Institut für Formale Methoden der Informatik

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3624

Offline Speicherung und Darstellung von Geodaten auf Mobilgeräten

Filip Krumpe

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Stefan Funke
Betreuer/in:	Prof. Dr. Stefan Funke
Beginn am:	10.02.2014
Beendet am:	12.08.2014
CR-Nummer:	E1, H5.2, I.3.5

Kurzfassung

Die Visualisierung von großen Mengen geographischer Basisdaten, wie Stadtnamen und Verwaltungsgebieten, ist mit verschiedenen Problemen verbunden. Werden zu viele Daten und Details auf zu kleinem Raum dargestellt, kann die Übersichtlichkeit der Darstellung leiden oder Überdeckung die Wahrnehmbarkeit einzelner Objekte stören. Ist eine Interaktion mit der Visualisierung möglich (beispielsweise durch verschieben, rotieren und zoomen des betrachteten Ausschnitts), kann eine große Menge an darzustellenden Daten Performanceprobleme verursachen und eine verzögerungsfreie Interaktion verhindern.

Die vorliegende Diplomarbeit beschäftigt sich mit der Visualisierung von Ortslagen und Grenzsegmenten von Verwaltungsgebieten der Bundesrepublik Deutschland. In der Arbeit werden zwei Methoden entwickelt, um die entsprechenden Daten zu strukturieren und zu speichern. Die entwickelten Datenstrukturen ermöglichen eine einfache Ableitung von Teilmengen der gespeicherten Daten, die in einer Visualisierung konfliktfrei und gut wahrnehmbar dargestellt werden können. Für die Speicherung der Daten wird ein binäres Dateiformat entwickelt, das eine kompakte Speicherung der Datenstrukturen ermöglicht.

Die Grenzsegmente der Verwaltungsgebiete können abhängig von ihrer Zugehörigkeit zu verschiedenen Verwaltungsebenen abgefragt werden. Um den Aufwand für das Zeichnen der Segmente zu verringern, können sie in verschiedenen Generalisierungsstufen abgefragt werden.

Im Rahmen der Diplomarbeit wurden, basierend auf den entwickelten Datenstrukturen, zwei Visualisierungen der Daten entwickelt. Sie ermöglichen ein interaktives Erkunden der Datensätze mittels Verschiebung, Rotation und Zoom des dargestellten Datenausschnitts in Echtzeit auf einem PC und einem Android Gerät.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

1. Einleitung

Die vorliegende Diplomarbeit beschäftigt sich mit der automatisierten Darstellung von geographischen Basisdaten, das heißt Ortsnamen und Grenzen von Verwaltungsgebieten, im Gebiet der Bundesrepublik Deutschland. Das Ziel der Arbeit war die Entwicklung einer Datenstruktur zur effizienten Speicherung und Abfrage der Daten. Aufbauend auf der Datenstruktur wurden zwei Anwendungen entwickelt, um die Datensätze zu visualisieren und diese Visualisierungen interaktiv mittels Rotation, Translation und Zoom erkundbar zu machen. Um die Anwendung der Interaktionen des Benutzers mit der Visualisierung des Datensatzes verzögerungsfrei umsetzen zu können, lag der Fokus bei der Entwicklung der Datenstruktur insbesondere auf der Geschwindigkeit der Abfrage. Ein weiteres Ziel war, den Platzverbrauch der Datenstruktur gering wie möglich zu halten, sodass der Datensatz offline in einer mobilen App gespeichert und visualisiert werden kann.

Als Datenbasis für die vorliegende Arbeit wurden zwei Datensätze (GN250 und VG250) des deutschen Bundesamtes für Kartographie und Geodäsie verwendet. Der Datensatz GN250 enthält unter anderem 45.037 Ortsnamen, deren Lage und einige weitere Informationen (für eine ausführliche Beschreibung des Datensatzes siehe Abschnitt 2.1). Der Datensatz VG250 enthält Informationen zu 16.707 Verwaltungsgebieten der Bundesrepublik Deutschland, er umfasst sowohl Informationen zum Staatsgebiet, zu den Bundesländern und den verschiedenen Regierungsbezirken als auch zu Kreisen, Verwaltungsgemeinschaften und Gemeinden in Deutschland. Die Informationen des VG250-Datensatzes umfassen die Grenzverläufe und Flächen der Verwaltungsgebiete sowie verschiedene Attribute der Gebiete, wie Einwohnerzahl und die Zugehörigkeit zu Verwaltungsgebieten in der nächsthöheren Verwaltungsebene (für eine ausführliche Beschreibung des Datensatzes siehe Abschnitt 2.2).

Das Ergebnis des praktischen Teils der Arbeit gliedert sich in verschiedene Komponenten. Sämtliche Komponenten wurden in Java umgesetzt und sind unter der Apache 2.0 Lizenz zugänglich. Die *erste Komponente* ist eine Konsolenanwendung, durch welche die zugrunde liegenden Datensätze in ein dafür entwickeltes, binäres Datenformat überführt werden können. In dieser Komponente werden zudem verschiedene Vorverarbeitungsschritte zur Strukturierung der Daten durchgeführt, die für die Visualisierung der Daten notwendig sind (siehe dazu auch Abschnitt 4.2 und 5.1). Eine *zweite Komponente* dient dem Zugriff auf den erstellten Datensatz zur Laufzeit. Die Komponente bildet eine Abstraktionsschicht über den grundlegenden Datenstrukturen und stellt einfache Abfragefunktionen für diese bereit. Diese Komponente ist mittels einer Suchstruktur über den Daten sowie einen direkten Zugriff auf die Datensatzdateien implementiert (siehe dazu Kapitel 6 und 7). Die *dritte und vierte Komponente* zeigen die Möglichkeiten der Datenstruktur, indem die bereitgestellten Daten jeweils in einer Anwendung visualisiert werden und dem Nutzer Interaktionsmöglichkeiten zum Erkunden der Datensätze zur Verfügung gestellt werden. Die entworfenen Anwendungen visualisieren die Daten mithilfe von OpenGL bzw. OpenGL ES als PC-Anwendung bzw. als Android App.

1.1. Verbundene Arbeiten

Die vorliegende Diplomarbeit baut auf meiner Studienarbeit von 2014 auf (vgl. [Kru14]). In dieser Studienarbeit lag der Fokus insbesondere auf der Verarbeitung, Speicherung und Darstellung der Grenzsegmente von Verwaltungsgebieten. Eine darin entwickelte Visualisierung basierte auf JavaFX. Die vorliegende Diplomarbeit geht über diese Arbeit hinaus, indem sie die Problematik der Verarbeitung und Darstellung der Ortsnamen vertieft. Dafür wurde eine Datenstruktur für die Verwaltung der Ortsnamen entwickelt, die die Ableitung einer überdeckungsfrei darstellbaren Teilmenge der Ortsnamen ermöglicht. Der in der Studienarbeit genutzte Ansatz der Verwaltung der Daten in einer SQLite-Datenbank wurde komplett durch einen neuen Ansatz ersetzt, in dem die Daten binär kodiert in Dateien abgelegt werden. Durch diese Maßnahme ist es möglich den benötigten Platz für die Speicherung der Daten zu verringern und zugleich die Geschwindigkeit für den Zugriff auf die Daten zu erhöhen. Zudem wurde die in der Studienarbeit entwickelte Visualisierung komplett verworfen und durch eine hardwarebeschleunigte, OpengGL-basierte Visualisierung ersetzt. Zusätzlich zu der Visualisierung als Java PC-Anwendung, wurde in der vorliegenden Diplomarbeit eine Visualisierung für die Android-Plattform entwickelt.

Durch den Fokus auf der Darstellung der Ortsnamen, ist die vorliegende Arbeit im erweiterten Kreis der Forschungsarbeiten zum Problem des *Map Labeling* zu sehen, das bereits 1962 in einer Arbeit von Eduard Imhof beschrieben wurde (vgl. [Imh62]). Die dort beschriebenen Map-Labeling Probleme (oder auch als Label Positioning Probleme bezeichnet) lassen sich in drei Klassen unterteilen (vgl. [CMS95]). Das *Labeling of Area Features* beschreibt das Problem der Platzierung von Labels, die sich auf ganze Bereiche einer Karte, wie Seen, Meere und Gebiete, beziehen. *Labeling of Line Features* bezeichnet das Problem, wie eine optimale Position für Labels, die entlang einer Linie beliebig platziert werden können, gefunden werden kann. Beispiele für ein solches Problem sind das Beschriften von Straßen oder Flüssen in einer Karte. Die dritte Problemklasse umfasst das *Labeling of Point Features*, dabei geht es um das Finden einer optimalen Position für Labels, die einem Punkt zugeordnet sind, beispielsweise einem Ort auf einer Karte. Dieser Klasse kann das, in dieser Arbeit behandelte, Problem der Platzierung der Ortsnamen zugeordnet werden, auch wenn sich die hier behandelte Fragestellung von dem klassischen Problem unterscheidet (die Unterschiede werden in Abschnitt 4.1 herausgearbeitet).

Die vorliegende Arbeit ist mit weiteren Arbeiten aus der Kartographie verbunden, welche das Problem der Generalisierung von Liniensegmenten, die durch eine Folge von Punkten definiert sind (Polylinien), betreffen. Das Problem tritt auf, wenn Polylinien mit einer sehr feinen Auflösung sehr grob aufgelöst dargestellt werden sollen. Dabei müssen viele Punkte der Polylinie verarbeitet werden, die den Verlauf der Linie in der Darstellung nicht effektiv beeinflussen, da sie in der Darstellung auf denselben Punkt abgebildet werden. Diese überflüssigen Schritte können den notwendigen Aufwand zur Visualisierung deutlich erhöhen. David H. Douglas und Thomas K. Peucker beschrieben 1973 ein Verfahren, das eine Reduktion der Details eines Liniensegments ermöglicht, ohne dass zugleich der grobe Verlauf der Polylinie geändert wird (vgl. [DP73] und [Ebi02]). Dieses Verfahren wird in der vorliegenden Arbeit angewendet, um die Grenzsegmente der Verwaltungsgebiete zu vereinfachen und so den Aufwand für die Darstellung der Segmente zu verringern. Das Verfahren wird in Abschnitt 5.2 ausführlich beschrieben.

1.2. Gliederung

Die vorliegende Arbeit gliedert sich in sechs große Kapitel, welche einzelne Aspekte der Entwicklungen im Rahmen der Diplomarbeit erläutern.

In Kapitel 2: “Datengrundlage” werden die zugrunde liegenden Datensätze der geographischen Basisdaten, sowie die darin enthaltenen Informationen der Datenelemente und der Aufbau der einzelnen Dateien der Datensätze beschrieben. Zudem wird das UTM-Koordinatensystem erläutert, welches in der vorliegenden Arbeit zur Beschreibung der Lage von geographischen Objekten verwendet wird.

Kapitel 3: “Anforderungen an die Datenstruktur und die Visualisierung” beschreibt die grundlegenden Anforderungen, die bei der Visualisierung der Daten erfüllt werden sollen. Die dort beschriebenen Anforderungen bilden die Grundlage für die Entwicklung der Datenstrukturen zur Repräsentation der Daten.

Die Datenstruktur für die Verwaltung der Ortsnamen wird in Kapitel 4: “Vorverarbeitung und Strukturierung der Ortsnamen” entwickelt. Die Datenstruktur basiert auf einer Klassifizierung der Ortsnamen. Darauf aufbauend wird eine Datenstruktur der Datenpunkte entwickelt, welche die Ableitung einer konfliktfrei darstellbaren Teilmenge der Datenpunkte ermöglicht. Das entwickelte Dateiformat, das für die Speicherung der Daten verwendet wird, wird beschrieben.

Im darauf folgenden Kapitel 5: “Vorverarbeitung und Strukturierung der Grenzsegmente” werden Maßnahmen beschrieben, die zur Strukturierung des Datensatzes der Verwaltungsgebietsgrenzen und zur Generalisierung der Grenzverläufe der Gebiete vorgenommen werden. Das, für die Speicherung Daten verwendete, selbst entwickelte binäre Datenformat wird ebenfalls beschrieben.

Um die Datenstrukturen, die in Kapitel 4 und 5 beschrieben wurden, zur Laufzeit möglichst effizient abfragen zu können, wurde in der vorliegenden Arbeit eine Suchstruktur für die enthaltenen Daten entwickelt. Diese Suchstruktur und ihre Funktion wird in Kapitel 6: “Die Suchstruktur und Abfrage der geographischen Basisdaten” erläutert.

Im Rahmen der vorliegenden Diplomarbeit wurden prototypisch zwei Anwendungen entwickelt, welche auf den entwickelten Datenstrukturen und Zugriffsmethoden basieren. Sie stellen eine interaktiv erkundbare Visualisierung der geographischen Basisdaten zur Verfügung und zeigen die Leistungsfähigkeit der entwickelten Datenstrukturen. Einige Implementierungsdetails, sowie Geschwindigkeitsmessungen der Anwendungen werden in Kapitel 7: “Implementierung und Messergebnisse” dokumentiert.

Eine Zusammenfassung und ein Ausblick schließen die Arbeit in Kapitel 8: “Zusammenfassung und Ausblick” ab.

2. Datengrundlage

Als Datengrundlage der vorliegenden Arbeit wurden die, bereits in der Studienarbeit verwendeten, Datensätze GN250 (vgl. Dokumentation in [GN-13]) und VG250 (vgl. Dokumentation in [VG-13]) genutzt. Beide Datensätze sind im Rahmen der OpenData Initiative des GeoDatenZentrums des Bundesamtes für Kartographie und Geodäsie (http://www.bkg.bund.de) als Open Source frei zugänglich. Diese Datensätze wurden insbesondere deshalb ausgewählt, weil sie im Zusammenhang mit der INSPIRE-Initiative der EU stehen. Das Ziel der Initiative ist die Bereitstellung von europaweiten geographischen Daten (vgl. http://inspire.ec.europa.eu/index.cfm). Es besteht die Hoffnung, dass somit ähnliche Datensätze für andere Länder der EU veröffentlicht werden und sich diese Datensätze ohne großen Aufwand in die entwickelte Lösung einpflegen lassen.

Der Datensatz GN250 enthält die Namen von geographischen Objekten, wie beispielsweise Gemeinden, Landschaften, Gebirgen, Bergen, Inseln und Flüssen. Um mögliche Konflikte zwischen der Darstellung der verschiedenen Objekte zu vermeiden, beschränkt sich die vorliegende Arbeit ausschließlich auf die Verarbeitung und Darstellung der Ortsnamen. Die Daten des GN250 Datensatzes sind in verschiedenen Koordinatensystemen verfügbar. In der vorliegenden Arbeit werden die UTM-Koordinaten verwendet. Für eine Erläuterung des Koordinatensystems siehe Abschnitt 2.3. Eine ausführliche Beschreibung des GN250 Datensatzes ist im Abschnitt 2.1 zu finden.

Der Datensatz VG250 enthält Informationen zu den Verwaltungsgebieten der Bundesrepublik Deutschland, beispielsweise den Verlauf der Verwaltungsgebietsgrenzen sowie verschiedene Attribute der einzelnen Verwaltungsgebiete, wie Einwohnerzahl und der Zugehörigkeit zu Verwaltungsgebieten der nächsthöheren Verwaltungsebene. Für die vorliegende Arbeit werden aus dem Datensatz die Verläufe der Grenzen zwischen den einzelnen Verwaltungsgebieten genutzt. Für die Referenzierung der Objekte werden ebenfalls die UTM-Koordinaten genutzt. Eine detaillierte Beschreibung des Datensatzes und der enthaltenen Attribute ist in Abschnitt 2.2 zu finden.

Der Vorteil der Nutzung des UTM-Koordinatensystems besteht darin, dass die Koordinaten in einem kartesischen Koordinatensystem angegeben sind, wobei die Koordinaten der Objekte in Meter angegeben sind. Das ermöglicht im Falle der Ortsnamen eine einfache Berechnung der Abstände zwischen verschiedenen Objekten. Im Fall der Verwaltungsgebietsgrenzen ermöglicht die Verwendung des Koordinatensystems eine einfache Berechnung der Abstände zwischen Punkten und Geraden, eine zentrale Operation bei der Approximation der Grenzsegmente.

2.1. Geographische Namen: GN250

Der GN250 Datensatz stellt die Namen von verschiedenen geographischen Objekten zur Verfügung. Er umfasst unter anderem die Namen von Städten, Gemeinden, Landschaften, Bergen, Flüssen und

Seen (vgl. [GN-13]). Der Detailgrad des Datensatzes orientiert sich an einem Kartenmaßstab von 1:250.000m und enthält insgesamt ca. 120.000 Einträge (vgl. [GN-13]). Zusätzlich zu den Namen und Koordinaten der geographischen Objekte enthält der Datensatz teilweise weitere Informationen zu den Objekten, wie beispielsweise ihre Höhe über dem Meeresspiegel, die Sprache, in der der entsprechende Name angegeben ist, sowie verschiedene Kennzahlen, die die Zuordnung der Objekte zu Verwaltungsgebieten ermöglicht. Weitere Informationen sind das umschließende Rechteck sowie bei Ortsnamen die Einwohnerzahl der jeweiligen Orte (siehe auch Abschnitt 2.1.1).

Die verzeichneten Objekte sind verschiedenen Typen zugeordnet, die mit einem speziellen Objektcode des Amtlich Topographisch-Kartographischen Informationssystems (ATKIS) bezeichnet werden (vgl. [Har01]). So werden beispielsweise Ortslagen, Gewässer, Straßenverkehrsanlagen usw. unterschieden. Eine Auflistung aller im Datensatz vorhandenen Objekttypen und deren Untertypen ist im Anhang unter A.1 zu finden. In der vorliegenden Arbeit liegt der Fokus auf den ATKIS-Objekten des Typs AX_Ortslage. Objekte dieses Typs beschreiben die Lage von Orten (Großstädten, Städten, Dörfern usw.).

Im folgenden Abschnitt 2.1.1 werden ausgewählte Attribute des GN250 Datensatzes beschrieben und die für die Visualisierung genutzten Attribute des Datensatzes erläutert. Im darauf folgenden Abschnitt 2.1.2 wird der Aufbau des originalen GN250-Datensatzes beschrieben.

