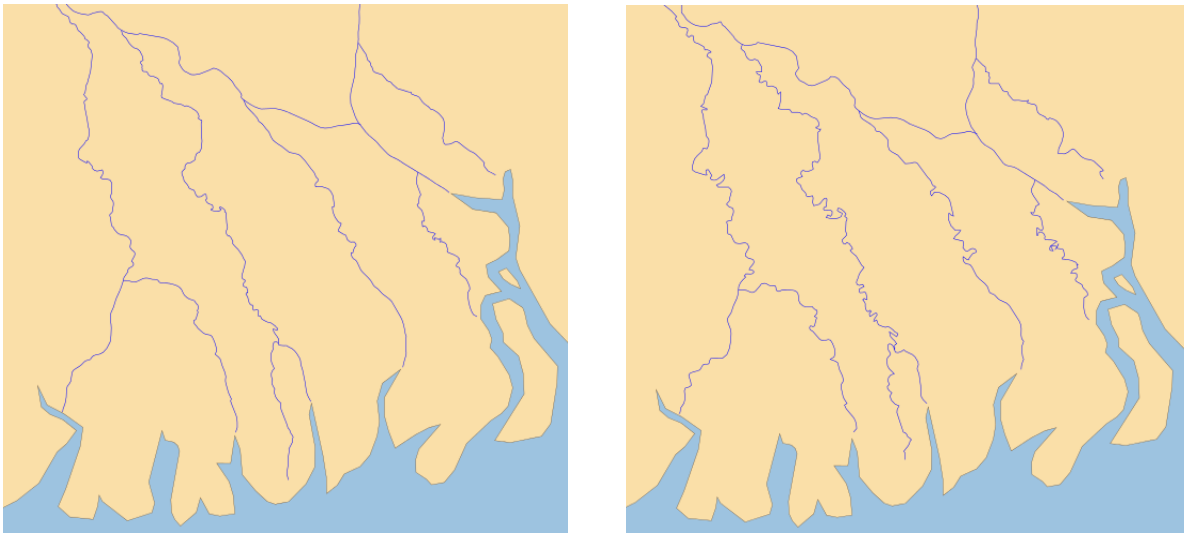


Abbildung 18: Resultatbeispiel Ganges, 3 Generalisierungsstufen

6.3 Fehleranalyse Flussdaten

In Abbildung 19 ist ein Problem, das bei der Generalisierung von Flussdaten auftreten kann. In der detaillierteren Flussdatensatz enthält eine geschlossene Schlaufe. Durch den Schnittpunkt in der Geometrie wird der Kreis als eigener Abschnitt gespeichert. Das Problem wird vermutlich bei der Untersuchung der Orientierung dieses Abschnitts auftreten. Angesichts der Punktreihenfolge im Kreis wird wohl der Spezialfall eintreten, dass der Abschnitt gesplittet wird. Dadurch ist am Ende die Schlaufe nicht mehr in der richtigen Reihenfolge in der neu aufgebauten Geometrie enthalten.



Abbildung 19: Fehler, Kreis

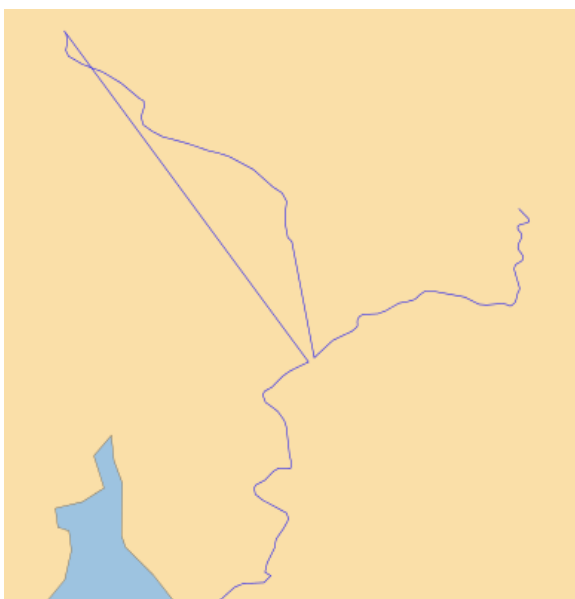


Abbildung 20: Fehler, Abschnitt ohne Zuordnung

Das Problem in Abbildung 20 ist, dass der Abschnitt oben links nur im detaillierteren Datensatz enthalten ist. Weil der erste Zuordnungsknoten am unteren Ende liegt, wird der Abschnitt irrtümlicherweise trotzdem einem Abschnitt zugeordnet. Um den Abschnitt zu eliminieren müsste eine neue Funktion programmiert werden. Die derzeitige Programmversion löscht Abschnitte aber nur in zwei Fällen: Zum einen, wenn der nächste Abschnitt mehr als ein Bogengrad vom Zuordnungsknoten entfernt ist. Ausserdem, wenn alle Punkte des Abschnitts dem selben Punkt zugeordnet werden und damit keine Orientierung bestimmt werden kann.