2.1.1. Die Attributen der geographischen Objekte

Objekte des GN250 Datensatzes besitzen unter anderem die in Tabelle 2.1 dargestellten Attribute (vgl. [GN-13]). Es ist zu beachten, dass verschiedene Attribute nicht für alle Datenpunkte definiert sind. Das Attribut EWZ (Einwohnerzahl) ist beispielsweise nur für Verwaltungsgebiete angegeben, EWZ_GER nur für Objekte des ATKIS-Typs *AX_Ortslage*.

Die für die vorliegende Arbeit interessanten Attribute werden im Folgenden kurz erläutert (vgl. dazu auch [GN-13]):

OBA beschreibt den Typ des Datenobjekts. Die vorliegende Arbeit beschränkt sich auf die Daten des ATKIS-Typs AX_Ortslage.

OBA_WERT enthält eine Kodierung des Attributs *OBA*.

EWZ enthält die Einwohnerzahl eines Verwaltungsgebiets.

EWZ_GER enthält die errechnete Einwohnerzahl einer Ortslage. Die errechnete Einwohnerzahl eines Ortes leitet sich aus der Einwohnerzahl der Gemeinde ab. Diese wird prozentual (relativ zur Größe der Bounding Box) auf die, der Gemeinde zugehörigen, Orte verteilt (vgl. [GN-13]).

VIRTUELL ist ein boolescher Wert und legt fest, ob eine entsprechend gekennzeichnete Ortslage real oder virtuell ist. Reine virtuelle Ortslage besteht tatsächlich aus verschiedenen realen Teilgemeinden. Ein Beispiel aus dem vorliegenden Datensatz ist Berlin, das als virtuelle Ortslage aus den Teilgemeinden Charlottenburg-Wilmersdorf, Friedrichshain-Kreuzberg, Lichtenberg, Marzahn-Hellersdorf, Mitte, Neukölln, Pankow, Reinickendorf, Spandau, Steglitz-Zehlendorf und Teptow-Köpenick besteht. Zu beachten ist, dass bei den als *virtuell* gekennzeichneten Ortslagen kein *EWZ_GER*-Attribut angegeben ist. Über den *AGS*-Schlüssel der virtuellen Ortslage

Attribut	Beschreibung
OBA	AKTIS-Typ des Objekts (siehe Liste der Objektarten im Anhang A.1)
OBA_WERT	ATKIS-Code des Objekts
NAME	Name des Objekts
ZUSATZ	Zusatz zur genaueren Beschreibung des Objekttyps
AGS	Einzigartiger Gemeindeschlüssel einer Gemeinde
HOEHE	Höhe des Objekts
HOEHE_GER	Gerechnete Höhe des Objekts (für Ortslage)
EWZ	Einwohnerzahl von Gemeinden
EWZ_GER	Gerechnete Einwohnerzahl (für Ortslagen)
VIRTUELL	Objekt (Gemeinde oder Ortslage) ist real oder virtuell
GEMEINDE	Name der zugehörigen Gemeinde
KREIS	Name des zugehörigen Kreises
REGBEZIRK	Name des zugehörigen Regierungsbezirks
BUNDESLAND	Name des zugehörigen Bundeslands
STAAT	Name des zugehörigen Staates
UTMRE	Rechtswert der UTM-Koordinaten
UTMHO	Hochwert der UTM-Koordinaten
BOX_UTM	Kleinstes umschließendes Rechteck des Objekts in UTM- Koordinaten

Tabelle 2.1.: Auszug aus der Liste der Attribute des GN250 Datensatzes (vgl. [GN-13]).

lassen sich jedoch zugehörige Ortschaften identifizieren (sie besitzen den selben *AGS*-Schlüssel) und daraus die Einwohnerzahl einer virtuellen Ortslage rekonstruieren.

BOX_UTM, **UTMRE**, **UTMHO** sowie **HOEHE** bzw. **HOEHE_GER** bezeichnen die Größe eines umschließenden Rechtecks und die Position des Objekts.

NAME bezeichnet den Namen des Objekts.

Die Attribute UTMRE und UTMHO sowie NAME sind die zentralen Attribute, die für die Visualisierung benötigt werden. Die Attribute OBA, EWZ_GER sowie VIRTUELL sind wichtige Attribute für die Strukturierung der Daten des GN250 Datensatzes.

Alle Attribute mit dem Präfix *UTM* beziehen sich auf das UTM-Koordinatensystem, das im Abschnitt 2.3 detaillierter erläutert wird. Der Datensatz kann auch mit Referenzen in Geographischen Koordinaten (Länge und Breite) und als Gauß-Krüger-Abbildung bezogen werden (vgl. [GN-13]).

Die verbliebenen Attribute aus Tabelle 2.1 ermöglichen abzuleiten, welchen Verwaltungseinheiten ein Objekt zugeordnet ist. Da derartige Informationen in der Visualisierung nicht darstellbar sind, werden die entsprechenden Attribute in der vorliegenden Arbeit nicht verwendet.

2.1.2. Format des Datensatzfiles

Der GN250 Datensatz ist auf der Homepage des Bundesamtes für Kartographie und Geodäsie (http://www.bkg.bund.de) unter der Rubrik OpenData verfügbar. Der gesamte Datensatz kann dort als Comma-Separated-Values (CSV) formatiertes Textfile heruntergeladen werden (gezippt ca. 32MB). Die erste Zeile der Datei enthält die Spaltennamen (d.h. die Attributsnamen), in den darauf folgenden Spalten ist jeweils der Attributssatz des Objekts enthalten.

Neben der oben genannten CSV-formatierten Datei ist der Datensatz zudem im SHAPE-Format (siehe Abschnitt 2.2.3 sowie vgl. [S-D98]) verfügbar. Er kann, neben den hier genutzten UTM-Koordinaten, auch referenziert in Geographischen Koordinaten (Länge und Breite) und als Gauß-Krüger-Abbildung bezogen werden (vgl. [GN-13]).

2.2. Verwaltungsgebiete: VG250

Der VG250-Datensatz beinhaltet sämtliche Verwaltungsgebiete der Bundesrepublik Deutschland von der Staats- bis zur Gemeindeebene (vgl. Dokumentation in [VG-13]). Der Datensatz enthält Definitionen sämtlicher Verwaltungsgebietsgrenzen, der Verwaltungsgebietsflächen sowie verschiedene Attribute der einzelnen Verwaltungsgebiete, wie Name und Zugehörigkeit zu Verwaltungseinheiten auf höherer Verwaltungsebene. Zusätzlich ist im Datensatz VG250-EW die Einwohnerzahl der Verwaltungsgebiete enthalten. Die Genauigkeit der Daten, insbesondere der Flächen- und Grenzverlaufsdefinition, orientiert sich an einem Kartenmaßstab von 1:250.000m. Die verfügbaren Verwaltungsgebiete sind sechs Verwaltungsebenen zugewiesen: Staats-, Bundesland-, Regierungsbezirks-, Kreis-, Verwaltungsgemeinschafts- und Gemeindeebene.

Die Daten werden in zwei verschiedenen Ausprägungen zur Verfügung gestellt (vgl. [VG-13]):

VG250_Kompakt: Hier sind die Verwaltungsgebiet der niedrigsten Ebene gespeichert. Über die Attribute der Objekte ist festgelegt, wie sich die Objekte einer höheren Verwaltungsebene aus den Objekten der niedrigeren Ebene zusammensetzen.

VG250_Ebenen: Hier sind für jede Verwaltungsebene eigene Datensätze vorhanden, in denen jeweils alle benötigten Datenobjekte enthalten sind. Dadurch werden verschiedene Objekte auf unterschiedlichen Ebenen mehrfach gespeichert, was zu einem erhöhten Speicherverbrauch führt.

Der Datensatz beinhaltet folgende Informationen (vgl. [VG-13]):

1. Informationen zu den Verwaltungsflächen (VG250_F.xxx - Codiert als SHAPE- Objekte)
2. Informationen zu den Grenzlinien (VG250_L.xxx - Codiert als SHAPE- Objekte)
3. Informationen zu den Attributen der Grenzsegmente (VG250_NAM.dbf)
4. Informationen zu der hierarchischen Struktur der Verwaltungsgebiet (VG250_ISN.dbf)

Die vorliegende Arbeit nutzt aus dem Datensatz die Informationen zu den Grenzverläufen zur Visualisierung der Verwaltungsgebiete. Die Flächen werden in der vorliegenden Arbeit nicht weiter betrachtet. Die Grenzsegmente sind in den Datensätzen durch eine Folge von Punkten definiert, die den Verlauf des entsprechenden Grenzsegments vorgeben. Es ist zu beachten, dass die Kreuzungspunkte von Grenzverläufen stets Endpunkten der Grenzsegmente entsprechen. Es ist demnach ausgeschlossen, dass ein Grenzsegment in der Mitte eines anderen Grenzsegments beginnt. Diese Beobachtung ist insbesondere für die unten beschriebene Generalisierung der Grenzsegmente (siehe Abschnitt 5.2) wichtig.

Im Folgenden werden die Attribute der Verwaltungsgebiete (in Abschnitt 2.2.1) und der Verwaltungsgebietsgrenzen (in Abschnitt 2.2.2) beschrieben. Daran anschließend wird in Abschnitt 2.2.3 der Aufbau der Dateien des VG250 Datensatzes erläutert, wobei insbesondere auf den Teil des Datensatzes eingegangen wird, der die Grenzsegmente enthält, da diese Daten für die Visualisierung genutzt werden.

2.2.1. Attribute der Verwaltungsgebiete

Eine Auswahl der Attribute des VG250 Datensatzes ist in Tabelle 2.2 dargestellt. Die Auswahl wurde so gewählt, dass lediglich Attribute enthalten sind, die für eine Visualisierung der Daten in einer Karte interessant sind. Eine vollständige Liste aller Attribute, sowie eine detailliertere Erläuterung der Attribute sind in der offiziellen Dokumentation des Datensatzes in [VG-13] zu finden.

Es ist zu beachten, dass die Attribute der Verwaltungsgebiete in der kompakten Version des Datensatzes zum Teil in eine gesonderte Tabelle ausgelagert sind, da die Verwaltungsgebiete höherer Verwaltungsebenen nicht direkt im Datensatz abgebildet sind, sondern über die Zusammenfassung kleinerer Flächen definiert werden.

Attribut	Beschreibung
USE	Beschreibt die unterste Verwaltungseinheit, der die Fläche zugehörig ist (1: Staat; 2: Bundesland; 3: Regierungsbezirk; 4: Kreis; 5: Verwaltungsgemeinschaft; 6: Gemeinde; 11: Bodensee Deutschland; 12: Bodensee Ausland)
RS	Regionalschlüssel. Er stellt die Verbindungen zwischen den verschiedenen Verwaltungsebenen dar.
GEN	Geographischer Name der Verwaltungseinheit
DES	Amtliche Bezeichnung der Verwaltungseinheit
NAMBILD	Binär. Beschreibt, ob DES dem Namen vorangestellt werden soll (beispielsweise 1 = “Kreis ...” oder 0 = “Salzlandkreis”)
DEBKG_ID	Verknüpfung zu DLM250 (Digitales Landschaftsmodell)
EWZ	Enthält die Einwohnerzahl des Statistischen Bundesamtes für die Verwaltungseinheit (nur in VG250-EW)

Tabelle 2.2.: Auswahl der Attribute der Verwaltungsgebiete des VG250 Datensatzes (vgl. [VG-13]).

2.2.2. Attribute der Verwaltungsgebietsgrenzen

Die Daten der Verwaltungsgebietsgrenzen enthalten den Verlauf der Grenzen zwischen Verwaltungsgebieten der Bundesrepublik Deutschland sowie Daten zum Verlauf von Küstenlinien innerhalb der Verwaltungseinheiten. Diese Küstenlinien werden durch den Attributwert 99 des Attributs *USE* identifiziert.

Die Grenzverläufe sind in Form von einzelnen Grenzsegmenten angegeben. Diese Grenzsegmente beschreiben den Verlauf einer Grenze zwischen zwei Grenz-Kreuzungspunkten. Für jedes Grenzsegment ist mittels des *USE*-Attributs festgelegt, zu welcher höchsten Verwaltungsebene das Segment gehört. Das heißt jedes Grenzsegment ist stets eine Verwaltungsgebietsgrenze auf der niedrigsten Ebene, der Gemeinde. Eine Teilmenge der Gemeindegrenzen bildet die Grenzen der nächsthöheren Verwaltungsebene, der Verwaltungsgemeinschaften. Wiederum eine Teilmenge dieser Verwaltungsgebietsgrenzen bildet die Grenzen der Kreise usw. *USE* bezeichnet die höchste Verwaltungsebene, der ein Grenzsegment zugehörig ist. Um beispielsweise alle Segmente der Staatsgrenze aus dem Datensatz auszulesen, genügt es die verfügbaren Grenzsegmente nach dem *USE*-Attribut mit dem Wert 1 (= Staatsgrenze) zu filtern, analog für die Bundeslandgrenzen (*USE* = 2) usw.

Das Attribut *LED* beschreibt den rechtlichen Status des Grenzsegments und unterscheidet zwischen Grenzsegmenten, deren Verlauf in einem Rechtsakt genau beschrieben sind (*ISN* = 1) und solchen, deren Verlauf rechtlich nicht verbindlich festgelegt ist (*ISN* = 2) (vgl. [VG-13]). Ein *ISN*-Wert von 9 definiert eine Küstenlinie, die keine Bedeutung für die Trennung zwischen Verwaltungsgebieten besitzt - im Datensatz ist dieses Attribut (mit einer Ausnahme bei 180 Vorkommnissen insgesamt) äquivalent zu dem *USE*-Wert 99, der Küstenlinien beschreibt.

Attribut	Beschreibung
USE	Beschreibt die höchste Verwaltungsebene, auf der die Grenze eine Verwaltungseinheit begrenzt (1: Staat; 2: Bundesland; 3: Regierungsbezirk; 4: Kreis; 5: Verwaltungsgemeinschaft; 6: Gemeinde; 99: Küstenlinie - trennt Land und Wasserfläche innerhalb einer Verwaltungseinheit)
LED	Rechtliche Definition des Grenzabschnitts (1: rechtlich festgelegte Grenze; 2: rechtlich **nicht** festgelegte Grenze; 9: Küstenlinie. Besitzt keine Bedeutung als Trennungslinie zwischen Verwaltungsgebieten

Tabelle 2.3.: Auswahl der Attribute der Grenzsegmente des VG250 Datensatzes (vgl. [VG-13]).

2.2.3. Format der Datensatzfiles

Der VG250 Datensatz ist auf der Homepage des Bundesamtes für Kartographie und Geodäsie (http://www.bkg.bund.de) unter der Rubrik OpenData verfügbar. Beide Versionen des Datensatzes können dort im SHAPE-Format heruntergeladen werden (Größe gezippt ca. 43MB (Kompakt) bzw. 74MB (Ebenen)). Der kompakte Datensatz besteht aus den beiden SHAPE-Teilen *VG250_F* und *VG250_L* sowie zwei Datenbanktabellen *VG250_NAM.DBF* und *VG250_ISN.DBF*. Die beiden letztgenannten enthalten die Attribute der Verwaltungsgebiete (Tabelle NAM) bzw. deren hierarchische

Struktur (Tabelle ISN). Die verbleibenden beiden Teile beschreiben die Verwaltungsflächen (*VG250_F*) und die Verwaltungsgebietsgrenzen (*VG250_L*). Beide Datensätze sind nach dem SHAPE-Standard aufgebaut und enthalten ein Main File (*.shp*), ein Index File (*.shx*) und eine Datenbank Tabelle (*.dbf*) sowie eine Projektdatei (*.prj*). Letzteres ist nicht Teil des SHAPE-Standards (vgl. [S-D98]). In dem Datensatz, der die Objekte nach Verwaltungsebenen sortiert speichert, sind die Flächengebiete (der *VG250_F*-Teil) für jede Verwaltungsebene separat gespeichert. Die Grenzverläufe sind im selben Format wie kompakten Datensatz enthalten.

Wie oben erwähnt wird in diesem Abschnitt insbesondere auf den Datensatz der Grenzsegmente eingegangen, der durch die Dateien vg250_l.xxx definiert ist. Die Flächendaten in den Dateien vg250_f.xxx entscheiden sich im Aufbau lediglich durch den verwendeten SHAPE-Datentyp, d.h. den Aufbau der Datenrecords des Main Files (siehe Abschnitt 2.2.3). Sie sind sonst analog aufgebaut.

Die Datei vg250_l.shp aus dem VG250-Datensatz enthält die Verläufe der Verwaltungsgebietsgrenzen in einer binären Kodierung. Der Aufbau dieser Datei wird im Folgenden beschrieben. Dabei muss beachtet werden, dass SHAPE-Dateien im Allgemeinen verschiedene Typen von geometrischen Objekten beschreiben können (z.B. Point, Multipoint, Polyline) (vgl. [S-D98]). Da das vorliegende SHAPE-File den Typ *Polyline* enthält, wird im Folgenden auf diesen SHAPE-Typ eingegangen und der Aufbau der anderen Typen nicht weiter vertieft.

Generell gilt, dass Numerische Werte nach dem SHAPE-Standard entweder als Signed 32-Bit Integer mit 4 Bytes Größe oder in Form eines Signed 64-Bit IEEE Double-Precision Floating Point Number mit 8 Bytes Größe abgelegt sind. Die Byte-Order der Kodierung ist innerhalb der Dateien unterschiedlich, entweder Big-Endian oder Little-Endian. Welche der beiden Optionen für ein Attribut zutrifft ist in den Abbildungen gesondert vermerkt.

Die .shp-Datei besteht aus einem Header mit fester Länge, der in Abschnitt 2.2.3 beschrieben ist. Darauf folgen beliebig viele Records (bestehend aus Record Header und Record Body), die jeweils ein SHAPE-Objekt beschreiben. Der Aufbau der Records ist ebenfalls in Abschnitt 2.2.3 beschrieben. Neben der Zuordnung der Bits zu den entsprechenden Informationen ist beim Parsen insbesondere die Byte Order der Kodierung zu beachten. Diese unterscheidet sich bei unterschiedlichen Datenfeldern innerhalb der Datei.

Der Dateiheader

Der Dateiheader hat eine feste Länge von 100 Byte. Die Belegung der einzelnen Positionen und deren Byte-Order ist in Abbildung 2.1 dargestellt. Die Dateilänge ist ab Byte 24 als Länge der gesamten Datei in 16-Bit Worten angegeben, sie umfasst auch den Header selbst. Der SHAPE-Type ab Byte 32 beschreibt den SHAPE-Type der folgenden Objekte, in diesem Fall Polyline (Attributwert “3”). Hier ist zu beachten, dass die SHAPE-Datei immer nur Objekte eines SHAPE-Typs enthält. Die einzige Ausnahme bilden Null-Shapes, diese dürfen in jeder Datei eingefügt werden.