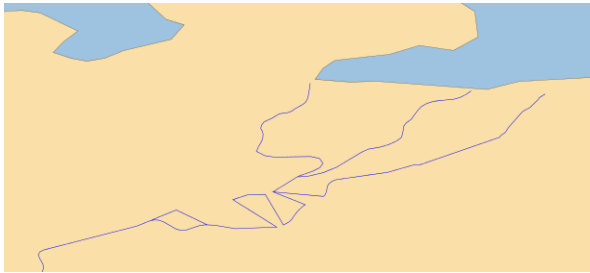


Abbildung 21: Fehler, Abschnittszuordnung

Abbildung 21 zeigt einen Fall, bei dem die Abschnittszuordnung versagt. Das Problem ist, dass dieses Flussdelta aus vielen kleinen Abschnitten besteht. Um dieses Problem zu lösen, müsste die Abschnittszuordnung für Abschnitte mit wenigen Stützpunkten verfeinert werden.

6.4 Kompatibilitätstest mit Landesgrenzen

Die Landesgrenzen wurden nicht für die Entwicklung des Programms verwendet. Das heisst, dass die Funktionen zur Generierung der neuen Geometrie nicht auf diese Datensätze abgestimmt sind. Zum Testen wurde das Programm auf die Landesgrenzen der Staaten Schweiz, Norwegen und Island angewendet. Bei allen drei Ländern wurde jeweils die Datensätze mit den Masstäben 1:10 Millionen und 1:110 Millionen verwendet. Dabei ergaben sich unterschiedliche Resultate.

Die einfachste Grenzstruktur dieser drei Länder weist die Schweiz auf. Bei beiden Datensätzen sind alle Stützpunkte im gleichen Abschnitt gespeichert. Wie man in Abbildung 22 erkennen kann, hat das Programm keine Schwierigkeiten bei der Modellierung der Generalisierung.



Abbildung 22: Analysebeispiel Schweiz, 2 Generalisierungsstufen

Anders sieht das im Beispiel Norwegen (Abb. 23) aus. Betrachtet man die neugenerierte undetaillierte Geometrie im linken Bild, stimmt auf den ersten Blick gar nichts mehr. Das rechte Bild der genauen Geometrie zeigt jedoch, dass bei der Umstrukturierung nur ein grob Fehler passiert. Dabei handelt es sich um die zahlreichen Inseln vor der Küste Norwegens. Davon sind nicht wenige im detaillierten Datensatz enthalten. Der ungenaue Datensatz besteht aber aus einem einzigen Abschnitt. Anstatt die Inseln zu löschen, werden einige trotzdem dem einzigen Abschnitt im ungenauen Datensatz zugeordnet. Damit ergibt sich das gleiche Problem wie bei den Flüssen in der Abbildung 20.

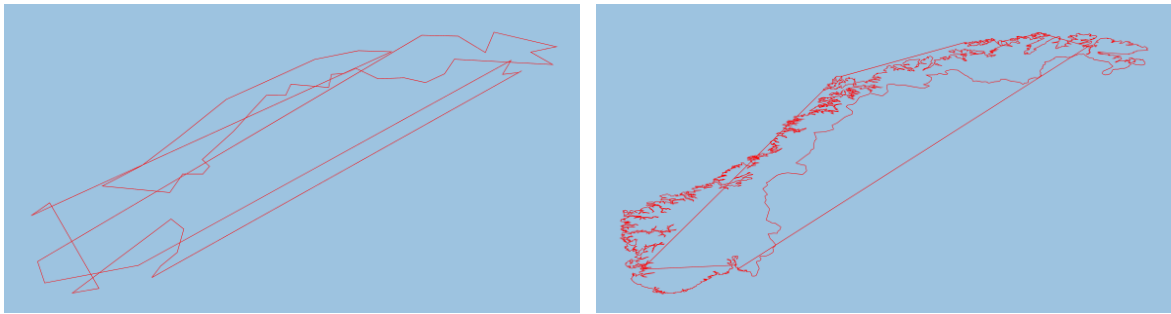


Abbildung 23: Analysebeispiel Norwegen, 2 Generalisierungsstufen

Dass die mangelhafte Umstrukturierung nicht durch verwinkelte Fjordlandschaft verursacht wird, beweist das Beispiel Island (Abb. 24). Auch die feinen Strukturen im Nordwesten des Landes werden korrekt verarbeitet.