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
Byte 4	Unused	0	Integer	Big
Byte 8	Unused	0	Integer	Big
Byte 12	Unused	0	Integer	Big
Byte 16	Unused	0	Integer	Big
Byte 20	Unused	0	Integer	Big
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
Byte 52	Bounding Box	Xmax	Double	Little
Byte 60	Bounding Box	Ymax	Double	Little
Byte 68*	Bounding Box	Zmin	Double	Little
Byte 76*	Bounding Box	Zmax	Double	Little
Byte 84*	Bounding Box	Mmin	Double	Little
Byte 92*	Bounding Box	Mmax	Double	Little

* Unused, with value 0.0, if not Measured or Z type

Abbildung 2.1.: Aufbau SHAPE-File Header (Quelle: [S-D98]).

Position	Field	Value	Type	Byte Order
Byte 0	Record Number	Record Number	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

Abbildung 2.2.: Aufbau SHAPE-File Record Header (Quelle: [S-D98]).

Der Datenrecord

Nach dem Dateiheader folgt eine beliebige Menge an Objekt-Records, die alle nach folgendem festen Muster aufgebaut sind. Sie beginnen mit einem 8 Byte langen Record Header (siehe Abbildung 2.2), der in Byte 0 bis 3 die laufende Nummer des Objekts und von Byte 4 bis 7 die Länge des Record-Bodys enthält. Die Länge ist hier wieder in 16 Bit Wörtern angegeben, allerdings ohne den Record-Header. Die Gesamtlänge eines Records ergibt sich also zu 4 + die angegebene Länge des Records.

Der SHAPE-Typ des VG250_L Datensatzes ist Polyline. Eine Polyline beschreibt eine Folge von Punkten, die miteinander verbunden einen Linienzug, d.h. ein Grenzsegment, bilden. Ein SHAPE-Polyline-Objekt kann aus mehreren Teilen bestehen, wobei ein Teil immer aus einer Kette von verbundenen Punkten besteht. Der Aufbau des Record Bodys ist in Abbildung 2.3 dargestellt. Die enthaltenen Informationen sind in Tabelle 2.4 aufgelistet.

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	3	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Note: X = 44 + 4 ∗ NumParts					

Abbildung 2.3.: Aufbau SHAPE-File Record Body für den Datentyp Polyline (Quelle: [S-D98]).

Attribut	Beschreibung
Shape Type	Der Typ des SHAPE-Objekts (in diesem Fall null oder Polyline)
Box	Box: Die Kodierung der Bounding Box des Objekts (kodiert als 4 Double- Werte: Xmin, Ymin, Xmax, Ymax)
NumParts	Die Anzahl der Teile, aus denen das Objekt besteht
NumPoints	Die Anzahl der Punkte, die das Objekt definieren
Parts	Ein Integer-Array der Länge NumParts, das für jedes Teilobjekt den Index des Startpunktes in dem Point-Array enthält
Points	Ein Array bestehend aus 2 Double-Variablen pro Punkt, zwischen den einzelnen Werten ist kein Trennzeichen oder ähnliches vorhanden

Tabelle 2.4.: Beschreibung der Attribute des Bodys eines SHAPE Records (vgl. [S-D98])

Es ist zu beachten, dass die Informationen im Record Header im Big Endian Format angegeben sind, wohingegen die Informationen im Record Body im Little Endian Format gespeichert sind.

2.3. Das UTM Koordinatensystem

Zur Referenzierung der geographischen Objekte werden UTM-Koordinaten verwendet, das Koordinatensystem wird unter anderem in [UTM09] und [WP-13] beschrieben. Es unterteilt die Erde vertikal in 6° breite Streifen. Die erstellten Zonen werden ab dem 180. Längengrad West in östliche Richtung nummeriert (vgl. [UTM09] und [WP-13]). So ergibt sich für die Zone, in der der Großteil von Deutschland liegt, die Zone 32 (6° bis 12° östlicher Länge) und zu kleineren Teilen die Zonen 31 und 33 (vgl. dazu auch Abbildung 2.4).

In vertikaler Richtung werden die Zonen in 8° breite Zonenfelder unterteilt. Dabei wird bei 80° Süd begonnen. Den Zonenfeldern werden die Buchstaben des Alphabets zugeordnet, wobei bei C begonnen wird und die Buchstaben I und O ausgelassen werden, um eine Verwechslung mit den Ziffern 1 und 0 zu vermeiden. Das letzte Zonenfeld X ist mit 12° breiter als die restlichen Zonenfelder (vgl. [UTM09]

und [WP-13]). So ergibt sich, dass Deutschland in den Zonen 31, 32 und 33 und in den Zonenfeldern T und U liegt (siehe Abbildung 2.4).

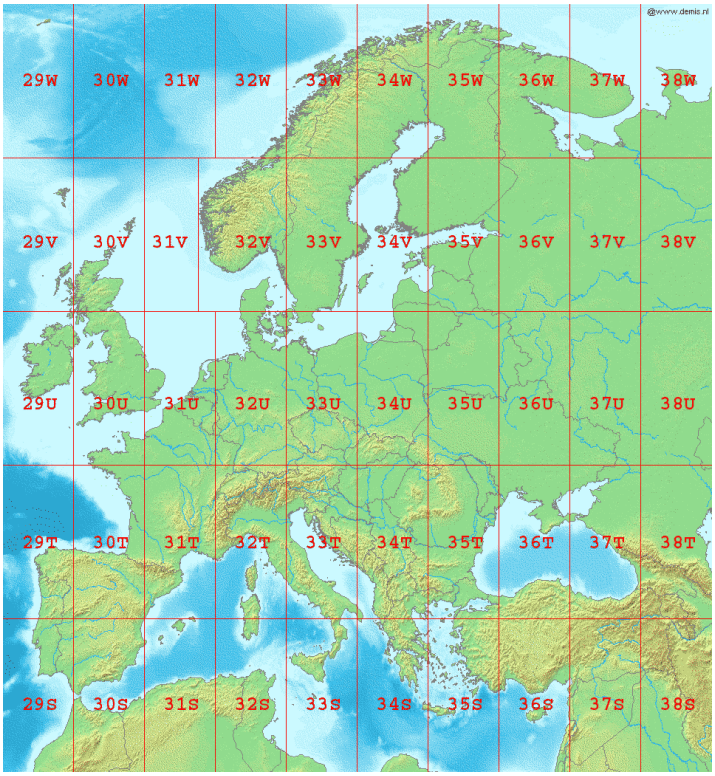

Abbildung 2.4.: UTM-Zonen in Europa (Quelle: https://de.wikipedia.org/wiki/Datei:LA2-Europe-UTM-zones.png).

Um Orte in der entsprechenden Zone zu referenzieren wird die Zone "ausgerollt" und das so entstandene Band mit einem kartesischen Koordinatensystem überzogen. Der Äquator bildet die X-Achse und der Mittelmeridian die Y-Achse des Koordinatensystems.

Um negative Werte bei der Beschreibung der Punkte zu vermeiden, werden von der X-Koordinate der Punkte 500.000m abgezogen, der Mittelmeridian erhält also die X-Koordinate 500.000m. Somit werden alle Orte in der Zone auf X-Werte zwischen 100.000m und 899.999m abgebildet (vgl. [UTM09] und [WP-13]).

Werden Orte auf der Nordhalbkugel beschrieben, so wird der Äquator mit dem Y-Wert 0 angenommen. Um negative Werte auf der Südhalbkugel zu vermeiden, werden den dort liegenden Punkten 10.000.000m von der Y-Koordinate abgezogen, wodurch sie ebenfalls ausschließlich positive Werte erhalten (vgl. [UTM09] und [WP-13]).

Auch Orte außerhalb einer bestimmten Zone können in dem Koordinatensystem der Zone dargestellt werden, allerdings wird dadurch eine zunehmende Verzerrung in Kauf genommen (vgl. [WP-13]). Die in dieser Arbeit verwendeten geographischen Objekte werden alle im Koordinatensystem der 32. Zone referenziert.

3. Anforderungen an die Datenstruktur und die Visualisierung

Die im vorigen Kapitel beschriebenen geographischen Basisdaten umfassen für Deutschland 45.037 Ortsnamen sowie 36.783 Grenzsegmente, die durch insgesamt 1.021.246 einzelne Punkte definiert sind (dabei sind die Start- und Endpunkte der Segmente mehrfach gezählt). Die durchschnittliche Anzahl an Punkte pro Grenzsegment beträgt somit ca. 28. Die Darstellung dieser Menge von Daten birgt einige Probleme, die in der vorliegenden Arbeit durch verschiedene Hierarchisierungen, Klassifizierungen und Generalisierung der Daten gelöst wurden.

In dem vorliegenden Kapitel werden die grundlegenden Anforderungen an die Visualisierung der Daten beschrieben (Abschnitt 3.1) und Probleme abgeleitet, die sich in Bezug auf die Datenstruktur und Visualisierung der Ortsnamen (Abschnitt 3.2) als auch der Grenzsegmente (Abschnitt 3.3) ergeben.

In den folgenden Kapitel 4 und 5 wird beschrieben, welche Maßnahmen zur Strukturierung der Datensätze vorgenommen wurden, um die in diesem Kapitel hergeleiteten Anforderungen zu erfüllen. Kapitel 4 bezieht sich dabei auf den Datensatz der Ortsnamen, wohingegen in Kapitel 5 die Maßnahmen zur Strukturierung des Datensatzes der Grenzverläufe beschrieben werden.

3.1. Anforderungen an die Visualisierung

Die im vorhergehenden Kapitel vorgestellten Datensätze enthalten eine große Anzahl an Daten, die in der vorliegenden Arbeit visualisiert und somit erfassbar gemacht werden. Dabei gilt es einige Probleme zu vermeiden, durch die die Erfassung der dargestellten Daten behindert werden könnte.

Einige dieser Probleme entstehen durch die große Menge an Daten, die in den Datensätzen enthalten sind, andere entstehen im Zusammenhang mit den Interaktionsmöglichkeiten, die den Benutzer in die Lage versetzten, den Datensatz interaktiv zu erkunden. Die erwähnten Probleme determinieren die in dieser Arbeit entwickelten Lösungsansätze und werden daher hier gesondert eingeführt.

Aus der großen Datenmenge, die zu visualisieren ist, ergeben sich folgende Anforderungen:

A1.1: Die Menge der visualisierten Daten muss an die Darstellung angepasst werden können. Das heißt, wenn ein großer Abschnitt der Daten zu sehen ist, muss die Menge der dargestellten Daten verringert werden, damit die dargestellte Information noch wahrgenommen werden kann (vgl. dazu Abbildung 3.1 und 3.2).

A1.2: Es ist darauf zu achten, dass wichtige Datenobjekte (beispielsweise die Stadt Stuttgart) auch in großen Datenausschnitten noch sichtbar sind, während weniger wichtige Objekte (wie zum Beispiel der Stuttgarter Stadtteil Vaihingen) erst in kleineren Datenausschnitten dargestellt werden müssen.

A1.3: Um die Wahrnehmbarkeit einzelner Datenobjekte zu gewährleisten, dürfen sich die dargestellten Objekte (beispielsweise Städtenamen) nicht überschneiden.

Weitere Anforderungen an die Visualisierung ergeben sich aus der Interaktion des Nutzers mit den visualisierten Daten. Dem Nutzer werden folgende Interaktionsmöglichkeiten gewährt:

Verschieben (Translation) des aktuellen Datenausschnitts, d.h. bei gleichbleibender Zoomstufe und Rotation wird der sichtbare Datenausschnitt verschoben.

Zoom (Skalierung) des aktuellen Datenausschnitts, d.h. beim *Herein zoomen* wird ein kleinerer Datenausschnitt mit mehr Details dargestellt, beim *Heraus zoomen* wird ein größerer Datenausschnitt weniger detailreich dargestellt.

Rotation des aktuellen Datenausschnitts, d.h. die Ausrichtung des Datenausschnitts wird verändert, sodass sich Norden in der Visualisierung nicht mehr zentral oben befindet, sondern beispielsweise im rechten oberen Eck. Dabei werden Zoomstufe und der Mittelpunkt des sichtbaren Datenausschnitts beibehalten.

Aus den genannten Interaktionsmöglichkeiten ergeben sich weitere Anforderungen an die Visualisierung:

A2.1: Der Detailgrad der Visualisierung wird nur verändert, wenn die Größe des dargestellten Datenausschnitts geändert wird (d.h. bei der Interaktion Zoom).

A2.2: Es soll jederzeit gelten, dass Datenobjekte, die in einem gewissen Datenausschnitt sichtbar sind, in einem kleineren Datenausschnitt weiterhin sichtbar bleiben. Ein Beispiel ist der Zoom in einen Datenausschnitt hinein: Objekte, die auf einer Zoomstufe sichtbar sind, dürfen beim weiteren Hineinzoomen nicht wieder verschwinden.

A2.3: Bei der Rotation gelte, dass sich der Detailgrad der Visualisierung nicht ändert (folgt aus A2.1). Es muss auch nach der Rotation gelten, dass alle übrigen Bedingungen (insbesondere A1.3: konfliktfreie Darstellung) erfüllt sind.

Die hier festgelegten Anforderungen an die Visualisierung haben Auswirkungen auf die Verarbeitung der Ortsnamen und Grenzsegmente, die in den folgenden Abschnitten 3.2 in Bezug auf die Ortsnamen und 3.3 in Bezug auf die Verwaltungsgebietsgrenzen erläutert werden.

Unabhängig von der Visualisierung ist es notwendig, dass die Menge der Daten, die für die Visualisierung bearbeitet werden muss, so gering wie möglich gehalten wird. Damit wird der Aufwand zur Verarbeitung der Daten so gering wie möglich gehalten und somit die Dauer für die Verarbeitung minimiert. Es sollen also für die Visualisierung so wenige und so kleine Daten wie möglich bearbeitet werden müssen. Das bedeutet insbesondere, dass die nicht sichtbaren Objekte im Optimalfall bereits bei der Abfrage der Datenstruktur entfernt werden und nicht erst beim tatsächlichen Zeichnen.

3.2. Anforderungen an die Darstellung der Ortsnamen

Durch die allgemeinen Anforderungen an die Visualisierung (vgl. Abschnitt 3.1) ergeben sich verschiedene Anforderungen an die Darstellung der Ortsnamen. Wie in Abbildung 3.1 zu erkennen, ist insbesondere die in A1.1 definierte Notwendigkeit der **Reduktion der visualisierten Daten** notwendig. Diese muss in verschiedenen Stufen möglich sein (A2.1). Bei der Reduktion der Daten muss beachtet werden, dass weniger wichtige Orte zuerst, wichtigere Orte zuletzt entfernt werden. Es ist also eine gewisse **Hierarchie der Daten** zu definieren und im Reduktionsverfahren zu berücksichtigen. Diese Anforderung folgt aus A1.2. Die Hierarchie und das darüber definierte Verfahren der Reduktion müssen gewährleisten, dass eine **Konsistenz der Daten bei der Reduktion** erfüllt ist (Anforderung A2.2). Das bedeutet, dass Datenpunkte, die in einer Reduktionsstufe entfernt wurden, bei einer weiteren Reduktion der Daten nicht erneut enthalten sein dürfen.

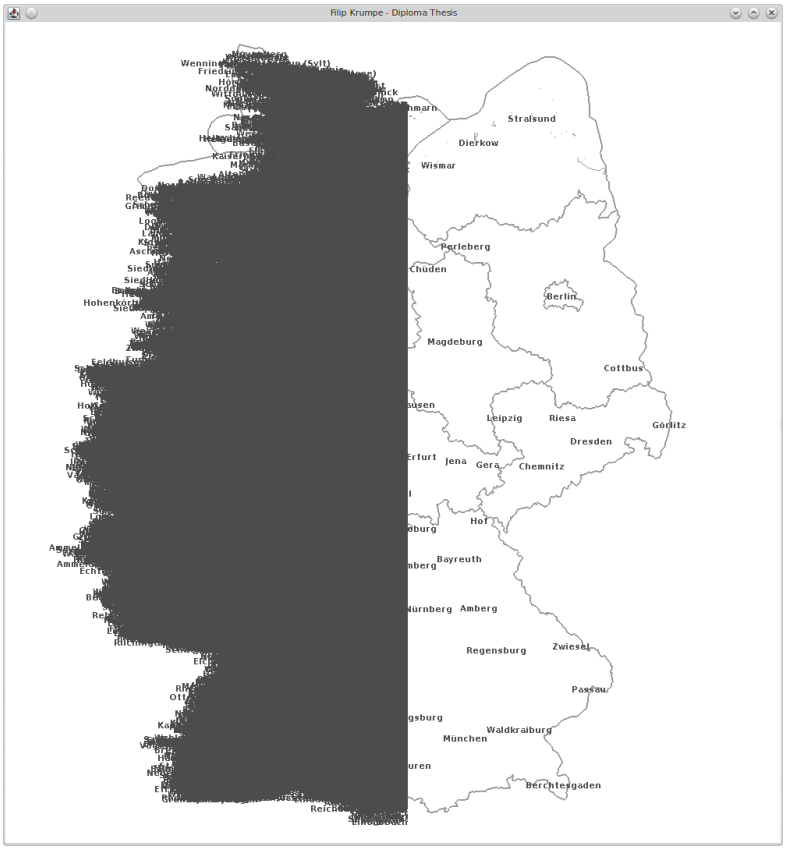

Abbildung 3.1.: Vergleich der Darstellung der Ortsnamen mit Filter (rechts) und ohne Filter (links).

Eine besondere Herausforderung beim Datensatz der Ortsnamen stellt die Darstellung der Datenpunkte mit Hilfe von Labels in einer Karte dar. Für diese Darstellung muss gewährleistet werden, dass sie Anforderung A1.3 erfüllt. Für einen beliebigen Datenausschnitt muss also eine **konfliktfreie Darstellung** und somit eine überdeckungsfreie Visualisierung der Ortsnamen möglich sein.

Bei der Darstellung der Ortsnamen als Label in der Karte ist vor allem die, in A2.3 geforderte, **Konsistenz bei Rotation** zentral. Es ist also notwendig, dass ein Datenausschnitt in jeder beliebigen Rotation alle Bedingungen aus Abschnitt 3.1 erfüllt (was insbesondere die konfliktfreie, d.h. überdeckungsfreie Darstellung einschließt), ohne dass dafür eine weitere Reduktion der Daten notwendig ist.

3.3. Anforderungen an die Darstellung der Verwaltungsgebietsgrenzen

Die allgemeinen Anforderungen an die Visualisierung aus Abschnitt 3.1 führen zu speziellen Anforderungen an die Darstellung der Verwaltungsgebietsgrenzen, die im Folgenden erläutert werden.

Die Besonderheit im Fall der Verwaltungsgebietsgrenzen ist, dass ihre Darstellung in Form von Linien auf natürliche Weise konsistent bei Rotation ist, das heißt, ein Grenzsegment kann nicht durch Rotation plötzlich durch ein anderes Segment überdeckt werden, da alle anderen Segmente ebenfalls von der Rotation betroffen sind (betrifft Anforderung A2.3). Ebenfalls trivial sind die Anforderungen bezüglich der Überdeckung von Objekten (A1.3), da die Grenzsegmente, im Gegensatz zu den Labels der Ortsnamen, ausschließlich den Raum ihres Verlaufs einnehmen und sich so gegenseitig nicht überdecken können.

Aus den Anforderungen in Abschnitt 3.1 ergibt sich für die Darstellung der Verwaltungsgebiete, dass eine **Reduktion der Daten** (Anforderung A1.1) ermöglicht werden muss, um die Wahrnehmbarkeit der dargestellten Informationen zu ermöglichen (vgl. Abbildung 3.2). Diese Reduktion muss schrittweise möglich sein (A2.1), wobei unwichtige Grenzsegmente zuerst und wichtige Grenzsegmente zuletzt reduziert werden (A1.2). Eine **Hierarchisierung der Daten** ist demnach notwendig. E ist darauf zu achten, dass Objekte, die bei der Reduktion entfernt wurden, bei der weiteren Reduzierung der Daten nicht erneut erscheinen (Anforderung A2.2). Es muss also eine **Konsistenz der Daten bei der Reduktion** gewährleistet sein.

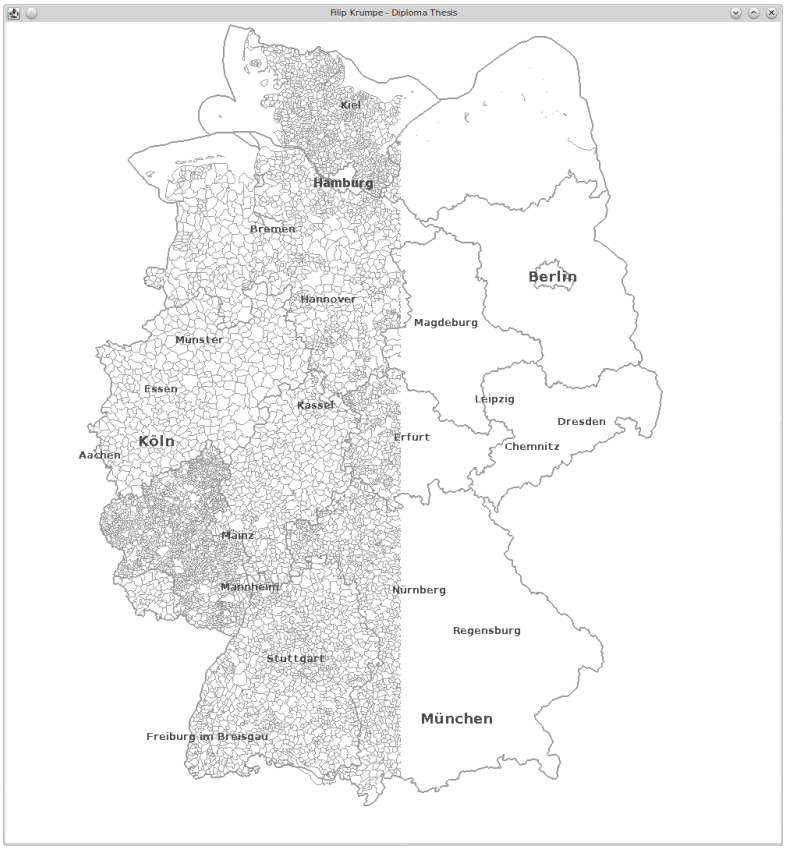

Abbildung 3.2.: Vergleich der Darstellung des Datensatzes der Verwaltungsgebietsgrenzen ohne (links) und mit Filter (rechts).

4. Vorverarbeitung und Strukturierung der Ortsnamen

Im vorherigen Kapitel 3 wurden verschiedene Anforderungen an die Darstellung der Ortsnamen erarbeitet (vgl. Abschnitt 3.2). Diese Anforderungen werden im vorliegenden Kapitel aufgegriffen, um daraus eine Datenstruktur für die Verwaltung der Ortslagen zu entwickeln. Eine zentrale Anforderung ist, dass die Ableitung einer konfliktfrei darstellbaren Teilmenge der gesamten Menge der Ortslagen möglich sein muss. Ein ähnliches Problem ist auch als das Problem des Point-Feature Labeling Placement bekannt, es wird in Abschnitt 4.1 erörtert. Dabei werden Unterschiede zwischen beiden Problemstellungen herausgearbeitet, die die Anwendung der verschiedenen Lösungsmöglichkeiten für das Point-Feature Labeling Placement in der vorliegenden Arbeit verhindern. Daraufhin wird in Abschnitt 4.2 die High-Level Idee für eine Strukturierung der Daten entwickelt, die eine Ableitung von Datenausschnitten ermöglicht, die die gegebenen Anforderungen erfüllen. Wie diese Datenstruktur aus dem GN250-Datensatz (vgl. Abschnitt 2.1) erstellt werden kann, ist in Abschnitt 4.3 beschrieben. Im Rahmen der vorliegenden Arbeit wurde das im folgenden beschriebene Strukturierungsverfahren implementiert und die so strukturierten Daten visualisiert. Die genutzte binäre Kodierung zur Speicherung der Datenstruktur wird in Abschnitt 4.4 beschrieben.

In Abschnitt 3.2 wurden folgende Anforderungen an den Datensatz abgeleitet:

1. **Reduktion der Daten:** Es soll möglich sein, die Menge der Daten in verschiedenen Stufen zu reduzieren.
2. **Hierarchie der Daten:** Bei der Reduktion sollen weniger wichtige Objekte zuerst und wichtigere Objekte zuletzt entfernt werden.
3. **Konsistenz der Daten bei der Reduktion:** Objekte, die in einer Reduktionsstufe entfernt wurden, dürfen bei einer weiteren Reduktion nicht wieder enthalten sein.
4. **Konfliktfreie Darstellung:** Die Ableitung einer Teilmenge der Objekte, die überdeckungsfrei darstellbar sind, soll möglich sein.
5. **Konsistenz bei Rotation:** Die abgeleiteten, überdeckungsfrei darstellbaren Objekte, sollen auch bei Rotation weiterhin überdeckungsfrei darstellbar sein.

Um ein einheitliches Verständnis der hier genutzten Begrifflichkeiten herzustellen, werden im Folgenden die verwendeten Begriffe bezüglich des Datensatzes der Ortsnamen kurz erläutert. *Datenpunkte* bezeichnen die im Datensatz vorhandenen Orte mit ihrer entsprechenden Position. Jedem Datenpunkt ist ein *Ortsname* zugeordnet, welcher eine bestimmte *Länge* (notiert als $|Name|$) besitzt. Die Länge eines Ortsnamen beschreibt die Anzahl der Buchstaben des Namens. Ein Ortsname wird als *Ortslabel*

in der Visualisierung dargestellt. Verschiedene Datenpunkte besitzen eine feste *Distanz* zueinander, welche durch ihre jeweilige Position eindeutig bestimmt ist.

4.1. Das Point-Feature Labeling Placement Problem

Das Problem des Point-Feature Label Placement (PFLP) beschreibt ein Standardproblem in der Kartographie und wurde bereits 1962 von Eduard Imhof beschrieben (vgl. [Imh62, CMS95]). Das PFLP ist auf Labels von einzelnen Datenpunkten beschränkt, im Gegensatz zum Area-Feature Labeling Placement, das die Beschriftung von Flächen betrifft, und zum Line-Feature Label Placement, das die Beschriftung von Linien beschreibt. Beim PFLP wird eine optimale Platzierung für Labels von Datenpunkten in einer Visualisierung gesucht. Für jedes Label sind verschiedene Positionen um den eigentlichen Datenpunkt möglich. In Abbildung 4.1 ist ein mögliches Set von diskreten Positionen und der Wert der einzelnen Positionen dargestellt. Niedrigere Werte kennzeichnen hier zu bevorzugende Positionen des Labels (vgl. [CMS95]).

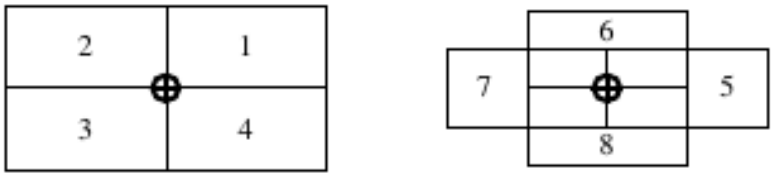

Abbildung 4.1.: Mögliche Positionen eines Labels in einer Visualisierung von Datenpunkten. Niedrigere Werte der Position entsprechen der bevorzugten Position (Quelle: [CMS95]).

Verschiedene wissenschaftliche Arbeiten befassen sich seit Ende der 1980er Jahre mit dem Optimierungsproblem des PFLP und suchen nach Verfahren für die Approximation einer möglichst optimalen Positionierung von Label. Einen sehr guten Überblick über diese Arbeiten vermittelt die Arbeit von Herbert Freeman in [Fre05]. Im Allgemeinen lassen sich zwei Ansätze der Arbeiten unterscheiden: Während einige Arbeiten die Anzahl der darstellbaren Datenpunkte in einem gegebenen Setting (aus Bildgröße und Darstellung der Label) zu optimieren versuchen, suchen andere nach einer maximalen Schriftgröße der Label, bei der sich diese noch überschneidungsfrei darstellen lassen (vgl. [CMS95, Mot07, Wol99]. Die verschiedenen Arbeiten unterscheiden sich zum Beispiel in den möglichen Positionen, die ein Label einnehmen darf, sowie in den zu optimierenden Zielfunktionen (engl. optimization functions), die beschreiben, wie gut eine gegebene Beschriftung (engl. labeling) ist.

Für einige dieser Probleminstanzen wurde gezeigt, dass sie NP-Hart sind, das heißt deterministisch nicht in polynomieller Zeit zur Eingabe gelöst werden können (vgl. [CMS95]). Dazu zählt insbesondere das PFLP mit den vier möglichen Label Positionen, die links in Abbildung 4.1 dargestellt sind (vgl. [CMS95]). Die NP-Härte kann für viele andere Probleminstanzen ebenfalls gezeigt werden. Es ist demnach nicht zu erwarten große Probleminstanzen in kurzer Zeit optimal lösen zu können.

Zu Beginn der wissenschaftlichen Untersuchungen des PFLP stand die Suche nach einer möglichst optimalen Lösung im Fokus der Arbeiten. Die Suche nach Algorithmen, die möglichst zeiteffizient annehmbare Lösungen finden, ist erst seit einigen Jahren im Fokus der Wissenschaft (vgl. [Mot07]).

Ein Überblick über Lösungsansätze zur zeiteffizienten Berechnung von Label Placements findet sich in der Arbeit von Kevin D. Mote (siehe [Mot07]). Viele der dort beschriebenen Lösungen basieren auf einem Konfliktgraph, der mögliche Konflikte zwischen Labels beschreibt. Dieser Konfliktgraph muss mit einem erheblichen Aufwand vorberechnet werden. Ist der Konfliktgraph einmal berechnet, ermöglichen die Verfahren jedoch eine schnelle Ableitung guter Ergebnisse in relativ kurzer Zeit.

Einen Ansatz zur Ableitung von Point Labelings in Echtzeit, der ohne eine Vorverarbeitung der Daten auskommt, beschreibt Kevin D. Mote in [Mot07]. Der dort entwickelte Algorithmus ermöglicht die Berechnung von Labelings für große Datensätze in sehr kurzer Zeit (laut eigenen Angaben in ca. einer Sekunde für 75.000 Datenpunkte). Ein Nachteil dieses Ansatzes ist, dass alle Labels mit einer festen Größe angenommen werden. Weitere Nachteile, die der Algorithmus mit den oben erwähnten anderen Algorithmen gemeinsam hat, sind im folgenden Absatz beschrieben.

In Bezug auf die vorliegende Arbeit ist ein Nachteil der beschriebenen Ansätze, dass sie einige der in Abschnitt 3.2 beschriebenen Anforderungen an die Visualisierung nicht erfüllen. Insbesondere Anforderung 3, die *Konsistenz der Daten bei der Reduktion*, und Anforderung 5, die *Konsistenz bei Rotation*, können in den Ansätzen zumeist nicht garantiert werden. Zudem wird die Hierarchie der Datenpunkte bei den meisten Ansätzen nicht in die Berechnung der darzustellenden Datenpunkte einbezogen.

Um diese Anforderungen zu erfüllen, wurde in der vorliegenden Arbeit ein neuer Ansatz für die Ableitung einer konfliktfrei darstellbaren Menge der Datenpunkte entwickelt. Der Ansatz basiert auf einer Vorverarbeitung der Daten. Unter Voraussetzung der vorverarbeiteten Daten ist die Ableitung einer konfliktfreie darstellbaren Teilmenge der Datenpunkten, welche die in Abschnitt 3.2 definierten Anforderungen erfüllt, in Linearzeit möglich. Die entwickelte Datenstruktur wird im Folgenden beschrieben.

4.2. Strukturierung der Ortsnamen

Um die gegebenen Anforderungen (siehe Abschnitt 3.2 und Einleitung zu diesem Kapitel 4) zu erfüllen, wurde eine Hierarchisierung der Daten des GN250 Datensatzes durchgeführt (für eine Beschreibung des Datensatzes siehe Abschnitt 2.1). Durch diese Hierarchisierung ist es möglich die “Wichtigkeit” der einzelnen Objekte zu vergleichen und so die Anforderung der *Hierarchie der Daten* zu erfüllen. Eine Beschreibung der Hierarchisierung wird in Abschnitt 4.2.1 gegeben.

Über die einfache Hierarchie der Daten wären die Anforderung der *Reduktion der Daten* sowie der *Konsistenz der Daten bei der Reduktion* bereits realisierbar, es ist jedoch nur mit sehr hohem Aufwand möglich aus der Menge an Daten eine *konfliktfrei darstellbare* Teilmenge der Daten herzuleiten, die zusätzlich *konsistent bei Rotation* ist (vgl. Abschnitt 4.1). Aus diesem Grund wird im Abschnitt 4.2.2 die Idee einer Datenstruktur und eine Erweiterung dieser Idee (in Abschnitt 4.2.3) entwickelt, welche die Ableitung einer *konfliktfrei darstellbaren* Teilmenge der Daten ermöglicht.

4.2.1. Hierarchisierung der Ortsnamen

Für eine einfache Hierarchisierung der Datenpunkte wird die Einwohnerzahl der Datenpunkte verwendet. Diese kann im zugrundeliegenden GN250 Datensatz über das Attribut *EWZ_GER* ausgelesen werden (siehe Abschnitt 2.1 und vgl. [GN-13]). Bei einigen Datenpunkten, die im Datensatz als *VIRTUELL* gekennzeichnet sind, fehlt die Angabe der Einwohnerzahl. Diese Objekte beschreiben virtuelle Orte, die aus mehreren realen Gemeinden bestehen (siehe Abschnitt 2.1 und vgl. [GN-13]). Um diese Objekte ebenfalls in die Hierarchie einzufügenden, wurde ihnen als Einwohnerzahl die Summe der Einwohnerzahlen der enthaltenen Gemeinden zugewiesen, welche über den gemeinsamen Wert für das Attribut AGS identifiziert werden können (siehe Abschnitt 2.1 und vgl. [GN-13]).

Zusätzlich zur Hierarchisierung wurde eine Klassifizierung der Daten vorgenommen. Für die Klassifizierung wurden die Daten nach ihrer Einwohnerzahl in die Klassen *>500.000*, *>100.000*, *>20.000*, *>5.000*, *>1.000* und *<1.000* unterteilt.

Über die Klassifizierung wären in einem einfachen Verfahren die Anforderungen 1. bis 3. erfüllbar. Eine *Reduktion der Daten* könnte durchgeführt werden, indem alle Datenpunkte einer Klasse entfernt würden. Würde stets die niedrigsten Klasse zuerst entfernt, so wäre dabei auch die Anforderung der *Hierarchie der Daten* erfüllt, da kleinere Orte zuerst entfernt würden, größere erst danach. Da für jeden Datenpunkt genau in einer Klasse enthalten ist, ist Anforderung 3 ebenfalls erfüllt, da das Datenobjekt vor der Entfernung der Klasse stets im Ergebnis enthalten wäre, in weiteren Reduktionen nicht wieder.

Um zusätzlich die Anforderungen 4. (*Konfliktfreie Darstellung*) und 5. (*Konsistenz bei Rotation*) zu erfüllen, ist ein komplexeres Verfahren notwendig. Die High-Level Idee zu einem möglichen Verfahren ist im folgenden Abschnitt 4.2.2 beschrieben. Da das dort skizzierte Verfahren nicht alle Anforderungen erfüllt, wird es in Abschnitt 4.2.3 erweitert. Daran anschließend wird das Verfahren zur Ableitung einer entsprechenden Liste der Datenpunkte detailliert beschrieben (Abschnitt 4.3).

4.2.2. High-Level Idee der Labelliste

Das entwickelte Verfahren ermöglicht die Menge der Datenpunkte des GN250 Datensatzes der Ortslabel so vorzusortieren, dass aus der erstellten Liste in einem Durchlauf, also in Linearzeit, eine konfliktfrei darstellbare Teilmenge der Objekte abgeleitet werden kann.

Grundlage des Verfahrens sind folgende **Annahmen**:

1. Ein Ortslabel wird stets zentral über der Position eines Datenpunktes dargestellt.
2. Um einen Datenpunkt wird ein Kreis mit der Größe des Labels reserviert, der durch kein weiteres Label geschnitten werden darf.
3. Die Datenpunkte können (im gegebenen Fall abhängig von der Einwohnerzahl der Orte) in eine Ordnung gebracht werden (siehe Abschnitt 4.2.1).
4. Eine Klassifizierung der Datenpunkte ist möglich (siehe Abschnitt 4.2.1).