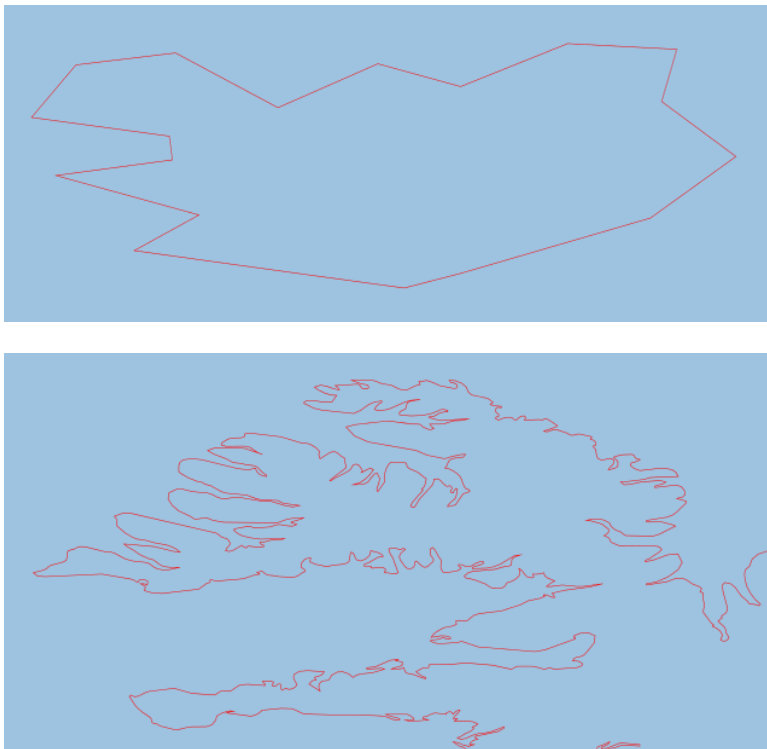


Abbildung 24: Analysebeispiel Island, 2 Generalisierungsstufen

7 Ausblick

7.1 Zielerreichung

Die Modellierung der kontinuierlichen Generalisierung zwischen zwei Flussdatensätzen wurde im Rahmen einer interaktiven Webkarte erfolgreich umgesetzt. Mit Hilfe einer Benutzeroberfläche wird dem Benutzer die Möglichkeit gegeben, den Zoom und die Generalisierungsstärke stufenlos zu wählen. Damit kann das erste Ziel als erreicht betrachtet werden. Einzig bei der Zuordnung von Abschnitten mit wenig Stützpunkten und bei der Erkennung von Abschnitten, welchen nur in einem Datensatz vorkommen, mussten kleinere Abstriche gemacht werden.

Die Programmierung der Funktionen zur Generierung der neuen Geometrie nahm allerdings deutlich mehr Zeit in Anspruch, als bei Projektbeginn erwartet. Deshalb wurden die weiteren Ziele, die Anpassung des Programms auf weitere Objektklassen, sowie die Berechnung eines Vektorfeldes zum Einbezug zusätzlicher Daten nicht umgesetzt. Zumindest wurden jedoch zum Abschluss dieser Arbeit die Eignung des Programms für eine weitere Objektklasse getestet. Dabei sind neben den bereits zuvor bekannten Fehlern keine neuen Probleme aufgetreten. Die Kompatibilität des Programms ist demzufolge zumindest in Bezug auf die Staatsgrenzen gegeben.

7.2 Weiterführende Ideen

Als erstes seien an dieser Stelle die geplanten weiteren Ziele erwähnt welche nicht umgesetzt wurden. Das Programm könnte mit weiteren Objektklassen getestet und gegebenenfalls darauf angepasst werden. Es könnte zusätzlich ausgebaut werden, sodass kontinuierliche Übergänge von mehreren Objektklassen in die gleiche Karte integriert werden können. Auch die Berechnung eines Vektorfeldes aus den Verschiebungen der Stützpunkt während der Generalisierung bleibt ein interessanter Ansatz. Damit könnten man neben zusätzlichen Vektordaten auch Rasterdaten wie ein Luftbild oder ein Relief in die Karte integrieren.