Annahme 1 ist zielführend, da ein Ortslabel auf diese Weise bei der Rotation nur eine minimal große, kreisförmige Fläche um die Position des Datenpunktes überdecken kann. Durch **Annahme 2** wird gewährleistet, dass sich Ortsnamen auch nach einer beliebigen Rotation nicht überdecken. Auf diese Weise werden die Anforderungen 4. (*konfliktfreie Darstellung*) und 5. (*Konsistenz bei Rotation*) erfüllt. **Annahme 3** ist für die Erweiterung des Verfahrens notwendig. Sie ermöglicht es, dass Datenpunkte abhängig von ihrer Position in der hierarchischen Ordnung früher oder später in die Datenstruktur eingefügt werden und somit die entsprechenden Ortsnamen in der Visualisierung früher oder später sichtbar sind. **Annahme 4** ist für die Erweiterung des Verfahrens notwendig, um eine Berücksichtigung der Wichtigkeit der Datenpunkte bei der Ableitung einer konfliktfreien Teilmenge der Ortsnamen zu ermöglichen. Die beiden letzten Annahmen sind die Voraussetzung für die Erfüllung der Anforderung 2. (*Hierarchie der Daten*).

Die **Idee zur Datenstruktur** ist, dass die Objekte abhängig von ihren Abständen zueinander in eine eindeutige Reihenfolge gebracht werden. Durch die Reihenfolge wird für jeden Datenpunkt garantiert, dass alle in der Folge vor diesem Datenpunkt liegenden Datenpunkte konfliktfrei dargestellt werden können, wenn der gegebene Datenpunkt konfliktfrei dargestellt werden kann.

Die entwickelte Datenstruktur basiert auf einem speziellen Distanzmaß $d_B(P,\ Q)$ zwischen den zwei Datenpunkten P und Q. Dieses Maß bezeichnet die Distanz zwischen zwei Datenpunkten, allerdings nicht die euklidische Distanz (im Folgenden als $d(P,\ Q)$ bezeichnet), sondern die Distanz relativ zur Länge der Label der beiden Datenpunkte. Die Distanz bzgl. d_B zwischen zwei Datenpunkten entspricht somit der maximalen Länge, die ein Buchstabe der beiden Ortslabel einnehmen kann, ohne dass sich die beiden Label in der Darstellung überschneiden.

Folgende **Beobachtungen** bezüglich d_B gelten:

1. Die Distanz $d_B(P,\ Q)$ der Punkte P und Q kann mittels folgender Formel berechnet werden: $d_B(P,\ Q)\ =\ \frac{d(P,\ Q)}{0{,}5\ *\ (|P.Name|+|Q.Name|)}$
2. Die Distanz $d_B(P,\ Q)$ projiziert die Darstellung der Ortslabel auf die Ebene der Datenpunkte.
3. Die Distanz $d_B(P,\ Q)$ zweier Punkte ist unabhängig von der tatsächlichen Darstellung (z.B. Schriftgröße und Schriftart) der Ortslabel.
4. Bei jeder Darstellung kann aus der Schriftgröße der Ortslabel in Pixel und dem Skalierungsfaktor (die Distanz, die durch einen Pixel repräsentiert wird), die Länge berechnet werden, die durch einen Buchstaben belegt wird.
5. Für eine Menge M von Datenpunkten und einen Datenpunkt S, der nicht in der Menge enthalten ist, gibt es einen Punkt T aus der Menge M, sodass $d_B(S,\ T)$ minimal ist.

Mit dem gegebenen Distanzmaß, den daraus resultierenden Beobachtungen und den oben gemachten Annahmen, lässt sich folgende **Aussage** herleiten:

- Ein Datenpunkt kann konfliktfrei dargestellt werden, wenn die Minimale Distanz $d_B(P,\ Q)$ zu den bereits dargestellten Punkten kleiner ist als die Länge eines Buchstabens.

Aus den oben gemachten Annahmen und Beobachtungen folgt folgende Idee für den Aufbau der geforderten Datenstruktur. Die Datenstruktur ist definiert als eine Liste von Datenpunkten, wobei Folgendes für einen beliebigen Punkt P aus der Datenstruktur gilt:

- Sei O die Menge der Punkte, die in der Datenstruktur oberhalb von P liegen und U die Menge der Punkte, die unterhalb von P liegen.
- Sei $\delta_P U$ die minimale Distanz bzgl. d_B zwischen dem Punkt P und den Punkten aus U, $\delta_U O$ die minimale Distanz bzgl. d_B, die zwischen den Punkten aus U und den Punkten aus O besteht.
- Es gilt $\delta_P U \ \geq \ \delta_U O$.
- Sei δ_P die minimale Distanz bzgl. d_B zwischen einem Punkt P und den Punkten der Menge O.

Die Datenstruktur besteht also aus einer Folge von Datenpunkten. Zu jedem Datenpunkt P ist die minimale Distanz bzgl. d_B zu allen Datenpunkten, die in der Folge vor dem P liegen, gespeichert. Es gilt immer für einen beliebigen Punkt Q, der in der Folge unter P liegt, dass $\delta_P \geq \delta_Q$ ist.

Ist die Datenstruktur auf diese Art festgelegt, so kann, mit Hilfe der berechneten Länge der Darstellung eines Buchstabens, die Liste der Datenpunkte durchgegangen werden und alle Datenpunkte mit einem größeren minimalen Distanz δ_P zur Menge der darzustellenden Datenpunkte hinzugefügt werden. Die abgeleitete Teilmenge des Datensatzes kann in der Visualisierung konfliktfrei dargestellt werden.

4.2.3. Erweiterung des Verfahrens

Die oben entwickelte Datenstruktur erfüllt die erste Anforderung, es ist möglich die Menge an Daten stufenweise zu reduzieren, sogar eine stufenlose Reduktion ist möglich. Die Konsistenz der Daten bei Reduktion (Anforderung 3.) ist ebenfalls erfüllt. Ist ein Punkt P in einer Reduktionsstufe nicht enthalten, da seine minimale Distanz kleiner als der aktuelle Grenzwert ist, gilt für einen größeren Grenzwert, dass die minimale Distanz des Punktes P immer kleiner ist als dieser Grenzwert. Die Anforderungen 4. (konfliktfreie Darstellung) und 5. (Konsistenz bei Rotation) sind ebenfalls erfüllt (nach den Annahmen des Verfahrens), falls die Buchstaben der Ortslabel nicht größer dargestellt werden als der zur Abfrage genutzte Grenzwert.

Anforderung 2., die *Hierarchie der Daten* wird durch das Verfahren nicht erfüllt, da das Verfahren bisher immer den Datenpunkt mit der größten minimalen Distanz $d_B(P,\ Q)$ zur Liste hinzufügt, unabhängig von der Hierarchiestufe des Datenpunkts.

Um diese Bedingung ebenfalls zu erfüllen, wird die Datenstruktur wie folgt angepasst. Die Bedingung der absteigenden minimalen Distanz δ_P in der Liste der Datenpunkte wird aufgegeben. Die Datenstruktur wird stattdessen zuerst mit Datenpunkten der größten Wichtigkeitsklasse gefüllt, bis die Klasse zu einem bestimmten Grad in die Liste eingefügt wurde. Daraufhin werden Datenpunkte der nächsten Wichtigkeitsklasse zu den verbleibenden Datenpunkten hinzugenommen und in die Datenstruktur eingefügt (für eine detaillierte Beschreibung des Verfahrens der Generierung der Datenstruktur siehe auch Abschnitt 4.3).

Durch die beschriebene Maßnahme werden nach der Hinzunahme weiterer Datenpunkte neue Datenpunkte mit einer größeren als der bisher kleinsten minimalen Distanz δ_P in die Datenstruktur eingefügt, die Liste der Datenpunkte ist also nicht mehr streng absteigend nach der minimalen Distanz bzgl. d_B sortiert, dafür ist die Beachtung der Hierarchie der Daten (zumindest in Bezug auf die Hierarchieklassen) gewährleistet. Eine Ableitung einer konfliktfrei darstellbaren Teilmenge der Datensätze ist jedoch immer noch in einem Listendurchlauf (und somit in Linearzeit) möglich.

4.3. Generierung der Datenstruktur

In diesem Abschnitt wird das Verfahren zur Generierung der Datenstruktur der Ortsnamen genauer erläutert. Eine Beschreibung des Verfahrens in Pseudocode kann Listing 4.1 entnommen werden. Es ist zu beachten, dass der in Listing 4.1 beschriebene Algorithmus sehr kurz gehalten ist und so sehr viele Berechnungen mehrfach durchführt werden. Durch diese Maßnahme wird die Komplexität des Algorithmus auf ein Minimum beschränkt, was das Verständnis des Algorithmus erleichtern soll. Eine wesentlich effizientere Implementierung des Algorithmus ist jedoch möglich.

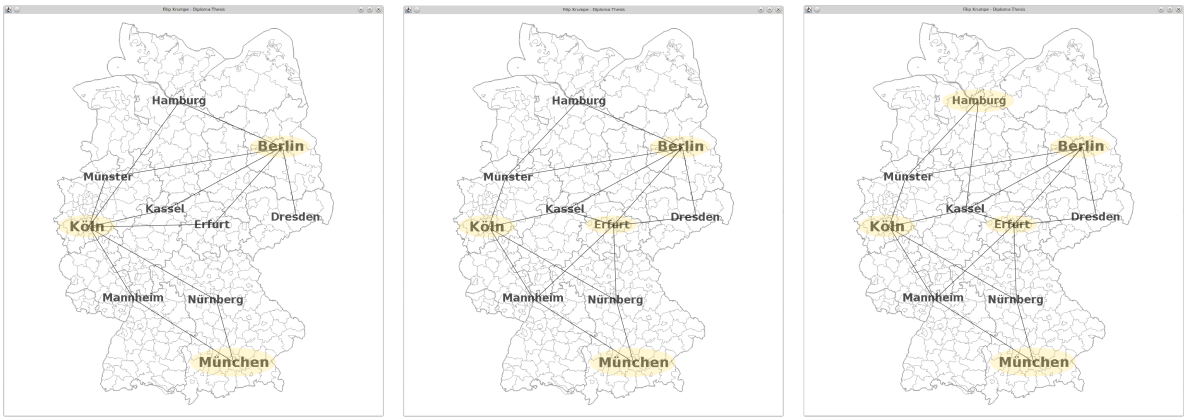

Abbildung 4.2.: Exemplarischer Ablauf des Algorithmus zur Generierung der Datenstruktur nach dem Berechnen der Distanzen im ersten (Links), zweiten (Mitte) und dritten (Rechts) Durchlauf. Gelb hinterlegt sind die Datenpunkte in der Ergebnismenge R, die schwarzen Linien stellen einige der betrachteten Distanzen dar.

Das Verfahren ist in Abbildung 4.2 beispielhaft, jeweils nach der Berechnung der Distanzen im ersten, zweiten und dritten Durchlauf der äußeren Schleife, dargestellt.

Die Eingabe für den Algorithmus ist eine Menge von Datenpunkten. Ein Datenpunkt besteht dabei mindestens aus einem Label (dem Ortsnamen), einer Lage (in X und Y-Koordinate) und einem Level, das die Klasse der "Wichtigkeit" des Datenpunktes angibt. Die Anzahl der Klassen der Wichtigkeit ist für die Funktion des Algorithmus' irrelevant. Die Datenpunkte des Level 0 sind sehr wichtig und die Seed-Werte des Verfahrens.

In jedem Durchlauf der Schleife wird getestet, ob die Menge der einzufügenden Datenpunkte S mindestens 20% ihrer ursprünglichen Größe besitzt (Zeile 11). Ist dies nicht der Fall, so werden die verbleibenden Datenpunkte in $SRemaining$ gesichert (Zeile 15) und die Datenpunkte des nächsten Levels zu S hinzugefügt (Zeile 17) und die Größe der Liste neu gesetzt.

Ab Zeile 20 beginnt die Suche nach dem Datenpunkt, der von den Datenpunkten in der Ergebnisliste R die größte minimale Distanz bezüglich d_B hat. Dazu wird in Zeile 22 ein Array initialisiert, das pro Datenpunkt aus S die minimale Distanz bzgl. d_B zu einem Punkt aus R enthält. Jeder Eintrag des Arrays wird mit dem maximalen Wert initialisiert.

Die folgende Schleife in den Zeilen 26 bis 31 führt zu einer bevorzugten Behandlung der verbleibenden Datenpunkte aus dem vorherigen Level. Der Teil des Algorithmus ist aber nicht unmittelbar für das Verständnis des Verfahrens notwendig und wird deshalb erst am Ende des Abschnitts erläutert.

In Zeile 33 werden die minimalen Distanzen bzgl. d_B zwischen allen Datenpunkten aus der Ergebnismenge R und der Menge S der einzufügenden Datenpunkte berechnet (Zeile 35). Ist die berechnete minimale Distanz kleiner als die aktuell im Array $minDelta$ gespeicherte, so wird der dort gespeicherte Wert überschrieben (Zeile 36). Wurden alle $deltaS$ der Paare von Datenpunkten aus S und R berechnet, dann enthält das Array $minDelta$ die minimalen Abstände bzgl. d_B der Datenpunkte in S und allen Datenpunkten in R. Der Datenpunkt mit dem größten Wert in $minDelta$ wird als nächstes zur Ergebnisliste R hinzugefügt (Zeilen 40 und 41). Der zuletzt hinzugefügte Datenpunkt wird nun aus den Mengen S und $SRemaining$ entfernt, falls es dort enthalten war (Zeilen 42 und 43). Ist die Menge S leer, so wurden alle verfügbaren Datenpunkte in die Datenstruktur R eingefügt und diese kann als Ergebnis zurückgegeben werden (Zeile 46).

Nach der Anforderung an die Datenstruktur der Ortsnamen (siehe Abschnitt 3.2 und Einleitung in Kapitel 4) ist die Beachtung die Hierarchie der Daten ein wichtiger Aspekt, den die Datenstruktur gewährleisten soll. Dieser Anforderung wird durch das schrittweise Einfügen der Klassen in die Menge S Rechnung getragen. Da die Datenpunkte des folgenden Klassifizierungslevels jedoch bereits nach der Verarbeitung von 80% der Datenpunkte des vorhergehenden Levels hinzugefügt werden, würde die Anforderung lediglich für 80% der Datenpunkte jeder Klasse gelten. Um das zu verhindern werden die verbleibenden Datenpunkte vor dem Einfügen der neuen Punkte separat gespeichert und bevorzugt behandelt. Diese bevorzugte Behandlung ist in den Zeilen 25 bis 34 des Pseudocodes in Listing 4.1 definiert und wird im Folgenden erläutert.

Das $minDelta$ Array beinhaltet, wie oben erwähnt, für jeden Datenpunkt aus S die minimalen Distanz bzgl. d_B des Datenpunktes zu den Datenpunkten in R. Die Schleife von Zeile 26 bis 31 bewirkt, dass die minimalen Distanzen der Datenpunkte aus S, die nicht in $SRemaining$ enthalten sind, zusätzlich mit den minimalen Distanzen zu den Datenpunkten aus $SRemaining$ verglichen werden. Die Schleife behandelt die Punkte in also so, als ob diese bereits in R eingefügt wären. Dieser Schritt bewirkt, dass aus der Menge der $S \setminus SRemaining$ keine Datenpunkte in R eingefügt werden, die potentiell die Datenpunkte aus $SRemaining$ verdecken könnten. Somit ist gewährleistet, dass keine Datenpunkte des niedrigeren Levels zur Datenstruktur hinzugefügt werden, die verhindern könnten, dass im weiteren Verlauf ein Datenpunkt aus $SRemaining$ zur Datenstruktur hinzugefügt wird. Die Anforderung 2. wird so erfüllt.

4.4. Speicherung der Daten

In diesem Abschnitt wird der Aufbau der Datei beschrieben, die die initiale Information über die Datenstruktur der Ortsnamen bereitstellt.

Die Datei der Ortsnamen enthält die Liste aller Datensätze der Ortsnamen in der Sortierung wie im vorigen Abschnitt beschrieben (siehe Abschnitt 4.2). Der Datensatz enthält die Daten in binärer Kodierung. Sämtliche ganzzahlige Werte sind als 16- oder 32-Bit Integer und Fließkommazahlen als

IEEE Double Wert mit 64-Bit im Little-Endian-Format enthalten. Ein Überblick über den Aufbau eines Datenrecords ist in Abbildung 4.3 zu finden.

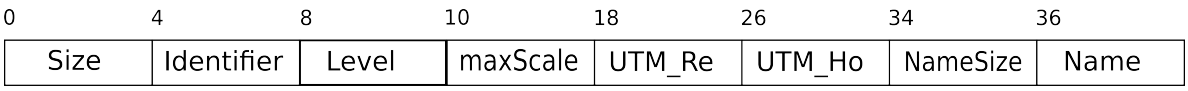

Abbildung 4.3.: Aufbau eines Datenobjekts der geographischen Namen

Ein Datensatz-Record enthält in den Bytes 0 bis 3 einen Integer, der die Größe des Records in Bytes definiert. Es folgt eine eindeutige Identifikationsnummer in den Bytes 4 bis 7. Dieser entspricht der ID im GN250-Datensatz, dem die Daten entstammen. Daran anschließend definieren die folgenden 2 Bytes ein 16-Bit Integer-Wert, das Level des Records. Die folgenden 8 Bytes in den Positionen 10 bis 17 beschreiben die minimale Distanz bezüglich d_B des Records, die für die Ableitung einer konfliktfrei darstellbaren Menge an Labels notwendig ist.Darauf folgen zwei Double Werte für den UTM-Rechtswert und den UTM-Hochwert zur Beschreibung der Position des Ortes des Records. Die UTM-Koordinaten beziehen sich auf die UTM-Zone 32 (siehe dazu Abschnitt 2.3). Der in Position 34 bis 35 definierte 16-Bit Integer-Wert beschreibt die Länge n des eigentlichen Ortsnamens in Bytes. Der Ortsname ist anschließend in den verbleibenden n Bytes ab Position 36 kodiert.

Listing 4.1 Algorithmus zur Generierung der Datenstruktur der Ortsnamen

```
1
2  funktion Generate_GN_Datastructure (listOfPoints)
3   R := listOfPoints.getElementsOfLevel(0); // the seed points
4   S := {};
5   currentLevel := 0;
6   sourceSize := 0;
7
8   do
9    // add new elements from the next level if the size of S is less than 20%
10   // of it's original size
11   if (S.size < 0.2 * sourceSize && currentLevel < maxLevel)
12    // increment the current level
13    currentLevel := currentLevel + 1;
14    // store the remaining elements in S
15    SRemaining := S;
16    // add elements of the new level to the source set
17    S := S ∪ listOfPoints.getElementsOfLevel(currentLevel));
18    sourceSize := S.size;
19   end if
20
21   // determine the next element that should be inserted into R
22   minDelta[] := {maxValue}; // stores the minimal distance per element
23
24   // calculate distance from remaining elements of the last to elements of the
25   // current level to prevent the former to be overlapped by the later
26   for r in SRemaining do
27    for s in S \ SRemaining do
28     deltaS = calculateDistanceB(s, r);
29     minDelta[s.index] := min(minDelta[s.index], deltaS);
30    end do
31   end do
32
33   for r in R do
34    for s in S ∪ SRemaining do
35     deltaS = calculateDistanceB(s, r);
36     minDelta[s.index] := min(minDelta[s.index], deltaS);
37    end do
38   end do
39
40   elementToAdd := elementWithMaxMinDelta();
41   R := R ∪ {elementToAdd};
42   S := S \ {elementToAdd};
43   SRemaining := SRemaining \ {elementToAdd};
44   while S.size > 0 do
45    return R;
46  end function
47
48  function calculateDistanceB(Element s, Element r)
49      return (euclidean distance between s and r) /
50            (0.5 * (s.labelSize + r.labelSize))
51  end function
```

5. Vorverarbeitung und Strukturierung der Grenzsegmente

Wie in Abschnitt 3.3 festgestellt, sind im Fall des Datensatzes der Grenzsegmente keine besonderen Maßnahmen notwendig, um die Überdeckung der einzelnen Datenobjekte (also Grenzsegmente) zu vermeiden. Zudem ist die Darstellung der Grenzsegmente immer konsistent bei Rotation. Es ist unmöglich, dass durch die Rotation Überdeckungen in der Darstellung auftreten. Unabhängig davon wurden in Abschnitt 3.3 dennoch folgende Anforderungen an die Darstellung der Grenzsegmente definiert:

1. **Reduktion der Daten:** Es soll möglich sein die vorhandenen Daten schrittweise zu reduzieren.
2. **Hierarchie der Daten:** Bei der Reduktion soll eine Hierarchie der Daten beachtet werden, sodass weniger wichtige Objekte zuerst und wichtigere Objekte zuletzt entfernt werden.
3. **Konsistenz der Daten bei der Reduktion:** Wurde ein Objekt während einer Reduktion entfernt, darf es bei einer weiteren Reduktion nicht wieder im reduzierten Datensatz auftauchen.

Im Allgemeinen wurde in Abschnitt 3.1 festgestellt, dass die verarbeiteten Daten so gering wie möglich gehalten werden sollen, um die Verarbeitungsdauer auf ein Minimum zu beschränken.

Im Verlauf dieses Kapitels wird zuerst eine einfache Klassifizierung der Daten vorgestellt (Abschnitt 5.1). Durch diese Strukturierung können die Bedingungen 1. bis 3. gewährleistet werden. Um die Menge der zu verarbeitenden Daten zu reduzieren, wird in Abschnitt 5.2 ein Algorithmus zur Generalisierung der Grenzsegmente vorgestellt. Durch diesen kann die Anzahl der Punkte, die ein Segment definieren, reduziert werden, ohne den groben Verlauf des Segments zu verändern.

Um Missverständnissen bei den verwendeten Begriffen vorzubeugen werden zuerst einige Begrifflichkeiten definiert, die im Verlauf des Kapitels verwendet werden. Wie in Abschnitt 2.2 beschrieben, sind die Daten des GN250 Datensatzes in Form von *Grenzsegmenten* definiert. Jedes dieser Grenzsegmente besteht aus einer Folge von Punkten, die den Verlauf des Segments beschreiben. Der erste und letzte Punkt eines Grenzsegments werden im Folgenden als *Endpunkte* des Grenzsegments bezeichnet. Die Punkte zwischen diesen Endpunkten seien die *inneren Punkte* des Segments. Zu jedem Segment ist im Datensatz die höchste Verwaltungsebene angegeben, der das Grenzsegment zugeordnet ist, dieses wird im Folgenden *Level* eines Segments genannt.

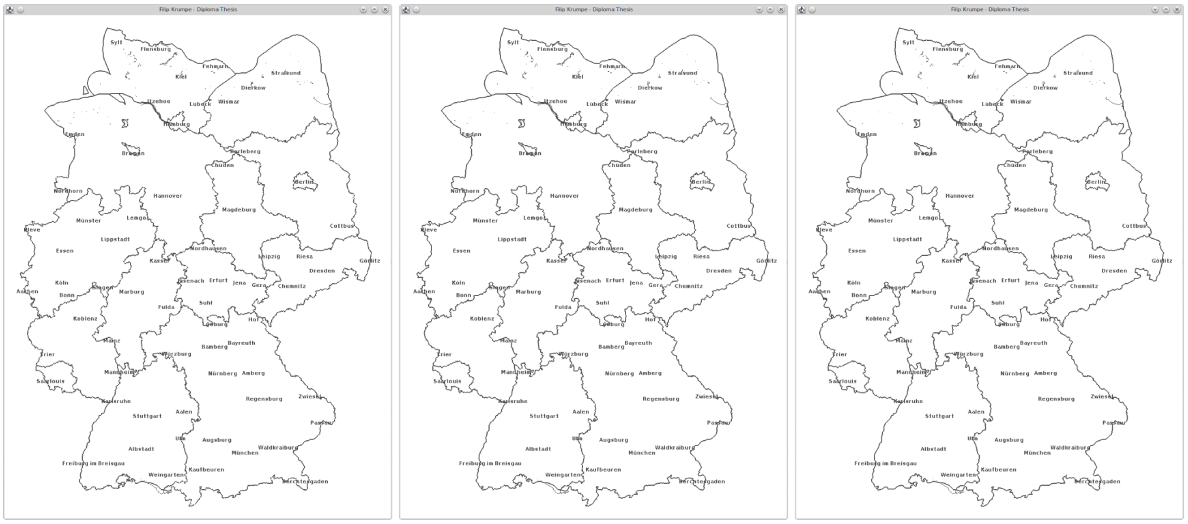

Abbildung 5.1.: Vergleich der Darstellung des Datensatzes mit verschiedenen Generalisierungsstufen (von links nach rechts: $\epsilon = 0$, $\epsilon = 500m$, $\epsilon = 1000m$) auf einer hohen Zoomstufe.

5.1. Strukturierung der Grenzsegmente

Eine Klassifizierung der Grenzsegmente ist bereits im GN250 Datensatz angelegt. Die einzelnen Segmente sind den verschiedenen Verwaltungsebenen der Bundesrepublik Deutschland zugeordnet, also den Klassen Staats-, Bundesland-, Regierungsbezirks-, Kreis-, Verwaltungsgemeinschafts- und Gemeindegrenze. Diese Klassen werden für die Reduktion der Daten verwendet.

Die Reduktion der Daten verläuft so, dass stufenweise die einzelnen Segmente der einzelnen Klassen aus dem Datensatz entfernt werden, damit ist Anforderung 1. (*stufenweise Reduktion der Daten*) erfüllt. Wird bei der Reduktion die niedrigste Verwaltungsebene zuerst, danach die darüber liegenden, entsprechend ihrer Reihenfolge, entfernt, so ist ebenfalls die Anforderung 2., die *Hierarchie der Daten* erfüllt. Indem bei der Reduktion immer die Klassen ebenfalls entfernt werden, die unter der zuletzt entfernten Klasse liegen, ist zudem die *Konsistenz der Daten bei der Reduktion* (Anforderung 3.) gewährleistet.

5.2. Generalisierung der Grenzsegmente

Wie in Abbildung 5.1 und Abbildung 5.2 zu sehen, ist es bei großen Zoomstufen ohne den Verlust sichtbarer Details möglich, die Genauigkeit der Grenzverläufe und somit die Anzahl der zu zeichnenden Punkte enorm zu reduzieren (vgl. Tabelle 5.1). Durch diese Maßnahme verringert sich sowohl die benötigte Zeit zur Abfrage der Daten bei einer Interaktion des Nutzers (vgl. Tabelle 5.1) als auch die benötigte Zeit für das Zeichnen der Grenzsegmente.

Um die Reduktion (oder auch Generalisierung) der Daten zu ermöglichen, ohne dabei die groben Details des Verlaufs der Grenzsegmente zu ändern, wird in der vorliegenden Arbeit der Douglas-

Epsilon	Anzahl Punkte	Speicherverbrauch
0	1.021.246	17.884.780 B
100m	333.789	6.885.468 B
200m	208.977	4.888.476 B
300m	159.356	4.094.540 B
400m	133.758	3.684.972 B
500m	118.042	3.433.516 B
600m	107.564	3.265.868 B
700m	100.409	3.151.388 B
800m	95.110	3.066.604 B
900m	91.067	3.001.916 B
1km	88.065	2.953.884 B

Tabelle 5.1.: Anzahl der Punkte pro Generalisierungsstufe, die für die Beschreibung aller Grenzsegmente der Verwaltungsgebiete der Bundesrepublik notwendig sind sowie Speicherverbrauch der Datenstruktur der Generalisierungsstufe in Bytes.

Peucker-Algorithmus verwendet. Der Algorithmus wird im folgenden Abschnitt 5.2.1 beschrieben. Daran anschließend ist in Abschnitt 5.3 der Aufbau der Datei zur Speicherung der Datenstruktur erläutert.

5.2.1. Der Douglas-Peucker-Algorithmus

Abbildung 5.2.: Vergleich der Darstellung des Datensatzes mit verschiedenen Generalisierungsstufen (von links nach rechts: $\epsilon = 0$, $\epsilon = 500m$, $\epsilon = 1000m$) auf einer kleinen Zoomstufe.

Der Douglas-Peucker Algorithmus (vgl. [DP73] und [Ebi02]) ist ein Algorithmus zur Vereinfachung von Liniensegmenten, die durch eine Menge von Punkten definiert sind. Die Besonderheit des Algorithmus ist, dass er die inneren Punkte eines Linienzugs nicht beliebig entfernt, sondern gezielt,

sodass grobe Details des Linienzugs erhalten bleiben. Ein Parameter ϵ legt dabei fest, bis zu welcher Abweichung vom eigentlichen Linienzug Punkte entfernt werden dürfen.

Listing 5.1 Douglas-Peucker-Algorithmus

```
1  funktion Douglas-Peucker(points, epsilon)
2   maxDistance := 0;
3   indexMaxDist := 0;
4   for i := 1 to line.size -2 do
5    if (distance(points[0], points[points.size - 1], points[i]) > maxDistance)
6     maxDistance := distance(points[0], points[line.size - 1], points[i]);
7     indexMaxDist := i;
8    end if
9   end do
10
11  if (maxDistance < epsilon)
12   return {points[0], points[points.size - 1};
13  else
14   return removeDoublePoint(
15    concat (Douglas-Peucker({points[0] ... points[i]}, epsilon),
16         Douglas-Peucker({points[i] ... points[points.size -1]}, epsilon)));
17  end if
18 end function
19
20 function distance(p1, p2, t)
21  return (|(x_2 - x_1)(y_1 - y_t) - (x_1 - x_t)(y_2 - y_1)|) /
22        (sqrt((x_2 - x_1)^2 + (y_2 - y_1)^2))
23
24 function concat(list1, list2)
25  return {list1[0 .. length-1], list2[0..length]};
```

Der Algorithmus (siehe Listing 5.1) geht wie folgt vor. Zuerst werden in den Zeilen 2 und 3 eine maximale Distanz, sowie der Index des Punktes mit der maximalen Distanz initialisiert. Die Distanz beschreibt hier den Abstand eines Punktes zur Geraden, die durch die beiden Endpunkte des Linienzugs verläuft (für die Berechnung der Distanz zwischen einem Punkt und einer Geraden siehe Abschnitt 5.2.1).

In den Zeilen 4 bis 9 wird für jeden inneren Punkt des Linienzugs die Entfernung zur Geraden durch die beiden Endpunkte berechnet (Zeile 5 und Abschnitt 5.2.1) und gespeichert, falls sie eine neue maximale Distanz definiert (Zeile 6 und 7).

Ist die maximale Distanz kleiner als ϵ, wird das gesamte Segment durch eine Gerade durch die beiden Endpunkte ersetzt (Zeile 12). Ist die maximale Distanz größer als ϵ, so wird der Algorithmus rekursiv auf die beiden Teilsegmente zwischen den Endpunkten und dem Punkt mit dem größten Abstand angewendet. Als Ergebnis wird die Verknüpfung der beiden Ergebnisse zurückgegeben. Bei der Verknüpfung der beiden Teilsegmente ist darauf zu achten, dass der Punkt mit der maximalen Distanz in beiden Teilsegmenten enthalten ist und deshalb einmal entfernt werden muss (siehe Funktion concat und Abschnitt 5.2.1).

Berechnung des Abstandes Punkt-Gerade

Um den Abstand $d(P, G)$ eines Punktes P zur Geraden G durch die Punkte $P1$ und $P2$ zu berechnen, wird folgende Abstandsformel (siehe Gleichung 5.1, vgl. [Wei14a]) verwendet. Die Distanzberechnung ist möglich, da das UTM Koordinatensystem ein kartesisches Koordinatensystem ist.

$$d(P,G) = \frac{|(x_2 - x_1)(y_1 - y_t) - (x_1 - x_t)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{5.1}$$

Verknüpfung der rekursiven Ergebnisse

Für den rekursiven Aufruf der Douglas-Peucker-Funktion wird die ursprüngliche Linie in zwei Segmente unterteilt: 0 bis i und i - Ende. Der Douglas-Peucker Algorithmus wird rekursiv mit beiden Teilsegmenten ausgeführt. Da sowohl Start als auch Endpunkt des Linienzugs auf jeden Fall im Ergebnis enthalten sind, ist der Punkt i beim Verknüpfen der Ergebnisse doppelt vorhanden, sodass ein Vorkommnis aus dem Ergebnisse entfernt werden muss.

5.3. Speicherung der Daten

Um die entwickelte Datenstruktur der Grenzsegmente zu speichern wurde ein spezielles binäres Dateiformat entwickelt, dessen Aufbau im Folgenden beschrieben wird. Der Aufbau orientiert sich grob am Aufbau der Dateien des VG250 Datensatzes (siehe 2.2).

Die Datei zur Speicherung der Grenzsegmentsdaten besteht aus einer Folge von Records, die jeweils ein Grenzsegment repräsentieren. Der Grenzverlauf wird, wie im ursprünglichen Datensatz, durch eine Folge von sich abwechselnden X- und Y-Koordinaten der enthaltenen Punkte beschrieben. Der Aufbau eines solchen Records ist in Abbildung 5.3 schematisch dargestellt. Es ist zu beachten, dass sämtliche ganzzahlige Werte sind als 16- oder 32-Bit Integer, Fließkommazahlen als IEEE Double Wert mit 64 Bit kodiert sind. Alle Zahlenwerte sind im Little-Endian-Format.

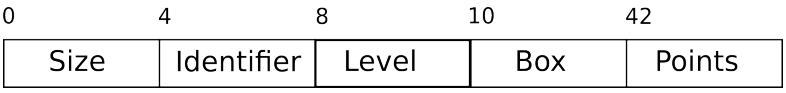

Abbildung 5.3.: Aufbau eines Records zur Codierung eines Grenzsegments

Die ersten 4 Bytes eines Grenzsegment-Records kodieren in einem Integer die Größe des gesamten Records. Daran anschließend enthalten die Bytes 4 bis 7 einen Integer Wert, die eindeutige ID des Segments, die mit der ID des Segments im ursprünglichen GN250-Datensatz übereinstimmt. Es folgt ein 16-Bit Integer in den Positionen 8 und 9, der das Level des kodierten Grenzsegments beschreibt (0: Staatsgrenze, 1: Bundeslandgrenze, 2: Regierungsbezirksgrenze, 3: Kreisgrenze, 4: Verwaltungsgemeinschaftsgrenze, 5: Gemeindegrenze und 6: Küstenlinie - was allerdings keine Grenze im rechtlichen Sinn darstellt (vgl. 2.2)).

Daran schließen 4 Double-Werte (X-Min, Y-Min, X-Max, Y-Max), mit jeweils 8- Byte, an, die die Bounding Box des Grenzsegments beschreiben. Die Koordinaten sind jeweils als Koordinaten im UTM-Koordinatensystem in der UTM-Zone 32 gegeben. Ebenfalls als UTM-Koordinaten in der 32. UTM-Zone sind die daran anschließenden Koordinaten der definierenden Punkte gegeben, die den verbleibenden Raum des Records einnehmen. Die Folge von Double-Werten beschreibt immer im Wechsel den X- und Y-Wert (also UTM-Rechts- und UTM-Hochwert) der Punkte, die, verbunden als Polyline, den Verlauf des Grenzsegments ergeben.

6. Die Suchstruktur und Abfrage der geographischen Basisdaten

Um die nach Kapitel 4 und 5 strukturierten Daten zu speichern, wurde in den Abschnitten 4.4 und 5.3 jeweils ein Dateiformat definiert. Mit Hilfe diese Dateiformate ist die persistenten Datenhaltung, also der Datenhaltung außerhalb von Anwendungen, ermöglicht. Diese persistente Datenhaltung wird ausführlich in Abschnitt 6.1 erläutert. Zur Laufzeit eines Programms ist es notwendig die Daten zusätzlich über weitere Strukturen zu verwalten, um beispielsweise eine einfache Abfrage der Daten innerhalb eines Datenfensters zu ermöglichen. Diese Datenhaltung zur Laufzeit ist im folgenden Abschnitt 6.2 beschrieben.

6.1. Die persistente Datenhaltung

Die persistente Datenhaltung der Daten der Ortsnamen geschieht durch eine Datei im Format, das in Abschnitt 4.4 beschrieben ist. Ein Auslesen der enthaltenen Daten kann durch zwei Methoden implementiert werden. Die erste Möglichkeit ist, alle Datenrecords sequentiell der Reihe nach einzulesen und dabei die nicht benötigten Daten zu verwerfen. Eine zweite Möglichkeit des Zugriffs kann durch eine direkte Adressierung der Datenrecords mittels Record Offsets realisiert werden. Dabei wird in einem ersten Durchlauf die Startposition jedes Records gespeichert, wodurch im weiteren Verlauf direkt auf die Records zugegriffen werden kann.