Ein weiterer Idee ist die Optimierung des Programms bezüglich der Ladezeiten. Wie bereits erwähnt, muss die Umstrukturierung der Geometrien nur einmalig beim ersten Programmaufruf ausgeführt werden. Weil diese Berechnungen nicht von Eingaben seitens des Benutzers abhängig sind, könnten sie genauso gut vorgängig ausgeführt werden. Anstelle der Shapefiles müssen neu einfach die im Aufbau vereinheitlichten Geometrien in der Datenbank abgelegt werden. Dadurch könnte die Ladezeit beim ersten Aufruf der Karte massiv verkürzt werden.

Eine weitere Möglichkeit zur Optimierung der Berechnungszeit, ist das speichern von einer sogenannten „Bounding Box“ für alle Objekte. Dadurch müssten beim rendern der Datensätze nur noch jene berücksichtigt werden, welche schlussendlich auch im Kartenausschnitt erscheinen. Weitere Ideen wurden teilweise im Hauptteil des Berichts erwähnt.

7.3 Fazit

Wie bereits erwähnt, konnten in dieser Arbeit mangels Zeit nicht alle festgelegten Ziele erreicht werden. Deshalb stellt sich die Frage, ob man in gewissen Bereichen hätte Zeit einsparen können. Dazu fällt grundsätzlich die wenig strukturierte Vorgehensweise auf. Statt zuerst einen Plan für das ganze Programm zu erstellen, wurde oftmals einfach drauf los programmiert. So wurde schliesslich die grösste Teil der Zeit zur Programmierung der Umstrukturierung verwendet.

Die Generalisierung selbst, wurde weit weniger ausführlich untersucht. Dies fällt auf, wenn die neu generierte Geometrie genauer betrachtet wird. Diese enthält gleich viele Punkte wie die beiden Ausgangssätze zusammen. Dies ergibt sowohl bezüglich der Generalisierung als auch der Berechnungszeiten keinen Sinn.

Positiv ist allerdings zu werten, dass mit der Weiterentwicklung des Programms trotzdem eine Webseite erzeugt wurde, bei der die kontinuierliche Generalisierung augenscheinlich funktioniert. Allerdings ist auch nach dem Abschluss dieser Arbeit, weiterhin einiges an Optimierungspotential vorhanden. Dies gilt sowohl für das Programm aber natürlich auch für das Projekt „Adaptive Zooming“ im Allgemeinen.

8 Literaturverzeichnis

Grundlagen

Geodaten

[Natural Earth]

Download: <http://www.naturalearthdata.com/downloads/>

Zugriff: 23.2.2011

Basisprogramm

[GitHub]

Download: <https://github.com/RandomEtc/shapefile-js>

Zugriff: 2.3.2011

Slider

[jQuery]

Download: <http://jqueryui.com/download>

Zugriff: 7.4.2011

Literatur

Internet

[Adaptive Zooming]

Cecconi, A., Galanda, M. (2011): Adaptive Zooming in Web Cartography

URL: http://www.svgopen.org/2002/papers/cecconi_galanda__adaptive_zooming/

Zugriff: 23.2.2011

[GitHub]

GitHub Inc (2011)

URL: <https://github.com/>

Zugriff: 18.5.2011

[Natural Earth]

Natural Earth (2011)

<http://www.naturalearthdata.com/>

Zugriff: 18.5.2011

[JavaScript: XMLHttpRequest]

Webmasterpro: Das XMLHttpRequest-Objekt

URL: <http://www.webmasterpro.de/coding/article/ajax-das-xmlhttprequest-objekt.html>

Zugriff: 19.5.2011

[JavaScript: onload]

PHP Performance (2008): JavaScript-Event onload und die bessere Alternative

URL: <http://phpperformance.de/javascript-event-onload-und-die-bessere-alternative/>

Zugriff: 19.5.2011

[JavaScript: Array]

SELFHTML (2007): Array

URL: <http://de.selfhtml.org/javascript/objekte/array.htm>

Zugriff: 9.3.2011

[Shapefile]

ESRI (1998): ESRI Shapefile Technical Description

URL: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

Zugriff: 18.5.2011

Bücher, Zeitschriften

[JavaScript]

Flanagan, D. (2002): JavaScript, The Definitive Guide. Sebastopol, CA: O'Reilly, 916 S.

[Canvas]

Flanagan, D. (2011): Canvas, Pocket Reference. Sebastopol, CA: O'Reilly, 99 S.

[HTML5]

Braun, H. (2011): Report - HTML5. c't-Magazin, 11/2011, 140-145.