Ähnlich wie die Daten der Ortsnamen, können die Daten der Grenzsegmente verwaltet werden. Der Aufbau der entsprechenden Dateien ist in Abschnitt 5.3 beschrieben. Einen Unterschied bilden die verschiedenen Generalisierungsstufen der Daten. Um diese abzubilden, wurde im Rahmen der vorliegenden Arbeit mit verschiedenen Dateien zur Speicherung der Datensätze gearbeitet. Jede Datei speichert den gesamten Datensatz in einer bestimmten Generalisierungsstufe. Dabei werden ϵ Werte von 0m bis 1000m in 100m Schritten zur Approximation verwendet.

Im Fall der Grenzsegmentsdaten sind dieselben Zugriffsmöglichkeiten gegeben, wie beim Datensatz der Ortsnamen. Das sequentielle Zugreifen auf die Dateien ist jedoch mit einem größeren Aufwand verbunden, da die zu parsenden Records wesentlich umfangreicher sind und somit umsonst geparste Records einen großen Mehraufwand bedeuten. Beim direkten Zugriff auf die Records mittels Offsets entsteht das Problem, dass die einzelnen Records bei der Reduktion kleiner werden, somit sind die Offsets der Records in den Dateien der unterschiedlichen Generalisierungsstufen nicht gleich. Um den direkten Zugriff auf die jeweiligen Records eines Grenzsegments trotzdem zu ermöglichen, wurde zusätzlich zu den Datenfiles eine Index Datei angelegt, in der die verschiedenen Offsets der Records in der jeweiligen Generalisierungsstufe gespeichert sind. Die Index Datei enthält somit pro Grenzsegment 11 Offsets, jeweils einen für die verschiedenen Generalisierungslevel. Für den direkten Zugriff auf

einen Record einer bestimmten Zoomstufe kann in der Index-Datei der Offset nachgeschlagen und mit diesem der Zugriff auf den Record in der entsprechenden Datei durchgeführt werden.

6.2. Die Datenhaltung zur Laufzeit

Die persistente Datenhaltung, die in Abschnitt 6.1 beschrieben ist, ermöglicht die sichere Speicherung der geographischen Basisdaten während nicht auf diese Daten zugegriffen wird. Werden die Daten in einem verwendet, müssen weitere Zugriffsmöglichkeiten auf die Daten bereitgestellt werden. Mögliche Abfrageparameter sind:

- Ein Abfragefenster, innerhalb dem die zurückgegebenen Daten liegen sollen.
- Ein minimales Verwaltungslevel, dem die zurückgegebenen Grenzsegmente zugeordnet sein sollen.
- Eine Distanz δ, die die Größe eines Labelbuchstabens angibt. Die zurückgegebenen Daten sollen in diesem Setting konfliktfrei dargestellt werden können.
- Ein Generalisierungslevel, welches die Genauigkeit der zurückgegebenen Grenzsegmente angibt.

Um derartige Anfragen effizient bearbeiten zu können, wurde im Rahmen der vorliegenden Arbeit eine interne Verwaltung der geographischen Basisdaten in einem Grid implementiert. Das Grid verwaltet die geographischen Objekte in Form von Offsets der entsprechenden Records in den persistenten Dateien sowie verschiedene Attribute der Objekte. Über die Attribute kann direkt ermittelt werden, ob ein Objekt in der Ergebnismenge enthalten ist, oder nicht. Die so ermittelte Menge der Offsets wird in einem weiteren Schritt, dem Zugriff auf die tatsächlichen Records in den entsprechenden Dateien, in die eigentliche Ergebnismenge umgewandelt. Das Grid sowie die Abfrage der Daten im Grid wird im folgenden Abschnitt 6.2.1 beschrieben. In Abschnitt 6.2.2 wird erläutert, wie die Datenobjekte von den globalen UTM-Koordinaten, unter der Berücksichtigung von Rotation und Skalierung, in lokale Koordinaten der Visualisierung transformiert werden können.

6.2.1. Das Grid

Das Grid unterteilt die Fläche, die die Daten belegen, in eine feste Anzahl an Gridzellen. Jede dieser Zellen repräsentiert einen rechteckigen Bereich der Gesamtfläche und enthält eine Referenz (in Form eines Offsets) auf alle geographischen Objekte in diesem Bereich.

Um die Daten innerhalb eines gegebenen Abfragefensters zu ermitteln, werden die Gridzellen, in denen die Eckpunkte des Abfragefensters liegen, gesucht. Diese Gridzellen und alle zwischen ihnen liegenden Gridzellen werden nun entsprechend der Parameter abgefragt und die Gesamtmenge aller Ergebnisse als Ergebnis zurückgegeben. Eine Besonderheit ergibt sich bei den Daten der Grenzsegmente. Da diese eine räumliche Ausdehnung besitzen und nicht, wie die Ortsnamen, einfache Datenpunkte sind, ist es möglich, dass einzelne Segmente in verschiedenen Gridzellen liegen und somit mehrfach in

einer Ergebnismenge vorkommen können. Dieser Sonderfall muss bei der Verarbeitung der Anfragen beachtet werden.

Die Abfrage aller Objekte, die mindestens ein gegebenes Verwaltungslevel besitzen, macht eine erweiterte Verwaltung der Referenzen auf die Grenzsegmente innerhalb einer Gridzelle notwendig. Um derartige Anfragen effizient zu beantworten, werden die Referenzen der Grenzsegmente in einer Gridzelle nach ihrem entsprechenden Verwaltungslevel gruppiert. Bei der Abfrage mit einem minimalen Verwaltungslevel, dem die zurückgegebenen Grenzsegmente zugeordnet sein sollen, können direkt die Referenzen zurückgegeben werden, die dem minimalen und den darüber liegenden Verwaltungsleveln zugeordnet sind.

Da die Abfrage der Indexdatei zur Ermittlung der tatsächlichen Offsets eines Grenzsegments unnötigen Aufwand bedeutet, werden die Offsets der verschiedenen Generalisierungslevel direkt in den entsprechenden Gridzellen gespeichert. Auf diese Weise kann bei der Abfrage des Grids direkt der entsprechende Offset des Records zurückgegeben werden. Durch dieses Vorgehen steigt zwar der Speicherverbrauch des Grids, im Gegenzug ist jedoch eine effizientere Verarbeitung von Anfragen der Daten in einem gewissen Generalisierungslevel gegeben.

Zur Abfrage einer konfliktfrei darstellbaren Teilmenge der Ortslagen, muss als Abfrageparameter die entsprechende Größe δ eines Labelbuchstabens übergeben werden. Diese muss in einer Länge des globalen Koordinatensystems angegeben oder in eine solche umgerechnet werden. In der Gridzelle liegen die Referenzen sowie deren minimale Distanz bzgl. d_b (siehe Abschnitt 4.2.2) der Objekte in dem entsprechenden Bereich vor. Sind die Objekte nach der minimalen Distanz sortiert, reduziert sich die Abfrage auf einen Durchgang der Objektliste bis zu der Stelle, an der ein Objekt eine minimale Distanz < δ besitzt. Die darüber liegenden Objekte können als Ergebnis der Abfrage zurückgegeben werden.

6.2.2. Transformation der Daten

Die in den Datensatz-Files enthaltenen Daten beschreiben die Datenpunkte in globalen UTM-Koordinaten. Die Darstellung der Daten geschieht jedoch in einer Visualisierung mit einer bestimmten Große und Auflösung. Die globalen Koordinaten der Datenpunkte müssen für die Visualisierung in lokale Koordinaten umgerechnet werden.

Für die Transformation der Koordinaten werden als Parameter die x- und y-Koordinate des Mittelpunktes des dargestellten Bereichs in globalen Koordinaten benötigt. Ein Skalierungsfaktor beschreibt die Größe eines Pixels in globalen Koordinaten. Der Wert bezeichnet die Seitenlänge des Quadrats in globalen Koordinaten, das auf einen Pixel abgebildet wird. Ein Rotationswinkel α beschreibt den Winkel der Darstellung.

Von den globalen Koordinaten, die aus der Datei ausgelesen werden, wird der x- bzw. y-Koordinate des Mittelpunktes subtrahiert. Durch diese Operation wird das Koordinatensystem des Datenausschnitts so verschoben, dass der Nullpunkt zentral im Datenausschnitt liegt. Die so erhaltenen Koordinaten werden nun durch den Skalierungsfaktor dividiert, um die Koordinaten an das Koordinatensystem der Visualisierung anzupassen.

Da der Nullpunkt des Koordinatensystems im Datenausschnitt zentriert ist, ist es möglich, den Datenausschnitt mittels einer einfachen Rotationsmatrix zu rotieren, ohne dass sich der Datenausschnitt durch die Rotation verschiebt. Um das endgültige Ergebnis der Transformation zu erhalten, werden die Koordinaten aller Punkte mit der folgenden Rotationsmatrix rotiert (vgl. [Wei14b]):

$$\begin{bmatrix} cos\ \alpha & -sin\ \alpha \\ sin\ \alpha & cos\ \alpha \end{bmatrix}$$

Das Ergebnis der Umrechnung sind Koordinaten der berechneten Punkte in lokalen Koordinaten der Visualisierung. Die Daten können so direkt dargestellt werden ohne dass eine weitere Transformation der Daten vorgenommen werden muss. Es ist zu beachten, dass durch die Rotation des Datenausschnitts einige Teile in einen nicht sichtbaren Bereich, andere Teile, die ohne Rotation nicht sichtbar gewesen wären, in einen sichtbaren Bereich verschoben werden. Es können Bereiche an den Rändern entstehen, in denen keine Daten dargestellt werden, da für sie keine Daten abgefragt wurden. Dieses Phänomen kann jedoch verhindert werden, indem initial ein größerer als der eigentlich darstellbare Bereich der Daten abgefragt wird.

7. Implementierung und Messergebnisse

Im Rahmen der vorliegenden Diplomarbeit wurden zwei Visualisierungen der Daten implementiert. Sie basieren auf den hier entwickelten Datenstrukturen (vgl. Kapitel 4 und Kapitel 5) und der Suchstruktur über den Daten (vgl. Kapitel 6). In beiden Implementierungen wird die Visualisierungen mittels OpenGL bzw. OpenGL ES hardwarebeschleunigt dargestellt. Eine Visualisierung stellt die Daten als Java-Anwendung für PC dar, die zweite Visualisierung basiert auf dem Android-SDK und stellt die Daten auf Android Geräte dar.

Um in beiden Implementierungen der Visualisierung einen gemeinsamen Datenzugriff zu ermöglichen, wurde dieser in einer getrennten Komponente (dem StorageHelper) separat entwickelt und in die beiden Visualisierungen eingebunden. Die Operationen des StorageHelpers entsprechen den Zugriffsmethoden, die in Kapitel 6 beschrieben wurden. Die Funktionen des StorageHelper sind im folgenden Abschnitt 7.1 beschrieben.

In Abschnitt 7.2 werden verschiedene Aspekte der Implementierung der Visualisierung erläutert. Abschnitt 7.3 enthält schlussendlich verschiedene Messungen zur Geschwindigkeit der Datenstrukturzugriffe auf den unterschiedlichen Plattformen PC und Android Gerät.

7.1. Der StorageHelper

Der StorageHelper implementiert Funktionen für die Verwaltung und den Zugriff auf die Datensätze. Er implementiert dafür die in Kapitel 6 beschriebenen Zugriffsmethoden auf die Daten. Die Abfrage der Datenstruktur unterteilt sich in zwei Teile. In einem ersten Schritt wird eine interne Suchstruktur abgefragt, um festzustellen welche Datenobjekte tatsächlich dargestellt werden müssen. Dafür wird das in Abschnitt 6.2 beschriebene Grid genutzt. Mit Hilfe der erhaltenen Datenoffsets werden daraufhin, aus den persistenten Daten (siehe Abschnitt 6.1), die korrekten Datensätze ausgelesen.

Um nicht den gesamten Datensatz im Arbeitsspeicher lagern zu müssen, werden im StorageHelper jeweils nur die Offsets der einzelnen Datenpunkte gehalten. Zusätzlich sind Attribute der Datensätze direkt im StorageHelper gespeichert, die notwendig sind, um festzustellen, ob ein Element dargestellt werden kann oder nicht. Für die einzelnen Ortslagen wird zusätzlich die minimale Distanz bzgl. d_B (also die Distanz relativ zur Länge der entsprechenden Ortsnamen) zum nächsten möglichen Konfliktpunkt gespeichert. Durch diese Maßnahme kann bereits vor der Abfrage der eigentlichen Datensätze, festgestellt werden, ob das Label eines Datenpunktes konfliktfrei dargestellt werden kann oder nicht (vgl. 6.2.1).

Für die Grenzsegmente wird zusätzlich zum eigentlichen Offset die Verwaltungsebene gespeichert, um direkt Datensätze auszuschließen, die nicht dargestellt werden sollen. Anstelle von einem Offset

werden pro Grenzsegment mehrere Offsets gespeichert, um die Grenzsegmente in den verschiedenen Generalisierungsstufen direkt adressieren zu können (vgl. 6.2.1).

Zur Abfrage der Datensätze werden verschiedene Parameter genutzt, die sich für die Datensätze der Grenzsegmente und der Ortslagen teilweise unterscheiden. Für beide Datensätze ist als Parameter die Bounding Box der abzufragenden Daten anzugeben. Diese beschreibt den räumlichen Ausschnitt des Datensatzes, der in die Suche nach den entsprechenden Daten einbezogen werden soll. Ebenfalls für beide Datensätze gleich sind die Parameter für die Skalierung, der Mittelpunkt des Datenausschnitts sowie der Rotationswinkel. Mit Hilfe dieser Parameter werden die Koordinaten der Datensätze, direkt beim Auslesen aus den persistenten Daten, in lokale Bildkoordinaten überführt. Die Abfrage der Grenzsegmente benötigt zusätzlich zwei Parameter, zum einen ein Verwaltungslevel, dem die Grenzsegmente mindestens zugeordnet sein müssen, zum anderen eine Generalisierungsstufe, die die Detailstufe angibt, in der die Grenzsegmente ausgelesen werden sollen. Die Abfrage der Ortslagen benötigt als weiteren Parameter die Größe eines Labelbuchstabens in globalen Koordinaten. Die Größe lässt sich durch die eine Multiplikation der Größe der Buchstaben in Pixel mit dem Skalierungsfaktor berechnen.

Die Abfrage des StorageHelpers ergibt also eine Menge von Grenzsegmenten und Ortslagen, die auf die lokalen Bildkoordinaten abgebildet wurden. Diese Daten können in einem nächsten Schritt direkt gezeichnet werden. Die Darstellung der Daten in der PC-Anwendung und der Android-App ist im folgenden Abschnitt beschrieben.

7.2. Darstellung der geographischen Basisdaten

Die Darstellung der Daten, die in einem ersten Schritt mit Hilfe des StorageHelpers ausgelesen wurden, geschieht sowohl in der PC-Anwendung, als auch in der Android App hardwarebeschleunigt mittels OpenGL bzw. mittels OpenGL ES in der Android App.

Zur Darstellung der Grenzsegmente wird der Typ *GL_LINE_STRIP* verwendet, durch den eine Folge von Punkten, definiert über eine Folge von x- und y-Koordinaten der Punkte, verbunden als Liniensegment dargestellt wird. Um die Staats- und Bundeslandgrenzen zu unterscheiden, werden diese mit einer größeren Strichstärke gezeichnet.

Die Darstellung der Ortslagen geschieht durch einen Punkt, der die Lage des Ortes markiert, und dem Label, das den Ortsnamen darstellt. Das Label ist zentral über dem Punkt des Ortes platziert, um bei der Rotation eine minimale Fläche zu überdecken. Längere Ortsnamen werden maximal einmal umgebrochen, um den zu reservierenden Umkreis so klein wie möglich zu halten. Dieser Umbruch des Ortsnamens muss bereits im Vorhinein bei der Berechnung der Datenstruktur berücksichtigt werden.

Aufgrund der unterschiedlichen Performanz des PCs und des Android Geräts beim Zugriff auf die Datenstrukturen (vgl. Abschnitt 7.3), wurde die Datenabfrage bei der Interaktion auf den verschiedenen Plattformen unterschiedlich umgesetzt. Bei der PC-Anwendung ermöglicht die Geschwindigkeit der Datenabfrage, dass der komplette Datensatz während einer Benutzerinteraktion abgefragt und dargestellt wird. So ist es möglich auch während einer Interaktion immer die aktuellsten Daten anzuzeigen.

Bei dem Android-Testgerät ist die Geschwindigkeit sowohl bei der Abfrage des Grids als auch bei der Abfrage der Datensätze aus den Dateien nicht performant genug (vgl. Abschnitt 7.3). Um dem Nutzer eine direkte Interaktion mit dem Datenausschnitt zu ermöglichen, wurde aus diesem Grund bei der Interaktion jeweils der aktuelle Datenausschnitt manipuliert, ohne jeweils die Daten neu zu laden. Erst nach Abschluss einer Interaktion, wurde eine neue Datenabfrage mit den aktuellen Parametern durchgeführt. Dieses Verfahren führt dazu, dass beispielsweise Label beim Zoom in die Darstellung vergrößert dargestellt werden oder Randbereiche der Daten sichtbar werden können. Es ermöglicht im Gegenzug eine verzögerungsfreie Interaktion des Nutzers mit der Visualisierung. Lediglich nach dem Abschluss einer Interaktion ist eine kleine Wartezeit (ca. eine Sekunde) notwendig, um die Darstellung zu aktualisieren.

7.3. Messergebnisse

Zur Analyse der entwickelten Datenstrukturen und Datenzugriffe, wurde verschiedene Messungen an den implementierten Visualisierungen vorgenommen. Die Messungen wurden jeweils mit der PC-Anwendung (siehe Abbildung 7.1) und mit der Android App (siehe Abbildung 7.2) durchgeführt. Die PC-Anwendung wurde auf einem Desktop-PC mit einem Intel Core i5 Prozessor der ersten Generation mit einem Basistakt von 2,67 GHz pro Kern, 8 Gigabyte Arbeitsspeicher und einer Magnetischen Festplatte durchgeführt. Das Testgerät für die Android App war ein Asus Eee Pad Transformer TF101 Tablet mit einem Nvidia Tegra2 Prozessor mit 1 GHz Basistakt und 1024 Megabyte Arbeitsspeicher.

Die Messung wurde anhand von zwei verschiedenen Bildausschnitten durchgeführt. Der erste Ausschnitt umfasste das gesamte Bundesgebiet der Bundesrepublik Deutschland. Es waren sowohl Staats und Bundeslandgrenzen sichtbar, die Grenzsegmente wurden in der Generalisierungsstufe 10 (= sehr grob aufgelöst) dargestellt. Der zweite Ausschnitt umfasste einen rechteckigen Ausschnitt aus Baden-Württemberg um Stuttgart, ungefähr von Hockenheim bis Heidenheim an der Brenz. Es wurden die Staats- und Bundeslandgrenzen sowie die Grenzen der Regierungsbezirke und Kreise dargestellt in der Generalisierungsstufe 3 (= fein aufgelöst) dargestellt. Die Größe der Visualisierung der PC-Anwendung wurde auf 1280x800 Pixel festgelegt, der nativen Auflösung des Android Tablets.

	Gesamtes Bundesgebiet			Baden-Württemberg		
	max (ms)	min (ms)	Ø(ms)	max (ms)	min (ms)	Ø(ms)
GN Grid Abfrage	33	2	2,87	1	<1	0,32
GN Dateiabfrage	1	<1	0,33	2	<1	0,31
GN Gesamt	33	2	3,21	2	<1	0,64
VG Grid Abfrage	30	1	2,88	1	<1	0,79
VG Dateiabfrage	99	8	17,7	85	5	11,1
VG Gesamt	113	9	20,61	87	6	11,91

Tabelle 7.1.: Messung der Laufzeiten zur Abfrage einer darzustellenden Teilmenge der Daten mit der PC-Anwendung.

Es wurden jeweils 100 Datenabfragen zur Ermittlung der darzustellenden Datenmenge durchgeführt. Aus den benötigten Zeiten der Abfragen werden der maximale und minimale Wert, sowie der Durch-

	Gesamtes Bundesgebiet			Baden-Württemberg		
	max (ms)	min (ms)	Ø(ms)	max (ms)	min (ms)	Ø(ms)
GN Grid Abfrage	285	81	158,72	122	6	15,58
GN Dateiabfrage	133	5	9,81	216	36	56,62
GN Gesamt	347	88	168,59	233	43	72,26
VG Grid Abfrage	327	74	183	219	14	67,74
VG Dateiabfrage	496	174	299,33	264	63	102,26
VG Gesamt	645	400	482,5	287	78	170,1

Tabelle 7.2.: Messung der Laufzeiten zur Abfrage einer darzustellenden Teilmenge der Daten mit der Android App.

schnitt aller 100 Werte in den entsprechenden Tabellen dargestellt. Gemessen wurden jeweils die benötigte Zeit zur Abfrage der Suchstruktur (bezeichnet als Grid Abfrage) sowie die Zeit zum Auslesen der Daten aus den entsprechenden Dateien. Zudem wurde die Gesamtlaufzeit beider Operationen gemessen (Zeile GN Gesamt).

Bei den Messungen sind insbesondere der Durchschnitt der verschiedenen Werte interessant. Die große Varianz der Werte der Abfrage der Suchstruktur (zu sehen an den großen Unterschieden bei maximalem und minimalem Wert) ist vermutlich auf die interne Speicherverwaltung der Geräte zurückzuführen. Die große Varianz in der Dateiabfrage lässt sich durch Buffering der entsprechenden Dateien erklären. Wie zu erwarten war, sind die Zeiten bei der Android App wesentlich schlechter, als bei der PC-Anwendung. Interessant ist hier insbesondere, dass bei den unterschiedlichen Datensätzen unterschiedliche Abfragen die Laufzeit dominieren. Während bei den Ortsnamen die Abfrage der Suchstruktur dominiert, dominiert bei den Grenzsegmenten das Auslesen der Daten aus den Dateien die Laufzeit.

8. Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Diplomarbeit wurden zwei Datenstrukturen für die Verwaltung von geographische Basisdaten entwickelt. Die Datenstrukturen dienen der Bereitstellung der geographischen Basisdaten für eine Visualisierung. Die Datenstrukturen ermöglichen die Ableitung von Teilmengen der Daten, die in einer gegebenen Visualisierung eine gute Wahrnehmbarkeit der einzelnen Daten garantieren (indem beispielsweise eine Teilmenge der Ortslagen bestimmt wird, die ohne Überschneidung in der Visualisierung dargestellt werden können). Der Fokus der Entwicklung lag auf der Optimierung der Geschwindigkeit der benötigten Operationen und einer möglichst platzsparende Speicherung der Daten. Dies soll Visualisierungen ermöglichen, in denen der Nutzer die Daten verzögerungsfrei mittels Zoom, Verschieben und Rotieren erkunden kann und eine Verwaltung der gesamten Daten offline, beispielsweise auf einem Android-Gerät, ermöglichen.

Eine Datenstruktur dient der Verwaltung von Grenzsegmenten, welche die Gebiete verschiedener Verwaltungsebenen der Bundesrepublik Deutschland begrenzen. Sie ermöglicht eine Abfrage der Basisdaten der verschiedenen Verwaltungsebenen, von Staatsgrenzen und Bundeslandgrenzen bis hin zu Gemeindegrenzen. Die Grenzsegmente können in verschiedenen Genauigkeitsstufen abgefragt werden. Bei einer höheren Genauigkeitsstufe werden die Grenzsegmente durch mehr innere Punkte definiert, bei einer niedrigeren Genauigkeitsstufe definieren weniger Punkte den Verlauf des Segments.

Eine zweite Datenstruktur dient der Verwaltung von Ortslagen, wie Städten und Dörfern, im Gebiet der Bundesrepublik Deutschland. Die entwickelte Datenstruktur ermöglicht es für die Visualisierung eine konfliktfrei darstellbare Teilmenge der Ortslagen abzuleiten. Diese Menge der Ortslagen ist auch bei einer Rotation der Visualisierung weiterhin konfliktfrei darstellbar. Durch die Datenstruktur wird außerdem gewährleistet, dass Ortslagen, die in einer Reduktionsstufe nicht enthalten sind, auch bei der weiteren Reduktion der Daten nicht enthalten sind. Es wurde zudem ein Dateiformat entwickelt, um die Datenstrukturen möglichst platzsparend zu speichern.

Für einen schnellen Zugriff auf die Daten wurde eine Suchstruktur für die Daten entwickelt, die eine schnelle Abfrage von Daten in einem gegebenen Datenfenster ermöglichen.

Im Rahmen der Diplomarbeit wurden zwei Visualisierungen entwickelt, die ein Erkunden der Datensätze ermöglichen. Die Visualisierungen werden basierend auf OpenGL auf einem PC, bzw. basierend auf OpenGL ES auf Androidgeräten, hardwarebeschleunigt dargestellt. Sie ermöglichen eine interaktive Erkundung der Datensätze mittels Zoom, Verschieben und Rotation der Daten - auf einem PC beinahe in Echtzeit. Zentraler Aspekte der Implementierung sowie verschiedene Messungen von Laufzeiten wurden beschrieben.

Als Datenbasis dienten zwei Datensätze des deutschen Bundesamtes für Kartographie und Geodäsie (http://www.bkg.bund.de). Der Datensatz GN250 stellt verschiedenste Ortslagen im Gebiet der

Bundesrepublik Deutschland zur Verfügung, die darin enthaltenen Ortslagen wurden im Rahmen der vorliegenden Diplomarbeit visualisiert. Der Datensatz VG250 enthält Daten zu den verschiedenen Verwaltungsgebieten in der Bundesrepublik Deutschland. Aus diesem Datensatz wurden die Informationen zum Verlauf der Grenzen der Verwaltungsgebiete verwendet. Beide Datensätze sind als als Open Source frei verfügbar.

Ausblick

Das Verfahren für die Ableitung einer konfliktfrei darstellbaren Menge von Ortslagen funktioniert sehr gut, wenn die Darstellung der einzelnen Label der Ortslagen sehr ähnlich ist. Um die Wichtigkeit der Ortslagen zu visualisieren wäre eine starke Variation der Schriftgröße der Label denkbar. Es wäre wünschenswert, wenn die Ableitung einer konfliktfrei darstellbaren Menge der Ortslagen in einem solchen Setting möglich wäre. Hierfür müsste die Datenstruktur der Ortslagen weiterentwickelt werden, sodass die verschiedenen Größen der Label bei der Ableitung einer konfliktfrei darstellbaren Teilmenge der Datenpunkte berücksichtigt werden kann.

Im Rahmen der vorliegenden Arbeit wurden nur die Label von Ortslagen verarbeitet und visualisiert. Es wäre interessant die Datenstruktur so zu erweitern, dass zusätzlich Label für die dargestellten Verwaltungsgebiete platziert werden könnten. Da diese nicht einzelne Punkte beschreiben, könnten die entsprechenden Labels in den Freiräumen, die aus der Reservierung der kreisförmigen Fläche um die Datenpunkte resultieren, dargestellt werden. Sie könnten dann während der Rotation immer wieder neu in den Freiräumen platziert werden.

Bei der Visualisierung des Verwaltungsgebietsdatensatzes wäre eine zusätzliche Färbung der Flächen für die Wahrnehmbarkeit der Verwaltungsgebiete hilfreich. Die Datenstruktur der Verwaltungsgebiete müsste angepasst werden, um diese zusätzlichen Informationen zu enthalten. Dabei müsste insbesondere darauf geachtet werden, dass die Verläufe der Grenzsegmente und der Flächen der Verwaltungsgebiete auch in den verschiedenen Genauigkeitsstufen übereinstimmen.

Das Bundesamtes für Kartographie und Geodäsie stellt, neben den hier genutzten Datensätze, auch Datensätze zur Verfügung, die die verschiedene Nutzung von Flächen, wie beispielsweise Ortslage, forstwirtschaftliche Fläche, Vegetationsfläche, enthalten. Eine zusätzliche Darstellung dieser Daten, beispielsweise durch Färbung der entsprechenden Flächen, wäre eine zusätzliche Bereicherung für die entwickelte Visualisierung.

Eine Erweiterung der Datenstruktur und der Visualisierung auf andere Länder wäre interessant, um die Skalierbarkeit der eingesetzten Verfahren zu testen. Dabei entsteht das Problem, dass Daten aus unterschiedlichen UTM-Zonen dargestellt werden müssten.

Die Umstellung der Datenstrukturen, sodass die Daten des OpenStreetMap-Projekts genutzt werden könnten, würde die Integration von Kartenmaterial der ganzen Welt ermöglichen. Da die OSM-Datensätze nicht in UTM-Koordinaten verfügbar sind, ist die Umstellung der Verfahren zur Erstellung der Datenstrukturen notwendig. Insbesondere ist eventuell eine neue Methode zur Berechnung der Distanzen im Datensatz notwendig. Zudem ist eine Umstellung der Transformation der neuen Koordinaten in Bildkoordinaten notwendig.

A. Anhang

A.1. ATKIS-Objekte im Datensatz GN250

A.1.1. Bauwerke

AX_Bahnstrecke: Eisenbahn, Güterzugbahn, Magnetschwebebahn, Museumsbahn, S-Bahn, Standseilbahn, Zahnradbahn

AX_Bahnverkehrsanlage: Bahnhof, Haltepunkt

AX_BauwerkImGewaesserbereich: Siel, Sperrwerk, Staudamm, Staumauer, Wehr

AX_BauwerkImVerkehrsbereich: Brücke, Tunnel/ Unterführung

AX_BauwerkOderAnlageFuerSportFreizeitUndErholung: Sprungschanze (Anlauf), Stadion

AX_Flugverkehr: Internationaler Flughafen, Regionalflughafen

AX_Flugverkehrsanlage: Hubschrauberlandeplatz, Sonderlandeplatz, Verkehrslandeplatz

AX_Gebaeude: Burg/ Festung, Hütte (mit Übernachtungsmöglichkeit), Kirche, Krankenhaus, Museum, Parlament, Schloss

AX_Grenzuebergang:

AX_IndustrieUndGewerbeflaeche: Deponie (oberirdisch), Kraftwerk

AX_Kanal: Binnenwasserstraße

AX_Schleuse: Kammerschleuse, Schiffshebewerk

AX_SchifffahrtslinieFaehrverkehr: Autofährverkehr, Personenfährverkehr

AX_SeilbahnSchwebebahn: Kabinenbahn/ Umlaufseilbahn, Luftseilbahn/ Großkabinenbahn, Materialseilbahn, Sessellift, Ski-/ Schlepplift

AX_SonstigesBauwerkOderSonstigeEinrichtung: Gedenkstätte/ Denkmal/ Denkstein/ Standbild

AX_SportFreizeitUndErholungsflaeche:

AX_Strasse: Bundesautobahn, Bundesstraße, Kreisstraße, Landesstraße/ Staatsstraße, sonstiges

AX_Strassenverkehrsanlage: Autobahnknoten, Raststätte

AX_TagebauGrubeSteinbruch: Erden/ Lockergestein, Industrieminerale/ Salze, Steine/ Gestein/ Festgestein, Torf, Treib- und Brennstoffe

AX_Turm:

AX_Wasserlauf: Seewasserstraße

AX_WegPfadSteig:

A.1.2. Natur

AX_DammWallDeich:

AX_Fliessgewaesser:

AX_Gewaessermerkmal:

AX_Heide:

AX_Hoehleneingang:

AX_Insel:

AX_Landschaft: (Tief-) Ebene/ Flachland, Becken/ Senke, Berg/ Berge, Gebirge/ Bergland/ Hügelland, Inselgruppe, Küstenlandschaft, Moorlandschaft, Plateau/ Hochfläche, Seenlandschaft, Siedlungs-/ Wirtschaftslandschaft, Tal/ Niederung, Wald- Heidelandschaft

AX_Meer:

AX_Moor:

AX_Sumpf:

AX_Wald:

A.1.3. Gebiete

AX_NaturUmweltOderBodenschutzrecht: Naturpark

AX_Ortslage:

AX_SchutzgebietNachNaturUmweltOderBodenschutzrecht: Biosphärenreservat, Nationalpark

AX_Siedlungsflaeche:

AX_SonstigesRecht: Truppenübungsplatz/ Standortübungsplatz

Key: AX_StehendesGewaesser: Landesgewässer mit Verkehrsordnung

Key: AX_UnlandVegetationsloseFlaeche: Eis/ Firn, Fels, Sand

Key: AX_Verwaltungsgemeinschaft_ATKIS:

A.1.4. Verwaltungsgebiete

AX_Bundesland:

AX_Gemeinde:

AX_KreisRegion:

AX_Nationalstaat:

AX_Regierungsbezirk:

Literaturverzeichnis

[CMS95] J. Christensen, J. Marks, S. Shieber. An Empirical Study of Algorithms for Point-feature Label Placement. *ACM Trans. Graph.*, 14(3):203–232, 1995. (Zitiert auf den Seiten 7, 10 und 30)

[DP73] D. Douglas, T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, S. 112–122, 1973. (Zitiert auf den Seiten 10 und 41)

[Ebi02] K. Ebisch. A correction to the Douglas-Peucker line generalization algorithm. *Computers & Geosciences*, 28(8):995 – 997, 2002. (Zitiert auf den Seiten 10 und 41)

[Fre05] H. Freeman. Automated Cartographic Text Placement. *Pattern Recogn. Lett.*, 26(3):287–297, 2005. (Zitiert auf Seite 30)

[GN-13] Geographische Namen 1:250 000: GN250, 2013. URL http://sg.geodatenzentrum.de/web_download/gn/gn250/gn250.pdf. (Zitiert auf den Seiten 7, 13, 14, 15, 16 und 32)

[Har01] R. Harbeck. 15 Jarhe ATKIS®, und die Entwicklung geht weiter. *Vermessung Brandenburg*, (1):3–14, 2001. URL http://www.geobasis-bb.de/GeoPortal1/produkte/verm_bb/pdf/101_s3-14.pdf. (Zitiert auf Seite 14)

[Imh62] E. Imhof. Die Anordnung der Namen in der Karte. *Internat. Yearbook of Cartography*, S. 93–129, 1962. (Zitiert auf den Seiten 10 und 30)

[Kru14] F. Krumpe. Hierarchisierung und Darstellung von Geodaten. Studienarbeit: Universität Stuttgart, Institut für Formale Methoden der Informatik, Theoretische Informatik, 2014. URL http://www2.informatik.uni-stuttgart.de/cgi-bin/NCSTRL/NCSTRL_view.pl?id=STUD-2432&engl=. (Zitiert auf Seite 10)

[Mot07] K. D. Mote. Fast Point-Feature Label Placement for Dynamic Visualizations, 2007. URL http://www.dissertations.wsu.edu/Thesis/Fall2007/k_mote_111307.pdf. (Zitiert auf den Seiten 30 und 31)

[S-D98] ESRI Shapefile Technical Description, 1998. URL http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf. (Zitiert auf den Seiten 7, 16, 19, 20 und 21)

[UTM09] UTM-Abbildung und UTM-Koordinaten, 2009. URL http://vermessung.bayern.de/file/pdf/1910/UTM%20Abbildung%20und%20Koordinaten.pdf. (Zitiert auf den Seiten 21 und 22)

[VG-13] Verwaltungsgebiete 1:250 000: VG250 und VG250-EW, 2013. URL http://sg.geodatenzentrum.de/web_download/vg/vg250-ew_3112/vg250-ew_3112.pdf. (Zitiert auf den Seiten 7, 13, 16, 17 und 18)

[Wei14a] E. W. Weisstein. Point-Line Distance–2-Dimensional, 2014. URL http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html. (Zitiert auf Seite 43)

[Wei14b] E. W. Weisstein. Rotation Matrix, 2014. URL http://mathworld.wolfram.com/RotationMatrix.htmll. (Zitiert auf Seite 48)

[Wol99] A. Wolff. Automated Label Placement in Theory and Practice, 1999. URL http://i11www.iti.uni-karlsruhe.de/~awolff/pub/w-alptp-99.pdf. (Zitiert auf Seite 30)

[WP-13] UTM-Koordinatensystem, 2013. URL https://de.wikipedia.org/wiki/UTM-Koordinatensystem. (Zitiert auf den Seiten 21 und 22)

Alle URLs wurden zuletzt am 10.08.2014 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift