

UNIVERSITÄT OSNABRÜCK

MASTERARBEIT

Möglichkeiten der Visualisierung von
Geobasisdaten auf mobilen Endgeräten
mit dem Geoviewer - Prototypen des
LGLN. Evaluierung und
Weiterentwicklung

vorgelegt von: Patrick Huesmann

1. Prüfer/in:

Prof. Dr. rer. nat. NORBERT DE LANGE

2. Prüfer/in:

Prof. Dr. - Ing. ELKE PULVERMÜLLER

Zusammenfassung

Die vorliegende Masterarbeit zeigt die Möglichkeiten der Visualisierung von Geobasisdaten am Beispiel des Geoviewer-Prototypen des LGLN auf. Die Landesbehörde versucht in Zukunft die digitale topographische Karte mithilfe von Basisvisualisierung zu ersetzen und verwendet dazu im Prototypen Vektor-Tiles. Das Datenformat ist eine Alternative zum etablierten Raster-Tile-Format und soll in der Arbeit auf Vor- und Nachteile getestet werden. Dazu wird der Prototyp mithilfe von Google Lighthouse auf der technischen Seite evaluiert und danach mit einem Geoviewer auf Rasterbasis verglichen. So soll die Frage beantwortet werden, wie die Nutzung des Vektor-Tiles-Formats im Vergleich zu anderen Formaten zu bewerten ist.

Ein weiterer Aspekt der Arbeit ist die Weiterentwicklung des Prototypen. Sie ist maßgeblich von den Ergebnissen der Evaluation beeinflusst und beschäftigt sich mit der Implementierung von Styles für Farbenfehlsichtige. Die Zielgruppe kann normale Kartendarstellungen aufgrund ihrer Erkrankung nicht vollständig wahrnehmen. Hier setzt die Masterarbeit an und versucht mithilfe der Visualisierung von Geobasisdaten barrierefreie Inhalte zu schaffen. Diese Entwicklung zielt dabei auf Dichromaten ab.

Abstract

The present master thesis shows the possibilities of visualization of geobase data by the example of the prototype geoviewer of the LGLN. In the future, the state authority is trying to replace the digital topographic map with basic visualization using vector tiles in the prototype. The data format is an alternative to the established raster-tile format and will be tested in the work on pros and cons. The prototype will be evaluated on the technical side using Google Lighthouse and then compared to a grid-based geoviewer. Thus, the question should be answered, how to evaluate the use of the vector tiles format compared to other formats. Another aspect of the work is the further development of the prototype. It is significantly influenced by the results of the evaluation and deals with the implementation of styles for color blind people.. The target group can not fully perceive normal map presentations because of their disease. This is where the master's thesis comes in and tries to create barrier-free content using the visualization of geospatial reference data. This development is aimed at dichromates.

Danksagung

Vor Ihnen liegt die Masterarbeit

Möglichkeiten der Visualisierung von Geobasisdaten auf mobilen Endgeräten mit dem Geoviewer - Prototypen des LGLN. Evaluierung und Weiterentwicklung.

Sie ist in Zusammenarbeit mit dem Landesamt für Geoinformation und Landesvermessung Niedersachsen erstellt worden. Ich möchte mich hiermit herzlich bei meinen Betreuern in der Landesbehörde Dr. Marcel Ziems und Sebastian Ratjens bedanken. Sie haben mich jederzeit mit Ratschlägen und konstruktiver Kritik unterstützt und so zum Gelingen dieser Masterarbeit beigetragen.

Zudem möchte ich Herrn Prof. Dr. rer. nat. de Lange für seine Anregungen und Gespräche danken, die mir während der Bearbeitung der Masterarbeit geholfen haben, den roten Faden der Arbeit zu verfolgen.

Abschließend möchte ich meinen Eltern bedanken, die mich während des Verfassens mit Tipps und aufmunternden Worten unterstützt haben.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	1
1.2	Aufbau und Inhalt der Arbeit	3
2	Begriffsdefinitionen	4
2.1	Visualisierung	4
2.2	Geobasisdaten	6
3	Literatur und verwandte Arbeiten	7
3.1	Arbeiten zur Visualisierung von Geobasisdaten	7
3.2	Evaluierung von mobilen Anwendungen	10
3.2.1	Praktische Leistungsmessung	12
4	Theoretische Grundlagen	13
4.1	Visualisierungsprozess	13
4.2	Tile-basierte Kartendarstellung	14
4.3	Raster & Vektor-Tiles	17
4.4	Mapbox Vector Tiles	22
4.4.1	Protocol Buffers	22
4.4.2	Geometrikodierung	23
4.4.3	Styling	24
4.5	Docker	25
4.6	T-rex	26
4.7	PostgreSQL & PostGIS	26
4.8	Google Lighthouse	26
4.9	Farbenfehlsichtigkeit	29
4.10	Web Content Accessibility Guidelines	30
4.11	Color Blindness Simulator	31
5	Prototyp	31
5.1	Webserver	35
6	Evaluierung	38
7	Weiterentwicklung des Prototypen	51
7.1	Styleentwicklung auf Basis von WCAG	51
7.2	Evaluation der entwickelten Styles	56
8	Diskussion & Ausblick	58

Abbildungsverzeichnis

1	Map use Cube nach Mac Eachren & Kraak. Quelle: Schiewe, J.[9] o.S.	5
2	Visualisierung von Geodaten mit Hilfe eines Portrayalwebdienstes nach Wiebrock. Quelle: Eigene Abbildung, verändert nach Wiebrock[14] S.31.	7
3	Visualisierungspipeline nach Schumann Quelle: Eigene Abbildung, verändert nach Schuhmann[7] S.15.	13
4	Tile Map von Melbourne, Australien. Quelle: Wikipedia [28] o.S.	15
5	Tile pyramid nach Garcia et al.. Quelle: Garcia et al. [29] S.26.	16
6	Google Maps Tilling Scheme nach Stefanakis. Quelle: Stefanakis [30] S.9.	17
7	Räumliche Information im Rastermodell nach de Lange. Quelle: verändert nach de Lange [13] S.26.	17
8	Vektormodell und seine geometrischen Bestandteile (Features) nach Bartelme. Quelle: verändert nach Bartelme [32] S.65.	18
9	Kachelerzeugung aus Rasterdaten. Quelle: Eigene Abbildung.	19
10	Vergleich von Raster- und Vektortiles. Quelle: verändert nach ESRI[33].	20
11	Leeres Vektor- Tile. Quelle: verändert nach Mapbox[41].	23
12	Gefülltes Vektor- Tile. Quelle: verändert nach Mapbox[41].	24
13	Absorptionsspektren der Sehzelle. Quelle: verändert nach Welsch & Liebmann[49] S.241	29
14	Aufbau eines Vektor Tile Stack für Custom-Data. Quelle: verändert nach Kalberer[45] o.S.	32
15	Schritt 1: Aufbau der Datenbank innerhalb von Docker. Quelle: Eigene Abbildung.	32
16	Schritt 2: Anbindung des Vektor- Tile- Server. Quelle: Eigene Abbildung.	33
17	Schritt 3: Funktion und Lage des Webservers. Quelle: Eigene Abbildung.	34
18	Frontend des Webservers. Quelle: Eigene Abbildung.	35
19	Renderingprozess Quelle: Eigene Abbildung.	36
20	Kartengeneralisierung Quelle: Eigene Abbildung.	37
21	Farbpalette für Farbfehlsichtige Quelle: verändert nach Krzywinski[57].	53
22	Farbpalette für Protanopen Quelle: Eigene Abbildung.	54
23	Farbpalette für Deutanopen Quelle: Eigene Abbildung.	54
24	Farbpalette für Tritanopen Quelle: Eigene Abbildung.	55
25	Farbpalette aus Sicht eines Protanopen Quelle: Eigene Abbildung.	56
26	Farbpalette aus Sicht eines Deutanopen Quelle: Eigene Abbildung.	57
27	Farbpalette aus Sicht eines Tritanopen Quelle: Eigene Abbildung.	57

Abkürzungsverzeichnis

API Programmierschnittstelle / Application Programming Interface

AJAX Asynchronus JavaScript and XML

LGLN Landesamt für Geoinformation und Landesvermessung Niedersachsen

VKV Vermessungs- und Katasterverwaltung

SQL Structured Query Language

XML Extensible Markup Language

XSQL Combining XML and SQL

AFIS Amtliches Festpunktinformationssystem

ALKIS Amtliches Liegenschaftskatasterinformationssystem

ATKIS Amtliches Topographisch-Kartographisches Informatonsystem

GIS Geoinformationssystem

IÖR Leibniz-Institut für ökologische Raumentwicklung

XHTML Extensible Hypertext Markup Language

VGI Volunteered geographic information

ODK Open Data Kit

OpenGL Open Graphics Library

WMS Web Map Service

WWW World Wide Web

JPEG Joint Photographic Experts Group

PNG Portable Network Graphic

MVT Mapbox Vector Tiles

OSM OpenStreetMap

ESRI Environmental Systems Research Institute

JSON JavaScript Object Notation

VM Virtuelle Maschine

SEO Search Engine Optimization

WCAG Web Content Accessibility Guidelines

Coblis Color Blindness Simulator

FOSSGIS Freie und Open Source Software für Geoinformationssysteme

UTM Universal Transverse Mercator

Sass Syntactically Awesome Stylesheets

URL Uniform Resource Locator

AdV Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik
Deutschland

IEC International Electrotechnical Commission

ISO International Organization for Standardization

1 Einleitung

Karten dienen schon seit Jahrtausenden als herausragende Möglichkeit, räumliches Wissen, oder allgemeiner formuliert Geoinformationen zu visualisieren(vgl. Weber et al.[1] S.39). Im Rahmen des 2005 von O'Reilly postulierten Web 2.0(vgl. O'Reilly[2] o.S.) entwickelten sich viele neue Technologien, die weitreichende Folgen für die Visualisierung von Karten mit sich brachten. Zum einen sind hier Mapping Programmierschnittstellen(API), wie beispielsweise Google Maps, OpenLayers oder OpenStreetMap zu nennen, die im Zuge dieser Entwicklung „die kartographische Bereitstellung von Inhalten auf Websites prägen“(Mittlböck et al.[3] S.333). Sie werden unterstützt durch weitere Neuerungen auf technologischer Ebene, wie das Cloud Computing oder neuen Webtechnologien wie Asynchronus JavaScript and XML(AJAX). Parallel zu diesen Veränderungen entwickelten sich neue Möglichkeiten zum Austausch und zur Bereitstellung raumbezogener Daten(vgl. Blankenbach et al.[4] S.3f.).

Eine dieser Neuentwicklungen, die auf den Technologien des Web 2.0 fußt, ist der Geoviewer. Er macht es möglich räumliche Zusammenhänge für Laien aufzubereiten und als mobile Anwendung zu präsentieren.

Hier setzt diese Masterarbeit an, in der die Möglichkeiten der Visualisierung mithilfe eines Geoviewer-Prototyps aufgezeigt werden soll. Der Prototyp ist vom Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN) entwickelt worden und zeigt die ersten Schritte der Landesbehörde in Richtung Basisvisualisierung von Geobasisdaten, insbesondere von Vektor-Tiles. Die Entwicklung wird durch diese Arbeit unterstützt, indem der Prototyp evaluiert und in einem darauffolgenden Schritt weiterentwickelt wird. Der Fokus der Arbeit ist dabei auf die Webentwicklung gerichtet. Das LGLN plant eine Ausweitung des Prototypen auf Smartphones, dieser Sachverhalt wird aber nicht in dieser Arbeit thematisiert.

1.1 Problemstellung

Das LGLN nimmt als Landesbehörde hoheitliche Aufgaben der Vermessung- und Katasterverwaltung (VKV) in Niedersachsen wahr. Diese werden durch die Digitalisierung und den damit verbundenen Veränderungen im Bereich Gesellschaft und Technologien maßgeblich beeinflusst. Um Antworten auf die dadurch resultierenden Fragen an die Behörde zu finden, hat das LGLN ein Zukunftskonzept für die VKV mit Ausrichtung auf das Jahr 2025 entwickelt(vgl. LGLN[5] S.18).

Das Konzept besteht aus sechs allgemeinen und 23 fachlichen Thesen, die Ziele für die VKV bis zum Jahre 2025 benennen. Für die Umsetzung dieser Ziele haben sich Projektgruppen gebildet, die aus Mitarbeitern der unterschiedlichen Bereiche des LGLN zusammengesetzt sind(vgl. Rausch[6] S.7f.). Eine dieser Arbeitsgruppen ist die Basisvisualisierung, die sich mit der These:

Basisvisualisierungen ersetzen die Digitale Topographische Karte (DTK) und sind Grundlage für sämtliche (Standard-) Präsentationen.

auseinandersetzt.

Seit einem Jahr wird in diesem Bereich Wissen aufgebaut um Neuentwicklungen zur Visualisierung von Geodaten zu forcieren. Eine Richtung, in die dieser Prozess der Behörde geht, sind Vektor-Tiles als Basis der Visualisierung. Aus dieser Idee ist eine Webentwicklung entstanden, die auf dem Vektor-Tile-Format basiert und es ermöglicht Geobasisdaten anzuzeigen.

In der Arbeit wird dieser Prototyp unter Beachtung folgender Forschungsfragen untersucht:

1. Wie ist die Nutzung des Vektor-Tile-Formats im Vergleich zu anderen in der Wissenschaft existierenden Standards zu bewerten?
2. Welche Methoden werden im Rahmen der Weiterentwicklung des Prototypen hinzugefügt? Welche Aussagen können dabei über die Visualisierung von Geobasisdaten abgeleitet werden?

1.2 Aufbau und Inhalt der Arbeit

Um die Möglichkeiten der Visualisierung von Geobasisdaten zu untersuchen, ist diese Arbeit in acht unterschiedliche Kapitel aufgeteilt.

Kapitel eins beschäftigt sich mit der Problemstellung dieser Arbeit. Dabei werden zwei Forschungsfragen vorgestellt, auf die die Arbeit in ihrem Verlauf Antwort geben soll.

In Kapitel zwei werden Visualisierung und Geobasisdaten im Kontext der Masterarbeit definiert. Dabei handelt es sich um Begriffe von besonderer Bedeutung, die Erläuterung bedürfen.

Kapitel drei gibt einen Überblick über Literatur und verwandte Arbeiten. Diese lassen sich in zwei verschiedene Gruppen aufteilen, zum einen in Arbeiten zur Visualisierung von Geobasisdaten, zum anderen in Arbeiten zur Evaluierung von mobilen Anwendungen.

Im nächsten Kapitel geht es um die theoretischen Grundlagen der Arbeit. Hier werden theoretische Konzepte, sowie Bestandteile des Prototypen und Programme vorgestellt.

Kapitel fünf befasst sich mit dem Aufbau des Prototypen. Zudem werden die Funktionen und deren Implementation beschrieben.

Das sechste Kapitel umfasst die Evaluierung des Prototypen. Dabei wird eine technische Evaluation mithilfe von Google Lighthouse durchgeführt und der Prototyp danach mit einem Raster-Tile-Geoviewer verglichen.

Im siebten Kapitel wird die Weiterentwicklung des Prototypen beschrieben. Dabei werden drei Styles für Farbenfehlsichtige vorgestellt. Diese werden im Zuge der Neuentwicklung evaluiert und ihr Nutzen bewertet.

Im letzte Kapitel werden die in Kapitel eins gestellten Forschungsfragen beantwortet und weitere Aspekte der Arbeit diskutiert. Zudem wird ein Ausblick über mögliche Entwicklungen des Prototypen gegeben.

2 Begriffsdefinitionen

Aufgrund der Komplexität des Themas ist es notwendig, vor Beginn der eigentlichen Arbeit wichtige Grundbausteine der Ausarbeitung, wie Visualisierung und Geobasisdaten zu erläutern. Dazu werden einige Grundbegriffe in diesem Kapitel erklärt und im Kontext der Masterarbeit eingeordnet.

2.1 Visualisierung

Eines der Kernthemen dieser Ausarbeitung kann mit dem Begriff der Visualisierung umschrieben werden.

Visualisierung beschreibt dabei den Prozess abstrakte Daten in eine graphische Form zu bringen und sie so erfassbar zu machen. Der Begriff ist in vielen wissenschaftlichen Disziplinen verankert und dient als Oberbegriff.

Schumann und Müller definieren ihn im technisch - wissenschaftlichem Zusammenhang als „rechnergestützte, visuelle Präsentation von Daten, Informationen und Wissen in einer für den Menschen adäquaten und für die jeweilige Anwendung in diesem Kontext sinnvollen Form zu verstehen“ (Schumann und Müller[7] S.VI).

Im Forschungsgebiet der Geowissenschaften wird in wissenschaftlichen Artikeln oft über eine Spezialform der Visualisierung, die Geovisualisierung gesprochen.

Der Begriff wurde unter anderem von Mac Eachren und Kraak sowie Harley und Woodward geprägt und beschreibt einen interdisziplinären Ansatz aus Kartographie, Geoinformationssysteme, Bildanalyse und Informationsvisualisierung um Methoden und Theorien zu entwickeln, die eine Analyse oder Präsentation von Geodaten ermöglichen(vgl. MacEachren und Kraak[8] S.3). Sie spezifizieren somit die reine Visualisierung auf ein Umfeld mit Raumbezug.

Ziele der Geovisualisierung zeigt der sog. „map use cube“ nach Mac Eachren und Kraak (1997), der auf der folgenden Seite abgebildet ist. Grundannahme in diesem Modell ist, dass sich die Kartennutzung in einem dreidimensionalem Raum in Abhängigkeit von drei Faktoren abbilden lässt. Die Nutzer der Karte, der Grad der Interaktion und der Informationsgehalt beeinflussen die Kartennutzung, sodass sich verschiedene Ziele der Geovisualisierung entwickeln.

Exploration

Das erste Ziel einer Kartendarstellung kann die Exploration von raumbezogenen Daten sein. Hier werden unbekannte Datensätze von Experten untersucht, um neue Erkenntnisse zu generieren. Werkzeuge der Geovisualisierung sind hier beispielsweise Animationen oder Viewer, die weiterführende Informationen zu den Karten liefern.

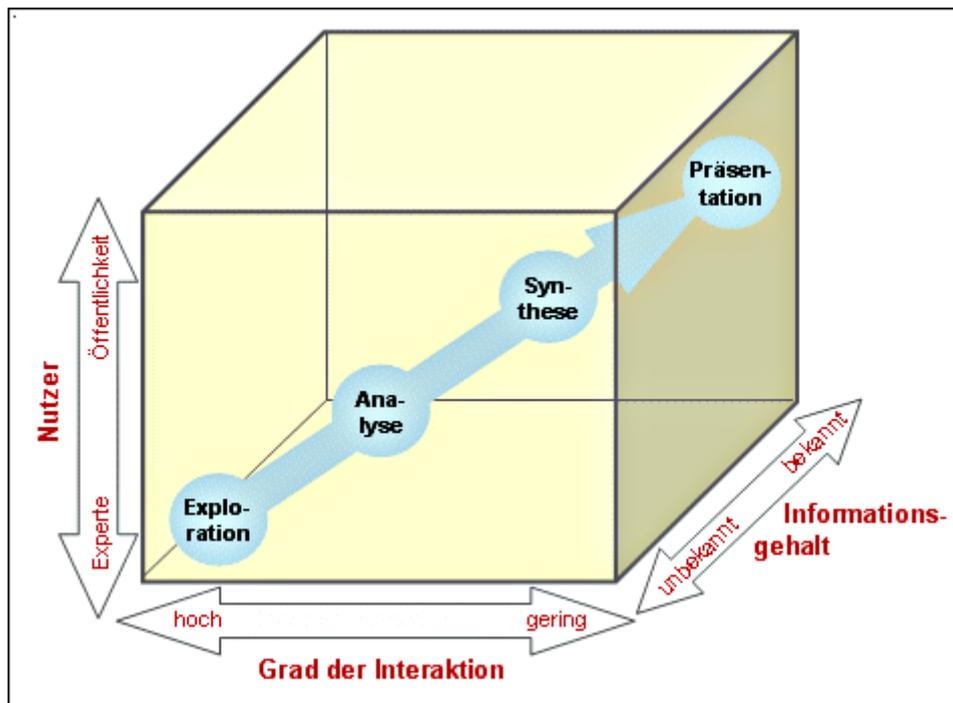


Abbildung 1: Map use Cube nach Mac Eachren & Kraak. Quelle: Schiewe, J.[9] o.S.

Analyse

Die Analyse ist ein weiterer Visualisierungszweck, bei dem nach Verhältnissen und Gesetzmäßigkeiten in bekannten Daten gesucht wird. Dazu wird die Datengrundlage zumeist durch räumlichen Operationen manipuliert, sodass die Beziehungen zwischen den Daten sichtbar werden. Ein Werkzeug zum Erreichen dieses Ziels sind beispielsweise Overlays, die eine räumliche Korrelation nahelegen.

Synthese

Die Synthese baut auf den Ergebnissen der Analyse auf, in diesem Schritt wird versucht, mithilfe einer Gruppe von Experten aus den vorhandenen Informationen weitere Ergebnisse zu ermitteln. Dazu werden Datensätze in ganzem Umfang und Komplexität zusammengestellt und Werkzeuge zur Bearbeitung wie Filter benutzt, um die Kernaussagen des Datensatzes zu abstrahieren.

Präsentation

Das vierte Ziel der Geovisualisierung nach Mac Eachren und Kraak ist die Präsentation der Ergebnisse. Zu diesem Zweck wird die generelle Aussage der Daten einer breiten Öffentlichkeit verständlich visualisiert. Die Art und Weise der Präsentation sollte dabei den Austausch von räumlicher Information betonen(vgl. Mac Eachren und Kraak[10] S.337-339).

Ein Werkzeug um die Informationen und Geodaten einem breiten Publikum zu zeigen, ist der Geoviewer - Prototyp des LGLN, der in dieser Arbeit evaluiert und weiterentwickelt werden soll. Somit bezieht sich die Visualisierung im Titel sowohl auf den Oberbegriff, als auch auf Geovisualisierung.

2.2 Geobasisdaten

Die Datengrundlage der Visualisierung bilden in dieser Arbeit Geobasisdaten des Landes Niedersachsen.

Geodaten

In der Wissenschaft werden Geodaten als eine Obermenge von Geobasisdaten verstanden und folgendermaßen definiert:

Geodaten bezeichnen „digitale oder analoge Informationen, die als wesentliches Merkmal einen Raumbezug haben. Sie können also einem Punkt, einer Linie oder einer Fläche auf der Erdoberfläche zugeordnet werden. Auf und mit Geodaten lassen sich Abfragen, Analysen und Auswertungen für unzählige raumbezogene Fragestellungen aus allen Bereichen von Verwaltung, Wirtschaft und Forschung durchführen“(LGLN[11] o.S.).

Geobasisdaten

Geobasisdaten hingegen sind grundlegende amtliche Geodaten, welche die Grundstücke, Gebäude sowie die Topographie anwendungsneutral beschreiben. Zu den Geobasisdaten zählen:

1. Parameter und Koordinaten eines geodätischen Referenzsystems,
2. die digitalen Landschaftsmodelle der Landesaufnahme,
3. die digitalen Geländemodelle und deren abgeleitete Produkte,
4. die Daten des Liegenschaftskatasters,
5. Luft- und Satellitenbilder und Orthophotos(Benish et al.[12] S.28f.)

Diese Daten werden über verschiedene Fachinformationssysteme repräsentiert.

Das amtliche Festpunktinformationssystem (AFIS) hält Geobasisdaten für den geodätischen Raumbezug, also die Festpunkte der Landesvermessung vor. Dazu gehören beispielsweise Schwerefestpunkte oder Nivellementpunkte.

Im Amtlichen Liegenschaftskatasterinformationssystem (ALKIS) werden Daten zur Beschreibung der Liegenschaften vorgehalten, diese resultieren aus den Vorgängern der automatisierten Liegenschaftskarte und dem automatisierten Liegenschaftsbuch.

Das dritte und letzte Fachinformationssystem ist das Amtliche Topographisch-Kartographische Informationssystem(ATKIS). Hier sind Geobasisdaten zur Beschreibung der Topographie der Bundesrepublik Deutschland, wie Landschafts- und Geländemodelle gespeichert(vgl. de Lange[13] S.218).

Der Prototyp basiert auf ATKIS-Daten und wird im Kapitel 5 genauer beschrieben. Bei den Daten handelt sich nach o. g. Definitionen sowohl um Geodaten, als auch um Geobasisdaten.

3 Literatur und verwandte Arbeiten

In diesem Abschnitt werden Inhalte wissenschaftlicher Arbeiten vorgestellt, die thematische Schnittmengen mit der Masterarbeit aufweisen. Dazu teilt sich dieses Kapitel in zwei Kategorien auf, die wissenschaftliche Erkenntnisse zu wichtigen Themen der Arbeit liefern.

Neben Arbeiten zur Visualisierung von Geobasisdaten wird auch Literatur betrachtet, die sich mit dem Thema der Evaluierung von mobilen Anwendungen beschäftigt.

3.1 Arbeiten zur Visualisierung von Geobasisdaten

Aufgrund der Nähe beider Begriffe¹ werden in folgendem Abschnitt Arbeiten zur Visualisierung von Geodaten / Geobasisdaten gleichermaßen vorgestellt. Die Grundannahme ist, dass sich die Visualisierung beider Datentypen stark ähnelt / identisch ist.

Zur kontextbasierten Visualisierung von Geodaten auf Basis von standardisierten Webdiensten (2011)

Wiebrock diskutiert in ihrer Dissertation die Möglichkeiten der kontextbasierten Visualisierung für Darstellungsdienste im Web. Die Arbeit verfolgt das Ziel einen Handlungsrahmen zu erstellen, der es erlaubt für einzelne Nutzergruppen spezifische Geodatenpräsentationen mithilfe von standardisierten Geowebdiensten zu erzeugen. Sie merkt an, dass der Fokus der Arbeit dabei auf der Visualisierung von Geodaten liegt(vgl. Wiebrock[14] S.10f.).

Dazu führt Wiebrock den Begriff Portrayal ein, der die Präsentation oder Visualisierung von

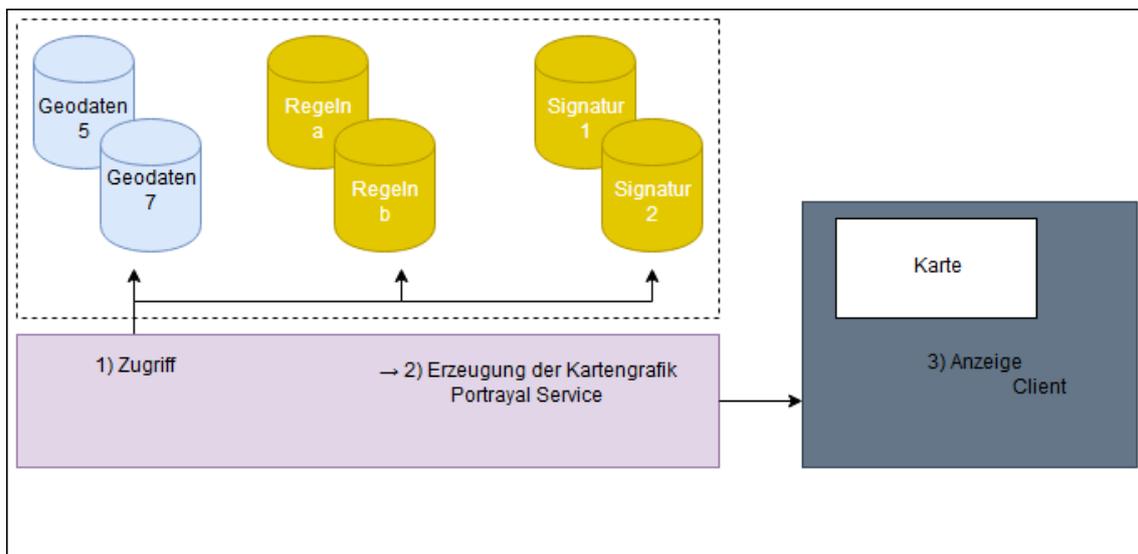


Abbildung 2: Visualisierung von Geodaten mit Hilfe eines Portrayalwebdienstes nach Wiebrock. Quelle: Eigene Abbildung, verändert nach Wiebrock[14] S.31.

Geoinformation beschreibt. Dieser ist nach ISO 19117 normiert und umfasst dabei den Prozess

¹ vgl. Kapitel 2.2

der Transformation von Geodaten in eine für den Menschen sichtbare Anzeige, den Portrayalprozess(vgl. Wiebrock[14] S.20).

In der Dissertation werden auch verschiedene Ansätze der Visualisierung diskutiert, die Verfasserin geht dabei auf Desktop- Geoinformationssysteme (GIS) gestützte, Web gestützte und dienstbasierte Verfahren ein. Alle genannten Methoden beruhen in der Arbeit dabei auf Signaturenkatalogen, die einheitliche Darstellungsregeln und Signaturen für Objektinstanzen beinhalten(vgl. Wiebrock[14] S.33). Der daraus entwickelte Visualisierungsprozess ist auf der vorherigen Seite abgebildet. Um die Daten für das Projekt zugänglich zu machen, werden sie in einer räumlichen Datenbank von Oracle persistent gespeichert.

Um das Ziel einer kontextbasierten Geodatensvisualisierung zu erreichen, werden Teile des Portrayalprozesses an den Kontext angepasst. Dazu verändert Wiebrock die Darstellungsregeln und Signaturen und führt zudem Kontextdateien und Filter ein, die zur Adaption des Systems an den Nutzer führen sollen. Die so entstandenen Adaptionenregeln werden im Darstellungsdienst mithilfe eines Apache Servers als XSQL-Dateien² gespeichert und bei Anfrage an den Server umgesetzt(vgl. Wiebrock[14] S.113f.)

Der aus dieser Vorarbeit entstandene Prototyp zeigt, dass die Nutzung von Kontextinformationen einen Mehrwert im Rahmen der Visualisierung darstellt. Die Autorin weist aber auch daraufhin, dass nur spezielle Anwendungsbereiche durch die im Prototyp implementierten Kontextelemente und Adaptionenregeln abgedeckt werden. Diese müssen bei Bedarf an den gewünschten Anwendungsbereich angepasst werden(vgl. Wiebrock[14] S.123).

Monitoring of settlement and open space development on the basis of topographic spatial data - concept, realization and first results(2010)

Meinel stellt in diesem 2010 erschienenen Artikel das Konzept sowie erste Eindrücke eines vom Leibniz Institut für ökologische Raumentwicklung(IÖR) entwickelten IÖR- Monitors vor. Dieser soll den Zustand und die Entwicklung der Flächennutzung in Deutschland anhand verschiedener Indizes beschreiben(vgl. Meinel[15] S.1f.). Die Indizes zeigen unter anderem die Bevölkerungs- und Gebäudedichte an und basieren auf topographischen Geobasisdaten aus Deutschland. Dabei werden ATKIS- Daten von einem Programm durch Filter und Klassifikationen ausgewertet und abgespeichert(vgl. Meinel[15] S.3).

Zur Visualisierung der Indizes sind zwei Geoviewer verfügbar, die unterschiedliche Anwendungsmöglichkeiten bieten.

Der erste, nutzerfreundliche Viewer zeigt die Indizes im Rahmen einer Übersicht und kann sie bis auf kommunaler Ebene darstellen. Basis dieser Darstellung ist das SVG-Format, in dem die Werte in einer standardisierten XML-Struktur abgebildet sind. Die Präsentation der Seite wird über XHTML³ umgesetzt und kann Nutzerpräferenzen speichern.

Die Geobasisdaten werden dabei in einer räumlichen Datenbank angelegt(vgl. Meinel[15] S.4)

² XSQL bezeichnet die Verbindung von XML und SQL und ermöglicht eine universelle Datenbeschreibung.

³ XHTML bezeichnet die Verbindung von XML mit HTML und besitzt ähnliche Vorteile wie XSQL.

und bei Bedarf abgerufen, analog zur Verwendung der Oracle-Datenbank in der Dissertation von Wiebrock.

Mit dem zweiten detaillierten Geoviewer sind die Indizes auch in sehr kleinem Maßstab verfügbar, sodass sie auf Hausblockebene dargestellt werden können. Die Visualisierung beruht dabei clientseitig auf OpenLayers und serverseitig auf ArcGIS Server. Zudem können WMS-Dienste zugeschaltet werden, die es ermöglichen eigene Daten in den Viewer einzubinden.

Im Ausblick verweist Meinel darauf, dass die Indikatoren für jeden Bundesstaat in einem zweijährigen Turnus berechnet werden.

Zudem soll der Monitor 2010 online verfügbar sein und weitere Indizes beinhalten (Meinel[15] S.5).

Neben Ansätzen zur 2D-Visualisierung von Geodaten finden sich in der jüngeren Vergangenheit auch Arbeiten, die den Raum oder die Zeit als dritte bzw. vierte Dimension miteinbeziehen. So entstehen multidimensionale intelligente Geoviewer, die in der Literatur oft auch als virtuelle Globen bezeichnet werden (vgl. Brovelli et al.[16] S.219). Als ein Beispiel für eine derartige Visualisierung wird hier die Arbeit von Brovelli et al. vorgestellt.

New Generation Platforms for Exploration of Crowdsourced Geo-Data (2018)

In diesem Kapitel des Buches Earth Observation Open Science and Innovation beschreiben Brovelli et al. die Entwicklung von Web-Mapping-Anwendungen im Zeitraum von 2005 bis 2018. Die Wissenschaftler identifizieren dabei unterschiedliche Faktoren, die zu ausschlaggebenden Veränderungen in diesem Bereich geführt haben.

Zum einen werden Technologien wie AJAX und Mapping-APIs genannt, die zu neuen Möglichkeiten im Bereich Web-Mapping geführt haben.

Zum anderen sprechen die Autoren den Begriff Volunteered geographic information (VGI) an. Freiwillig erhobene geographische Informationen haben für eine breite Geodatenbasis gesorgt und können beispielsweise in Projekten wie OpenStreetMap benutzt werden (vgl. Brovelli et al.[16] S.220f.).

Diese beiden im Rahmen des Web 2.0 entstandenen Neuerungen haben die Basis für die Entwicklung digitaler Globen gelegt. Als Beispiele für derartige Geoviewer stellen Brovelli et al. dazu zwei unterschiedliche digitale Globen vor⁴.

⁴ Aufgrund der inhaltlichen Nähe beider Beispiele wird im Folgenden nur der digitale Globus PoliCrowd vorgestellt.

PoliCrowd ist ein digitaler Globus, der von Brovelli et al. entwickelt wurde. Er basiert im Grundsatz auf zwei Komponenten, dem Open Data Kit (ODK) und der NASA World Wind Plattform.

Zur Datenhaltung dient die freie Software ODK. Sie besteht aus drei Komponenten und nutzt HTML, Datenbanken und Serversoftware um VGI zu sammeln, sie zu speichern und dem Projekt als Backend zur Verfügung zu stellen. Die Daten liegen dabei auf einem GlassFish - Server, der eine PostgreSQL - Datenbank nutzt.

Die Visualisierung dieser Daten wird durch die NASA World Wind SDK realisiert. Das dazugehörige Programm ist in Java geschrieben und benutzt die JOGL Bibliothek um Geodaten mithilfe der Open Graphics Library (OpenGL) anzuzeigen (vgl. Brovelli et al. [16] S.225 - 227). Zudem können OGC - konforme WMS - Server benutzt werden, um digitale Höhenmodell für die Darstellung des Geländes zu laden.

Um zeitliche Geoinformationen zu visualisieren benutzen Brovelli et al. VGI - Daten. Die Nutzer von ODK Connect, ein Teil der ODK Software, können zu bestimmten Orten Kommentare oder Multimediainhalte hinterlegen. Diese werden bei der Interaktion mit einem Point of Interest angezeigt und können durch den PoliCrowd - Klienten abgerufen werden.

Die Serverstruktur und Funktionalitäten dieses Projekts zeigen dabei Ähnlichkeiten mit den beiden zuvor vorgestellten Arbeiten. Darauf aufbauend erweitern die Autoren ihren Globus mit neuer Visualisierungstechnik auf Basis von OpenGL, um die dritte und vierte Dimension abzubilden.

Neben diesen Arbeiten existieren noch viele weitere Ausarbeitungen zum Thema Visualisierung von Geobasisdaten. In diesem Kontext sei auf weitere Forschungsarbeiten verwiesen, die sich beispielsweise mit der Visualisierung von Geodaten im Smartphonebereich (vgl. Menninghaus [17] o.S.) oder in der Raumplanung (vgl. Steffen et al. [18] o.S.) beschäftigen.

3.2 Evaluierung von mobilen Anwendungen

Grundsätzlich unterscheidet man in der Literatur unterschiedliche Formen der Evaluierung von mobilen Anwendungen.

Auf der einen Seite sprechen Forscher den Themenkomplex Usability, also die Nutzbarkeit einer Anwendung an. Dazu existieren unterschiedliche Standards, die in wichtigen Arbeiten zu diesem Thema genannt werden.

Auf der anderen Seite muss die Leistung einer mobilen Anwendung messbar sein. Dazu stellen Autoren der Forschungsarbeiten verschiedene Programme vor, die wichtige Indizes zur Identifizierung der Leistung ermitteln. Dieser Abschnitt ist also in die Themenkomplexe Usability und praktische Leistungsmessung unterteilt.

Usability

In der Literatur ist der Begriff Usability mehrfach besetzt und muss vor dem Hintergrund des Anwendungsgebiets betrachtet werden. So beschreiben Fernandez et al. Usability im Bereich der Mensch-Computer-Interaktion mithilfe der ISO-Norm 9241-2011, als das Ausmaß „inwieweit ein Produkt von bestimmten Nutzern verwendet werden kann, um bestimmte Ziele mit Effektivität, Effizienz und Zufriedenheit [...] zu erreichen“ (Fernandez et al.[19] S.2).

Neben dieser Betrachtungsweise kann man Usability auch im Kontext des Software-Engineering definieren, wie beispielsweise Abran et al. unter Berücksichtigung der ISO-Norm 9126-2001. In ihrer Forschungsarbeit beschreiben die Autoren Usability als „eine Reihe von Attributen, die sich auf den für die Nutzung erforderlichen Aufwand und die individuelle Bewertung einer solchen Nutzung [...] beziehen“ (Abran et al.[20] S.7).

Dieser 2001 entwickelte Standard ist mittlerweile durch die ISO 25000 ersetzt worden, zu dem die Normreihe 250xx mit weiteren Spezifikationen zum Thema Usability gehören. Lew et al. stellen in ihrem Artikel Quality, Quality in Use, Actual Usability and User Experience as Key Drivers for Web Application Evaluation ein Framework vor, welches die Qualität einer Webentwicklung modelliert. Die Entwicklung berücksichtigt dabei die ISO Norm 25010, also Qualitätskriterien zur Bewertung von Softwareprodukten und weitere Faktoren, wie die Qualität der Information und Lernbarkeit der Applikation (vgl. Lew et al.[21] S.231).

Die Bewertungen nach international anerkannten Standards ist dabei in der Literatur weit verbreitet. Auch Nayebi et al. (vgl. Nayebi et al.[22] S.1f.) und Fernandez et al. (vgl. Fernandez et al.[19] S.2) bewerten die Usability von Webapplikationen anhand der ISO Standards und deren Ausprägungen.

Dabei sind die Normen als eine Art Richtlinie zu verstehen. Die konkrete Durchsetzung einzelner Test variieren deutlich. Um einen Überblick über unterschiedliche Testmethoden zu geben existieren verschiedene Forschungsarbeiten, im Folgenden wird eine Mapping-Studie von Fernandez et al. aus 2011 vorgestellt. Die Forscher unterteilen dazu Forschungsarbeiten in unterschiedliche Kategorien (vgl. Fernandez et al.[19] S.7), unter anderem basierend auf der von Ivory und Hearst entwickelten Klassifikation *usability evaluation methods* (vgl. Ivory und Hearst[23] S.473). Dazu gehören:

1. User test
2. Inspection
3. Inquiry
4. Analytical Modeling
5. Simulation

Dabei zeigt sich, dass eine einzelne Methode nur bedingt für die Identifizierung von allen Artefakten einer Webseite geeignet ist. Die Autoren schlagen deshalb vor, mehrere der oben genannten Test zu verbinden, um ein besseres Ergebnis zu erzielen(vgl. Fernandez et al.[19] S.9). Die häufigste Methodik, die dabei in der Literatur zur Untersuchung der Usability verwendet wird, ist der Nutzertest.

Bei der Evaluation des Prototypen wird auf den Test der Usability verzichtet. Grund dafür ist, dass der Prototyp noch kein Endprodukt darstellt und dementsprechend noch viele Veränderungen im Bereich Nutzbarkeit der Applikation folgen. Dennoch gehört dieser Teil der Evaluierung in das Literaturkapitel, da es wesentliche Konzepte, die im Ausblick dieser Masterarbeit genannt werden beschreibt.

3.2.1 Praktische Leistungsmessung

Neben der Nutzbarkeit ist die Performance des Prototypen von wesentlichem Interesse. Die Visualisierung von Geobasisdaten ist technisch mit Mapbox umgesetzt und soll als Grundlage für Geoviewer des LGLN dienen. Vor diesem Hintergrund wird in diesem Unterkapitel ein Überblick über die unterschiedlichen Möglichkeiten in der Literatur gegeben, die Performance einer Webapplikation zu messen.

Relevante Forschungsarbeiten bedienen sich zu diesem Zweck oftmals an vorhandenen Testprogrammen, die die Leistung der Webseite bewerten. Zu den in der Literatur verwendeten Programmen gehören beispielsweise GTMetrix oder Pingdom (vgl. Kaur et al.[24] o.S. & Krol[25] o.S.).

Metriken, die die Programme dabei erfassen sind beispielsweise die Ladezeit der Seite, die Seitengröße oder die benötigte Bandbreite(vgl. Kaur et al.[24] S.11).

Neben etablierten Programmen existieren überdies Forschungsarbeiten, die neue Frameworks zur Leistungsbewertung vorstellen. Beispielhaft kann hier auf das 2012 entwickelte Framework Software Performance for Autonomic Computing von Litoiu und Barna verwiesen werden(vgl. Litoiu & Barna[26] o.S.).

Es bewertet den Nutzen einer Webentwicklung anhand der notwendigen Leistung, einer eigens entwickelten Utility-Funktion, die auf der erwarteten und tatsächlichen Antwortzeit der Applikation basiert sowie einer Entscheidungsfunktion, die die Leistungsfähigkeit der Architektur gegen die tatsächlich benötigt Leistung abwägt(vgl. Litoiu & Barna[26] S.872f., S.888f.).

So ist es mit dem Framework im Vergleich zu einzelnen Indizes möglich, die Softwarearchitektur an die tatsächlichen Bedürfnisse der Applikation anzupassen und so im Bezug auf die Leistungsfähigkeit zu verbessern.

In der Arbeit wird die Leistung des Prototypen anhand des Google Chrome Werkzeugs Lighthouse getestet. Die mit diesem Programm durchgeführten Tests prüfen dabei die Leistungsfähigkeit des Prototypen in verschiedenen Kategorien, die in Kapitel 4.8 vorgestellt werden.

Das Literaturkapitel hat gezeigt, dass Teilaspekte der Masterarbeit in einzelnen Forschungsarbeiten vorgestellt werden. Die Visualisierung von Geobasisdaten auf mobilen Endgeräten und deren Analyse hinsichtlich Leistung und Usability ist jedoch in der Forschung noch nicht ausreichend erforscht.

4 Theoretische Grundlagen

Im folgenden Kapitel werden relevante theoretische Inhalte, die das Thema der Masterarbeit berühren vorgestellt. Dazu gehören zum einen die Bestandteile des Prototypen, als auch die Programme, mit denen er evaluiert und weiterentwickelt wird.

4.1 Visualisierungsprozess

Nachdem Kapitel 2.1 einen Überblick über die Definition von Visualisierung und dessen Kontext gegeben hat, soll in diesem Abschnitt der Prozess der Visualisierung erklärt werden.

Dazu wird die Visualisierungspipeline vorgestellt, die den Weg von abstrakten Daten hin zum visualisierten Bild in verschiedenen Schritten beschreibt. Sie bezieht sich dabei auf die grundlegende Visualisierung, beschreibt jedoch wichtige Schritte, die auch in der Geovisualisierung und insbesondere in dieser Arbeit Anwendung finden.

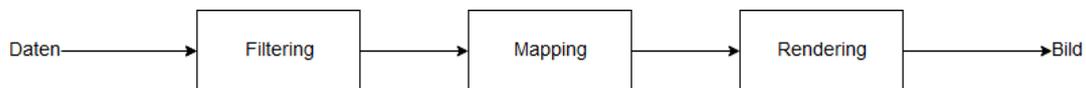


Abbildung 3: Visualisierungspipeline nach Schumann Quelle: Eigene Abbildung, verändert nach Schuhmann[7] S.15.

Filtering

Der erste Schritt der Visualisierungspipeline ist das Filtering, das die Aufbereitung und Filterung der vorhandenen Daten umfasst. Ausgangspunkt sind dabei die Rohdaten, die mithilfe von unterschiedlichen Operationen verändert werden. Je nach Fragestellung kann der Datensatz in diesem Schritt durch Algorithmen vervollständigt oder reduziert werden, um den Datensatz für die Visualisierung vorzubereiten.

Im darauffolgenden Schritt wird die so erstellte Datenmenge gefiltert, sodass es möglich ist Fehler aus dem Datensatz auszuschließen oder den Datensatz nach bestimmten Kriterien zu durchsuchen. Realisiert wird dies durch verschiedene Extrem- und Gradientenfilter, die auf die Daten angewendet werden.

Ergebnis des Filterings sind die aufbereiteten Daten, die in der Pipeline weiterverarbeitet werden können(vgl. Schuhmann[7] S.15f.).

Mapping

Im zweiten Schritt werden die aufbereiteten Daten in geometrische Daten umgewandelt. Die Datenwerte der Ausgangsdaten sind dabei die Grundlage, um eine Darstellung mithilfe von geometrischen Primitiven und der dazugehörigen Attribute zu ermöglichen.

Dieser Schritt ist der Kern der Visualisierungspipeline und beeinflusst maßgeblich das Ergebnis der Visualisierung. Hier werden die Darstellungsregeln, also die Entscheidung welche geometrischen Primitive und Attribute erzeugt werden, festgelegt (vgl. Schuhmann[7] S.16).

Dabei werden unter anderem Techniken wie die Klassifikation benutzt, um eine eindeutige Zuordnung der Datenwerte zur Eigenschaft zu gewährleisten (vgl. Schuhmann[7] S.255).

Rendering

Im letzten Schritt der Visualisierungspipeline werden die Geometriedaten als Bilddaten dargestellt. Der Prozess wird auch Rendering genannt und kann unterschiedliche Darstellungsarten umfassen, Beispiele sind:

1. Realitätsnahe Bilder - Realistische Visualisierung der Daten.
2. Abstrahierende Bilder - Details der Daten werden abstrahiert⁵.
3. Mentale Bilder - Innere und äußere Strukturen des Bildes werden zusammengefasst.
4. Animationen - Das Bild verändert sich kontinuierlich über die Zeit (vgl. Schuhmann[7] S.16f.).

Je nach Anwendungsfeld eignen sich unterschiedliche der aufgelisteten Darstellungsarten besser oder schlechter (vgl. Schuhmann[7] S.17) und werden dementsprechend für das Rendering gewählt.

Im Prototypen der Masterarbeit wird die Visualisierungspipeline aufgrund des Formats der Eingangsdaten nur teilweise durchlaufen. Der ATKIS-Datensatz bietet bereits Geometriedaten, die mithilfe von Mapbox im Vektor-Tile-Format dargestellt werden. Um einen Überblick über die möglichen Darstellungsformen von Geodaten zu geben, werden im Folgenden unterschiedliche Formate, die dem Stand der Technik entsprechen vorgestellt.

4.2 Tile-basierte Kartendarstellung

Die Darstellung von Karten im World Wide Web (www) hat sich im Verlauf seiner Entwicklung maßgeblich verändert. In den Anfängen der web-basierten Kartendarstellung gegen Mitte und Ende der 1990er Jahre wurden Karten im Web zumeist durch eine Bilddatei repräsentiert. Diese Art der Darstellung ist dabei mit Nachteilen verbunden. So besitzen Funktionen, die auf die Datengrundlage zugreifen, wie beispielsweise der Download oder das Verschieben der Karte,

⁵ Beispielsweise Generalisierung von Straßenzügen.

eine lange Wartezeit(vgl. Sample & Ioup[27] S.1).

Im Rahmen des Web 2.0 hat sich die Darstellung von Karten im Internet grundlegend geändert. 2005 stellte Google einen neuen Kartentyp vor, der „Slippy Map“, „Tiled web map“ oder „tile map“ genannt wird.

Das Konzept dieser Kartendarstellung beruht dabei primär auf der Teilung der Karte in verschiedene Kacheln und wird in folgendem Abschnitt erläutert.

Eigenschaften von Tile Maps

Kachel-basierte Kartensysteme weisen mehrere Kernfunktionalitäten auf, die sie von anderen Kartensystemen unterscheiden. Die hier abgebildete Liste nach Ioup und Sample zeigt die wichtigsten Kernfunktionen auf.

1. Kartenansichten basieren auf mehreren diskreten Zoomstufen, die jeweils einem festen Kartenmaßstab entsprechen.
2. Mehrere Bildkacheln werden verwendet, um eine einzelne Kartenansicht zu virtualisieren.
3. Bildkacheln sind unter Verwendung eines diskreten Adressierungsschemas zugänglich.
4. Auf einem Serversystem gespeicherte gekachelte Bilder werden mit minimalem Verarbeitungsaufwand an den Client gesendet. Dabei wird soviel Verarbeitung wie möglich im Voraus durchgeführt(vgl. Sample & Ioup[27] S.2).

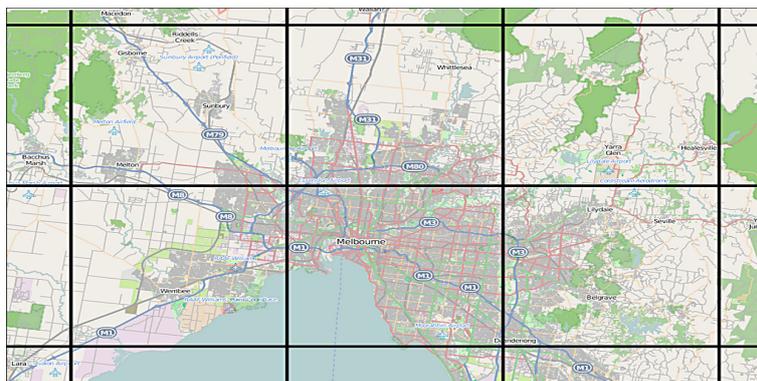


Abbildung 4: Tile Map von Melbourne, Australien. Quelle: Wikipedia [28] o.S.

Kachelschemata

Die Repräsentation von verschiedenen Detailgraden der Karte⁶ wird bei tile maps über die Zoomfunktion gesteuert. Je nach Zoomstufe werden Details der Karte verringert oder mehr Informationen angezeigt. Die Technik hinter dieser Art der Visualisierung wird durch einen

⁶ In der Literatur oftmals Level Of Detail (LOD) genannt.

Web-Map-Server gewährleistet.

Die Karte wird dabei anhand des gewählten festen Kartenmaßstabs gerendert und in Kachelform angezeigt. Dieser Prozess lässt sich mithilfe der tile pyramid verdeutlichen, die im folgenden Abschnitt vorgestellt wird.

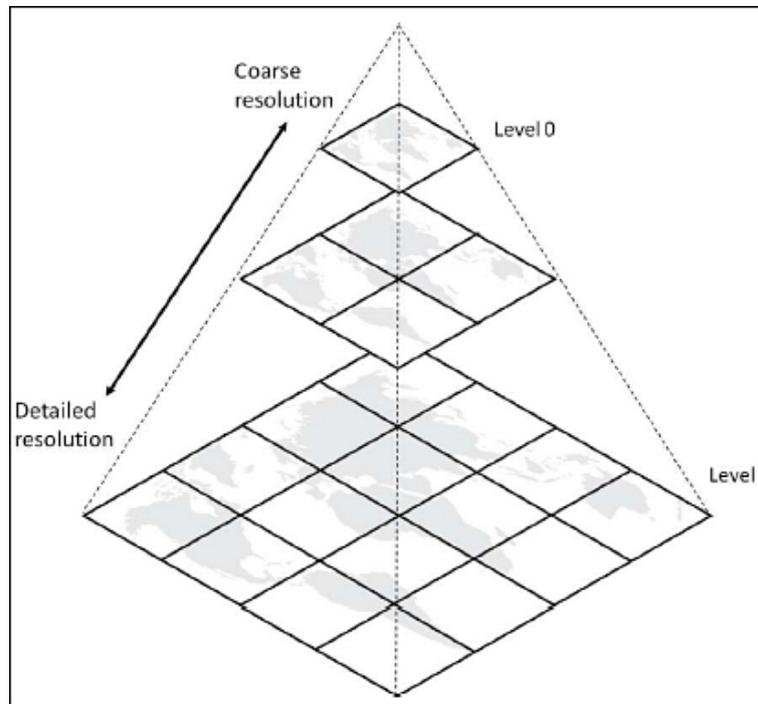


Abbildung 5: Tile pyramid nach Garcia et al.. Quelle: Garcia et al. [29] S.26.

Die Abbildung zeigt den Aufbau der Kachelpyramide der Karte. An der Spitze der Pyramide wird die Karte durch eine einzige Kachel abgebildet (LOD 0). Je detailreicher die Repräsentation gestaltet werden soll, desto mehr Kacheln werden für die Darstellung benötigt. Für jedes zusätzliche LOD werden in diesem Beispiel vierfach so viele Kacheln benötigt, wie im vorherigen LOD[29].

Diese Aufteilung ist dabei von der Konvention der tile web map abhängig, der De-facto-Standard ist die Google Maps Konvention, nach der viele Karten im Internet aufgebaut sind. Sie umfasst unterschiedliche Merkmale, die hier in Listenform aufgeführt sind:

Google Maps Konvention

1. Kartenkacheln besitzen gleiche Größe und haben eine Ausdehnung von 256 x 256 Pixeln.
2. Auf Zoomstufe 0 (LOD 0) wird die Welt durch eine Kachel repräsentiert.
3. Die Kartenprojektion der Darstellung ist Web Mercator⁷ mit WGS 84 als Datum.
4. Jede Kachel auf dem Zoomlevel k wird durch vier gleichgroße Kacheln auf Zoomlevel k+1 ersetzt.

⁷ EPSG: 900913

5. Die Nummerierung einer Kachel wird durch zwei Integer - Variablen (x,y) vorgenommen, X zeigt die Spalte und Y die Zeile der Kachel an(vgl. Stefanakis[30] S.8).

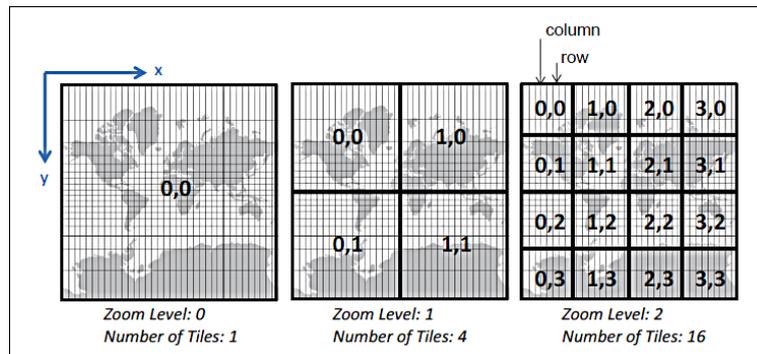


Abbildung 6: Google Maps Tilling Scheme nach Stefanakis. Quelle: Stefanakis [30] S.9.

Abbildung sechs zeigt das Kachelschemata von Google Maps. In jeder Zoomstufe kann eine beliebige Kartenkachel durch eine X und Y - Position adressiert werden.

4.3 Raster & Vektor - Tiles

Kachelbasierte Darstellungen beruhen auf den zugrundeliegenden Geodaten, die entweder im Raster oder Vektorformat vorliegen. In diesem Teilkapitel sollen die wesentlichen Merkmale beider Formate aufgeführt und miteinander verglichen werden.

Rasterdaten

Im Rasterdatenmodell werden Bildinhalte als quadratische Rasterzellen in einer zweidimensionalen Datenmatrix abgebildet(vgl. Christiansen[31] o.S.). Die Adressierung erfolgt dabei über die Angabe der Reihen- und Spaltenanzahl. Beispielhaft für räumliche Information in einer Rastergrafik folgende Abbildung:

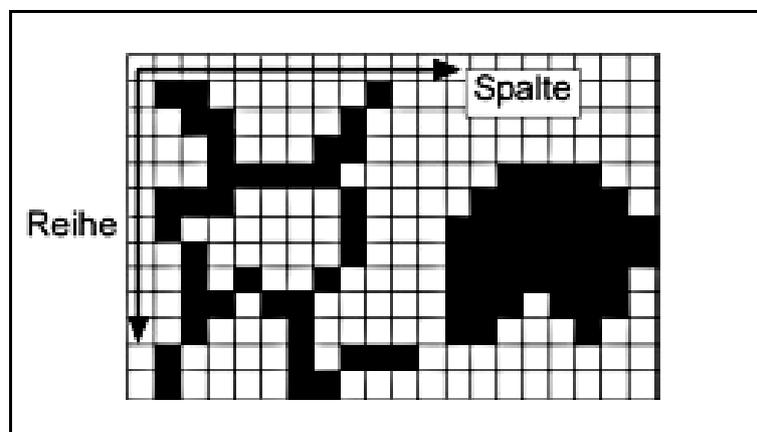


Abbildung 7: Räumliche Information im Rastermodell nach de Lange. Quelle: verändert nach de Lange [13] S.26.

Für diese Grafik beinhaltet jede Rasterzelle, auch Pixel genannt Informationen, wie beispielsweise einen Farbwert. Diese werden zusammen mit der Matrix abgespeichert.

Um den Platzverbrauch solcher Geodaten zu reduzieren, komprimiert man die Rasterdaten, bevor sie gespeichert werden. Aktuelle Formate der Speicherung sind unter anderem das Joint Photographic Experts Group (JPEG) oder das Portable Network Graphic (PNG)- Format(vgl. Sample & Ioup[27] S.27).

Vektordaten

Das Vektordatenmodell bildet die Welt durch die elementaren geometrischen Elemente Punkt, Linie und Fläche (den sogenannten Features) ab. Diesen werden thematische Charakteristika zugeordnet, die das Geoobjekt definieren. Dabei ist die Beziehung zwischen Attributen und Features notwendig für die Beschreibung der Daten(vgl. Bartelme[32] S.65). Abbildung acht zeigt mögliche geometrische Bestandteile des Vektormodells.

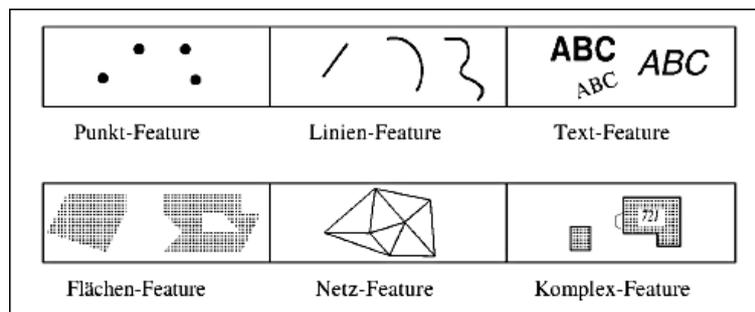


Abbildung 8: Vektormodell und seine geometrischen Bestandteile (Features) nach Bartelme. Quelle: verändert nach Bartelme [32] S.65.

Erzeugen der Kachelemente

Um aus Geodaten Kacheln zu erzeugen existieren unterschiedliche Ansätze, die sich in zwei Gruppen einteilen lassen. Dabei unterscheidet man zwischen Systemen, die die Bilddaten vorher erzeugen (prerendern) und Systemen, die diesen Prozess bei Bedarf ausführen (on-the-fly).

Bei Rasterdaten werden die Bilder vorher erzeugt, da Bildbearbeitungsschritte wie beispielsweise Skalieren oder Reprojizieren der Kachel zu viel Zeit für eine On-the-fly-Prozessierung benötigen.

Zudem bedingt der pyramidenförmige Aufbau (vgl. Abbildung 4) des Schemata eine spezielle Abfolge der Bilderzeugung, die ebenfalls Zeit benötigt(vgl. Sample & Ioup[27] S.81f.).

Die Schritte der Kachelerzeugung für Rasterdaten sind hier verkürzt als beispielhafter Algorithmus aufgeführt und werden in der Folge erklärt.

Kachelset- Algorithmus verändert nach Sample & Ioup

1. Festlegen des Basislevels⁸ des Kachelsets.
2. Definition der geographischen Grenzen des Aggregats.
3. Kacheln werden anhand der Quelldatei erzeugt und gespeichert.

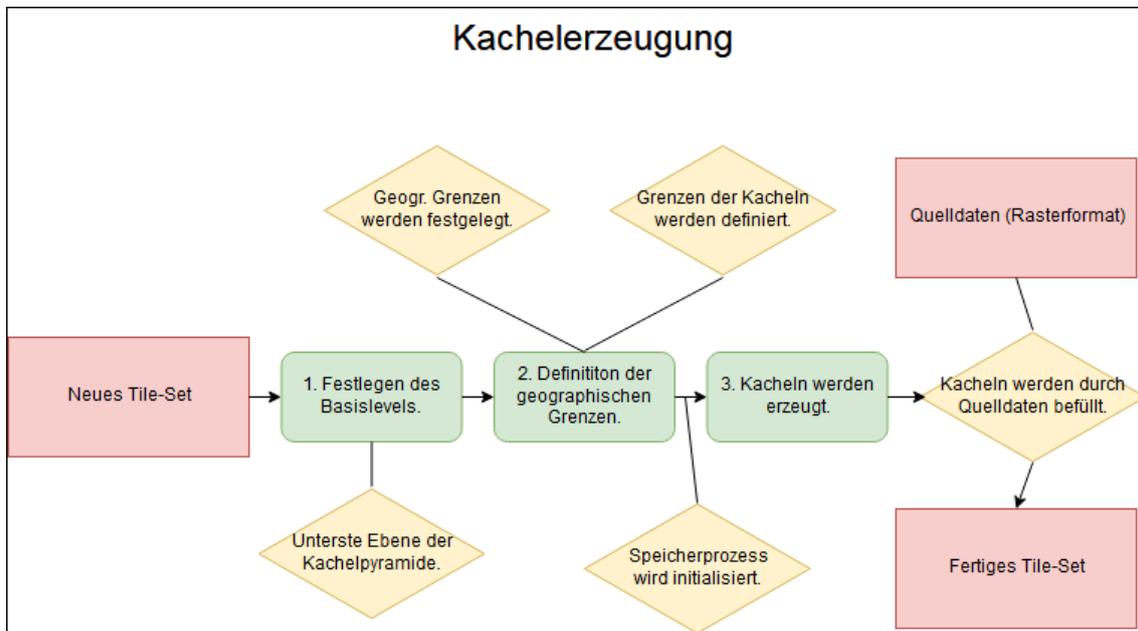


Abbildung 9: Kachelerzeugung aus Rasterdaten. Quelle: Eigene Abbildung.

Im ersten Schritt des Algorithmus wird das Basislevel des Kachelsets bestimmt. Es bietet den Ausgangspunkt zur Erzeugung weitere LOD-Ebenen und bildet die unterste Ebene der Kachelpyramide.

Der zweite Schritt umfasst die Festlegung der geographischen Koordinaten des Aggregats sowie der Grenzen der einzelnen Kacheln, ausgedrückt in Kachelkoordinaten. Beide sind wichtig um eine Zuordnung von Kachel zu Originalbild zu gewährleisten, die im weiteren Verlauf dazu benutzt wird, um einzelne Kacheln mit Bildausschnitten zu befüllen.

Zwischen Schritt zwei und drei wird der Speichermechanismus initialisiert, der das im Algorithmus erstellte Aggregat abspeichert.

Im letzten Schritt wird für jede Kachel das entsprechende Bildfragment der Quelldatei ermittelt und gespeichert. Dieser Prozess wird mithilfe des Cache der Quelldatei durchgeführt, nach Abschluss wird dieser wieder geleert(vgl. Sample & Ioup[27] S.83, 88f.).

Der dargestellte Algorithmus ist dabei als rudimentärer Ansatz zu verstehen, der noch viele Verbesserungsmöglichkeiten besitzt. Trotzdem stellt er eine Möglichkeit dar, aus Rasterdaten ein Tileset zu erzeugen, welches für die Visualisierung von Geodaten genutzt werden kann.

⁸ Das Basislevel beschreibt die unterste Ebene der Tilepyramide und zeichnet sich durch die größte Nummer des LOD innerhalb der Pyramide sowie die höchste Auflösung aus.

Die Erzeugung von Vektor-Kacheln hat viele Gemeinsamkeiten mit dem dargestellten Prozess. Der Hauptunterschied liegt darin, dass Vektordaten in gekachelte Bilder gezeichnet werden, anders als bei Rasterdaten, bei denen Kacheln der Quelldatei benutzt werden(vgl. Sample & Ioup[27] S.194).

Zudem ermöglicht das Vektorformat ein on-the-fly-Rendering der Daten, welches Vor- und Nachteile für die Prozessierung bedeutet, die im Verlauf dieses Kapitels erläutert werden.

Formatierung von Raster- und Vektortiles

Ein großer Unterschied zwischen Raster- und Vektortiles ist die Art der Speicherung. Rastertiles werden zumeist im PNG oder JPEG-Format gespeichert(vgl. 4.3). Die Formate ermöglichen dabei eine Komprimierung der Daten, die entweder verlustfrei (PNG) oder verlustbehaftet (JPEG) möglich ist.

Für Vektortiles sind aufgrund der Komplexität der Daten eigene Formate, wie beispielsweise Mapnik Vector Tiles oder Mapbox Vector Tiles (MVT) entwickelt worden. Sie liefern analog zu Rasterdaten eine Komprimierung der Datengrundlage durch unterschiedliche Techniken, wie beispielsweise die Speicherung in einer Datenbank.

Um die beiden Formate zu veranschaulichen, ist hier eine Gegenüberstellung der Firma Environmental Systems Research Institute (ESRI) Inc. abgebildet. Das Bild zeigt einen Ausschnitt

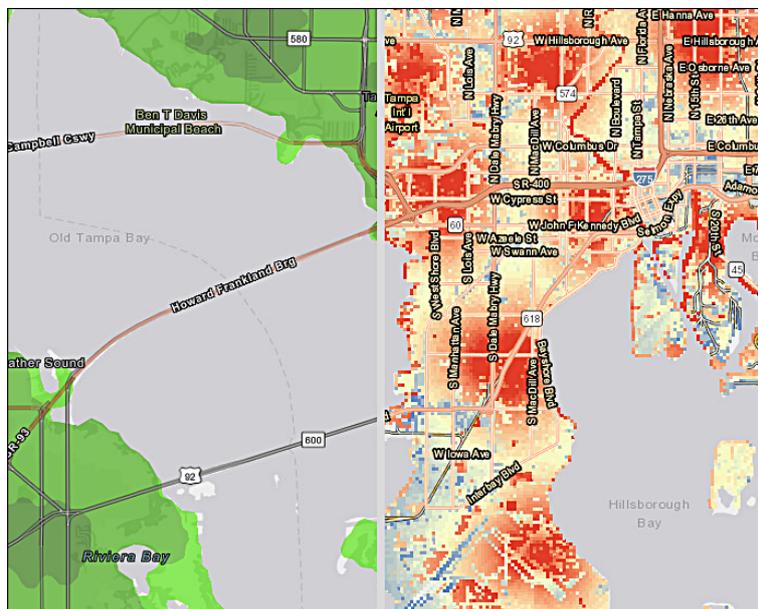


Abbildung 10: Vergleich von Raster- und Vektortiles. Quelle: verändert nach ESRI[33].

des US-Bundesstaats Florida. Im linken Teil wird dieser durch Rasterkacheln, im rechten durch Vektorkacheln dargestellt.

Gegenüberstellung beider Formate

Abbildung 10 deutet an, dass die Visualisierung von Raster- und Vektortiles einige Unterschiede aufweist. Im folgenden Abschnitt werden die Vor- und Nachteile beider Formate aufgeführt, um eine Gegenüberstellung zu ermöglichen.

Vorteile von Vektor - Tiles

1. Vektor - Tiles werden vom Clienten on - the - fly gerendert. Durch diese Technik können Karten für den Nutzer individuell gestaltet werden. So besteht die Möglichkeit, das Styling einer Vektor - Tile - basierten Karte zu ändern, ohne dass ein neuer Kartenlayer geladen werden muss.
2. Die Vektordaten beinhalten auch Objektgeometrien, die es ermöglichen einfache Geoverarbeitungsschritte durch den Clienten durchzuführen.
3. Die on - the - fly - Prozessierung sorgt dafür, dass das Pre-rendern der Daten obsolet wird. So kann die Zeit einer Aktualisierung der Daten verringert werden, was insbesondere im Hinblick auf zeitkritische Daten wichtig sein kann (vgl. Gaffuri[34] S.88f.).
4. Die Größe eines Vektor - Tiles ist normalerweise kleiner als die eines Raster - Tiles. Dieser Sachverhalt sorgt dafür, dass ein schnellerer Datentransfer möglich ist und weniger Bandbreite benötigt wird.
5. Da die Vektordaten beim Clienten verfügbar sind, können sie ohne Erhöhung der Bandbreite skaliert werden. Das gilt neben der Karte auch für Symbole oder Label (vgl. Cavazzi[35] S.11).

Nachteile von Vektor - Tiles

1. Grafikkonflikte oder Informationsverluste können anhand der Kachelgrenzen auftreten (vgl. Cavazzi[35] S.11). Dadurch werden weitere Arbeitsschritte notwendig, wenn beispielsweise ein Polygon durch den Tiling Prozess in verschiedene Teile aufgeteilt ist und wieder zusammengefügt werden muss.
2. Dadurch dass die Daten beim Clienten verfügbar sind, können sie vom Nutzer für andere, möglicherweise illegale Zwecke weiterverwendet werden (vgl. Ingensand et al.[36] S.1f.).
3. Vektor - Tiles haben einen großen Leistungsbedarf (vgl. Sample & Ioup[27] S.195), da das Rendering vom Clienten geleistet wird .
4. Geographische Daten benötigen unter Umständen eine Vorprozessierung, um vom Clienten gezeichnet werden zu können(vgl. OSGeo[37] o.S.).

Vorteile von Raster - Tiles

1. Rastertiles können vergleichsweise einfach generiert und implementiert werden(vgl. Yu et al.[38] S.2).
2. Etablierte Technologie, deren Umsetzung über Jahre getestet und verbessert wurde.

Nachteile von Raster - Tiles

1. Eingeschränkte Möglichkeiten die Kartenansicht durch neue Styles oder Farbschemata zu verändern, da vorgefertigte Kacheln vom Clienten geladen werden(vgl. ESRI[39] S.2).
2. Raster - Tiles benötigen im Vergleich zu Vektor - Tiles großen Speicherplatz sowie eine hohe Bandbreite(vgl. Yu et al.[38] S.2).

Die Auflistung der Vor- und Nachteile zeigt, dass Vektor - Tiles in der Literatur als fortschrittlicher angesehen werden und dem Stand der Technik auf dem Gebiet der Geovisualisierung entsprechen. Die Forschung an Vektor - Tiles befindet sich dabei noch in der Entwicklung, besitzt aber mehrere Standards und Datenformate, die im Web - Mapping der heutigen Zeit Anwendung finden.

4.4 Mapbox Vector Tiles

Im Prototypen des LGLN werden Mapbox Vector Tiles zur Visualisierung der Kartenelemente genutzt. Dieser Datentyp basiert auf der Mapbox Vector Tile Specification, einem quellenoffenen Standard, der unter der Creative Commons Lizenz 3.0 veröffentlicht ist. Der Standard wurde am 13.04.2014 eingeführt und mehrfach aktualisiert, zuletzt am 19.01.2016 in der Version 2.1. Das Format basiert dabei auf einer Encodierung in Protocol Buffers, einem Datenformat von Google, welches in folgendem Abschnitt kurz erklärt wird.

4.4.1 Protocol Buffers

Protocol Buffers sind ein von Google entwickelter Mechanismus zum Serialisieren von strukturierten Daten. Die Ausrichtung des Datenformats liegt dabei auf einer intelligenten, schnellen und einfachen Serialisierung der Daten. Das Schemata eines Protocol Buffers wird in der Message-Datei festgehalten. Sie umfasst mehrere Informationen, die als Namen - Wert - Paare in einer .proto-Datei gespeichert werden. Als Beispiel ist auf der nächsten Seite ein Schemata abgebildet, welches Informationen über eine Person zur Verfügung stellt:

```

message Person {
    required string name = 1;
    required int32 id = 2;
    optional string email = 3;

    enum PhoneType {
        MOBILE = 0;
        HOME = 1;
        WORK = 2;
    }

    message PhoneNumber {
        required string number = 1;
        optional PhoneType type = 2 [default = HOME];
    }

    repeated PhoneNumber phone = 4;
}

```

Der Codeabschnitt zeigt den Aufbau der Datei, für jeden Message-Typen gibt es ein oder mehrere Felder, die einen Namen und Wert umfassen. Felder können, wie beim Namen und der ID einer Person notwendig oder wie bei der Email optional sein. Der Protocol Buffers compiler kann mithilfe dieser Datei Personen-Protocol Buffers erzeugen und diese serialisieren.

Im Vergleich zu anderen Formaten wie beispielsweise XML, zeigen Protocol Buffers deutlich bessere Performance: Die Geschwindigkeit ist im Vergleich zu XML 20 bis 100 mal schneller und der Speicherplatzbedarf um drei bis zehnmal kleiner(vgl. Google[40] o.S.).

4.4.2 Geometrikodierung

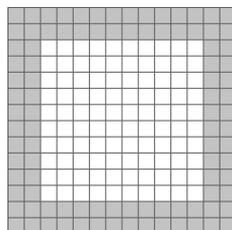


Abbildung 11: Leeres Vektor-Tile. Quelle: verändert nach Mapbox[41].

Um geographische Informationen in einer Vektor-Kachel zu kodieren, müssen die geographischen Koordinaten zunächst in das Bezugskoordinatensystem des Vektortiles überführt werden. Eine leeres Mapbox-Vektor-Tile besteht aus 10 x 10 Kacheln und einem Buffer von zwei Zellen(vgl. Abbildung 11). Dabei wird das Bildschirmkoordinatensystem benutzt, welches in der linken oberen Ecke den Koordinatenursprung hat. Die X-Achse befindet sich rechts vom, die

Y-Achse unter dem Ursprung.

Geometrien sind in diesem System als 32-bit Integer ohne Vorzeichen kodiert. Jedes Integer kann dabei einen Befehl oder einen Parameter beinhalten.

Befehle bestehen aus einer ID, die die Art des Befehls festlegt und je nach Befehl eine Anzahl an Durchführungen. Um ein Polygon zu erzeugen, kann beispielweise folgende Reihe an Befehlen abgearbeitet werden⁹:

```
MoveTo(1,2), LineTo(3,-1), LineTo(3,4), LineTo(-4,2), ClosePath()
```

Dazu gehören außerdem Parameteranweisungen, die Informationen über die jeweiligen Parameter beinhalten. Das durch die Befehle entstandene Polygon in der Kachel sieht folgendermaßen aus:

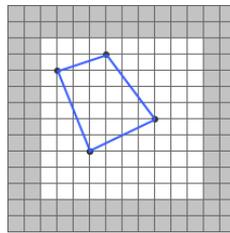


Abbildung 12: Gefülltes Vektor-Tile. Quelle: verändert nach Mapbox[41].

Die zu den Geometrien gehörenden Attribute werden für jedes Feature als Integer-Paar gespeichert. Die erste Zahl identifiziert dabei den Schlüssel, die zweite Zahl den Wert des Attributs(vgl. Mapbox [42]).

4.4.3 Styling

Da MVT eine on-the-fly-Prozessierung ermöglichen, kann der Style der Karte ohne großen Aufwand geändert werden. Im Prototypen wird jedes Styling der Karte als Mapbox-Style vorgehalten. Die dafür entwickelte Spezifikation schreibt die Möglichkeiten der Gestaltung der Karte vor und ist von Mapbox entwickelt worden.

Ein Style wird dabei durch ein JavaScript Object Notation (JSON)-Dokument repräsentiert und zeichnet sich durch einen strukturierten, teilweise vorgeschriebenen Aufbau aus. Die zentralen Eigenschaften des Styles werden in den Root Properties festgehalten. Diese bestehen aus notwendigen und optionalen Angaben, die die Ausprägung des Styles beschreiben.

Notwendige Attribute sind die Versionsnummer der Style-Spezifikation, die Datenquelle sowie ein Layer-Array.

Optional ist es zudem möglich Metadaten, eigene Schriftarten, Beleuchtung und weitere Stylingoptionen hinzuzufügen(vgl. Mapbox[43] o.S.).

Für den Prototypen werden Daten aus ATKIS verwendet. Diese werden durch Filterung der Objektart zu unterschiedlichen Klassen zugeordnet und dargestellt. Um diesen Prozess zu ver-

⁹ Die Befehle sind aus Gründen der Übersichtlichkeit nicht im Binärformat angegeben.

anschaulichen wird hier der Abschnitt des JSON-Dokuments der Klasse Gewässerflächen dargestellt:

```
{
  "id": "Gewaesserflaechen",
  "type": "fill",
  "metadata": {"map-editor:group": "water"},
  "source": "ATKIS",
  "source-layer": "gew01_f",
  "filter": [
    "all",
    ["in", "objart_txt", "AX_StehendesGewaesser",
      "AX_Fliessgewaesser", "AX_Hafenbecken"]
  ],
  "layout": {},
  "paint": {"fill-color": "#08519c"}
}
```

Aus der Datengrundlage ATKIS wird der Layer *gew01_f* ausgewählt und alle Features selektiert, welche die Signaturen *AX_StehendesGewaesser*, *AX_Fliessgewaesser* oder *AX_Hafenbecken* besitzen. Die Abfrage der Daten erfolgt dabei über die Art des Objektes, die Füllfarbe ist durch die JSON-Datei im hexadezimalen Farbsystem angegeben.

Anders als bei anderen Geoviewern des Landes, beruht das Styling nicht auf den Einträgen im Signaturenkatalog, sondern wird durch Mitarbeiter des LGLN festgelegt. Dies entspricht dem Ansatzpunkte des Landes, dass Basisvisualisierung die DTK ersetzen sollen (vgl. Kapitel 1.1).

Das Mapbox-Styling ist insbesondere für die Weiterentwicklung des Prototypen von Bedeutung, da es die Grundlage für die Kartendarstellung für Farbenfehlsichtige bietet.

4.5 Docker

Für die Entwicklungsphase des Prototypen wird die freie Software Docker verwendet. Das Programm beruht dabei auf dem Prinzip der Container-Visualisierung, die Software ermöglicht es also, ein oder mehrere virtuelle Maschinen (VM) zu erzeugen. Im Gegensatz zu einer gewöhnlichen VM benötigen Docker-Container deutlich weniger Ressourcen, da kein komplettes Betriebssystem emuliert werden muss. Aus diesem Grund wird Docker im Software Engineering für die Replikation von Entwicklungsumgebungen genutzt (vgl. Anderson[44] S.102f.).

Im Prototypen werden mehrere Container von Docker gestartet und die Abfolge mithilfe einer YAML¹⁰-Datei gesteuert. In Kapitel 5 wird diese Datei eingehend analysiert, um die Funktionsweise des Prototypen zu verdeutlichen.

¹⁰ YAML bedeutet YAML Ain't Markup Language und bezeichnet eine auf Unicode basierende Datenserialisierungssprache.

4.6 T - rex

T - rex ist ein auf die Veröffentlichung von MVT spezialisierter Vektor - Tile - Server. Das Programm ist in Rust¹¹ geschrieben und unterstützt räumliche Datenbanken wie PostGIS und verschiedene Vektorformate. Zudem besteht die Möglichkeit Mapbox - Styles einzubetten (vgl. Kalberer[45] o.S.).

Die Einstellung des Servers lassen sich über eine TOML - Konfigurationsdatei¹² steuern. In ihr werden die Datengrundlage, die Projektion und die Layer der zu veröffentlichen Daten gespeichert. Da der Server PostGIS unterstützt, ist es möglich, SQL - Abfragen an die verbundene Datenbank zu stellen. So können Layer definiert werden, die durch den Prototypen angesprochen werden können.

4.7 PostgreSQL & PostGIS

PostgreSQL ist ein 1996 erschienenes objektrelationales Datenbankmanagementsystem. Das Programm ist quellenoffen und wird von einer Open - Source - Community entwickelt (vgl. Momjiam[46] S.4).

PostgreSQL unterstützt dabei ACID - konforme Transaktionen und besitzt unterschiedliche Erweiterungen, von denen für die Arbeit PostGIS benutzt wird.

PostGIS erweitert PostgreSQL um räumliche Datenbanken, indem Geometrietypen und Funktionen hinzugefügt werden. Die dabei berücksichtigten Geometrien entsprechen in Teilen der Simple Feature Spezifikation für SQL von OpenGIS(vgl. Strobl[47] S.891).

Im Prototypen wird mithilfe von PostgreSQL und PostGIS eine Datenbank erzeugt, die die Grundlage des Geoviewers darstellt. Über pgAdmin, einem Administrations - Werkzeug von PostgreSQL wird die Datenbank verwaltet. Die genaue Funktionsweise und der Aufbau der Datenbank wird in Kapitel 5 erläutert.

4.8 Google Lighthouse

Die Evaluation des Prototypen wird in der Arbeit zu einem Teil mithilfe des Open Source Werkzeugs Lighthouse von Google durchgeführt. Das Programm ist in aktuellen Google Chrome Browsern vorinstalliert und basiert auf mehreren Tests, die unterschiedliche Aspekte einer Webseite prüfen. Für jede der vier Kategorien Performance, Accessibility, Best Practices und Search Engine Optimization(SEO) verteilt das Programm einen Punktwert von 0 - 100 und schlägt dem Entwickler Verbesserungsmöglichkeiten vor. Um die Kategorien und das Punktesystem besser zu verstehen, sind hier die einzelnen Tests kurz beschrieben.

¹¹ Rust bezeichnet eine von Mozilla Research entwickelte Programmiersprache.

¹² TOML steht für Tom's Obvious, Minimal Language und wurde von Tom Preston - Werner entwickelt.

Performance

Die Leistung der Webseite wird vom Programm durch sechs Metriken berechnet. Dazu zählen First Contentful Paint, First Meaningful Paint, First CPU Idle, Time to Interactive, Speed Index und Estimated Input Latency.

Die nachfolgende Tabelle zeigt eine Übersicht über die Leistungsindikatoren und beschreibt, was sie messen:

1. First Contentful Paint - Misst die Zeit in Sekunden, die der Browser benötigt, um den ersten Inhalt der Seite¹³ zu zeichnen.
2. First Meaningful Paint - Misst die Zeit in Sekunden, die der Browser benötigt um den ersten bedeutenden Inhalt der Seite zu zeichnen. Welcher Inhalt für die Seite als bedeutend gilt, wird über Heuristiken bestimmt.
3. First CPU Idle - Beschreibt die Zeit in Sekunden die vergeht, bis die Seite auf Userinput reagieren kann. Die Berechnung basiert ebenfalls auf Heuristik.
4. Time to Interactive - Berechnet die Zeit in Sekunden die benötigt wird, um eine interaktive Webseite abzurufen. Interaktiv ist dabei definiert als Zeitpunkt, an dem die Seite auf Nutzerinteraktion innerhalb von 50 Millisekunden reagiert und das First Contentful Paint angekommen ist.
5. Speed Index - Geschwindigkeitsindex, der die Zeit in Sekunden misst, in der die erste Seite vom Browser gezeichnet wird.
6. Estimated Input Latency - Misst die Zeit in Millisekunden, in der die Applikation auf Nutzereingaben reagiert. Der Wert sollte dabei unter 50 liegen.

Die durch die Metriken ermittelten Zeitangaben werden danach in ein Punktesystem überführt, das von 0-100 reicht und die Leistung in einer bestimmten Kategorie ausdrückt. Für diese Zuweisung wird eine logarithmische Normalverteilung benutzt, die aus den Leistungsdaten von realen Webseiten ermittelt wird(vgl. Google Chrome [48] o.S.).

Die so entstandenen Punkte der Kategorien werden dann nach einer Gewichtung¹⁴ in einen Punktwert für die Leistung der Webseite überführt.

Accessibility

Die Barrierefreiheit der Applikation wird mithilfe von 34 Tests ermittelt. Um unnötige Längen zu vermeiden, sind hier nur diejenigen Prüfungen aufgeführt, die maßgeblichen Einfluss auf den Punktwert haben¹⁵. Die Grundlage der Tests sind die Web Content Accessibility Guidelines (WCAG).

¹³ Beispielsweise ein Bild oder Text.

¹⁴ Die Gewichtung aller Kategorien und alle Tests sind unter <https://docs.google.com/spreadsheets/d/1Cxzhy5ecqJCucdf1M0i0zM8mIxC7mmx107o5nj38Eo/edit#gid=0> verfügbar.

¹⁵ Maßgeblich bezeichnet dabei eine Gewichtung mit Faktor 9 oder mehr(vgl. Fußnote 14).

1. Button name - Knöpfe auf der Webseite müssen einen Namen besitzen, um von Nutzern benutzt werden zu können, die auf einen Bildschirmleser angewiesen sind.
2. Bypass - Jede erreichbare Seite sollte eine Möglichkeit bieten, zum Hauptinhalt der Seite zu gelangen.
3. Label - Jedes Element benötigt ein Label, sodass Bildschirmleser den Zweck des Elements erkennen.
4. Link name - Links sollen klickbare Wörter umfassen, um die Inhalte für Nutzer und Suchmaschinen verständlich zu machen.

Jeder Test kann entweder bestanden oder nicht bestanden werden und fließt danach mit seinem Gewicht in die Gesamtbewertung der Accessibility ein.

Best Practices

Die Bewertung für moderne Webstandards entsteht aus 16 Tests, die aktuelle Webtechnologien für die Applikation fordern. Unter anderem wird geprüft ob keine veralteten Bibliotheken benutzt werden, HTTP2 statt HTTP verwendet wird oder die aktuelle Position des Geräts beim Start der Seite abgefragt wird.

Die Ergebnisse gehen mit 1/16 in die Gesamtpunktzahl mit ein, dementsprechend findet keine Gewichtung der einzelnen Tests statt.

Search Engine Optimization

Mithilfe der Testabfolge soll die Sichtbarkeit der Webseite bewertet werden. Dazu muss die Applikation 13 Tests durchlaufen, die ungewichtet in das Endergebnis für SEO eingehen. Die Prüfungen zielen dabei darauf ab, dass die Webseite von Suchmaschinen möglichst einfach gefunden und dargestellt werden kann.

Diese vier Kategorien dienen im Abschnitt Evaluation als Bewertungskriterien für den Prototypen. Das Programm erzeugt für jeden Test einen Testbericht als JSON - Datei, der neben dem Punktwert auch die Einzelbewertung der jeweiligen Kategorie umfasst. Zudem werden Veränderungen am Quellcode der Webentwicklung vorgeschlagen, die zu einer Leistungsverbesserung führen.

4.9 Farbenfehlsichtigkeit

Für den Prototypen werden im Verlauf der Arbeit mehrere Styles entwickelt, welche die Bedürfnisse von Menschen mit Farbenfehlsichtigkeit berücksichtigen. Aus diesem Grund soll die Krankheit und deren Ausprägungen im folgenden Abschnitt vorgestellt werden.

Die zwischen 1856 und 1867 entwickelte Young-Helmholtz-Theorie besagt, dass das menschliche Auge aufgrund des Baus der Netzhaut nur für die drei Farben Grün, Rot und Blau empfindlich ist. Demnach werden alle anderen Mischfarben aus einer Komposition dieser drei Grundfarben erzeugt (vgl. Welsch & Liebmann[49] S.227).

Grund dafür sind die Zapfen, eine Art Fotorezeptor im Auge, die für das Sehen des Menschen unter Tageslichtbedingungen zuständig sind. Sie liegen in drei unterschiedlichen Formen in der Netzhaut vor, die durch unterschiedliche Absorptionsmaxima bezüglich des Spektrums des Lichts gekennzeichnet sind. Abbildung 13 zeigt die Absorptionsspektren der blauen, grünen und

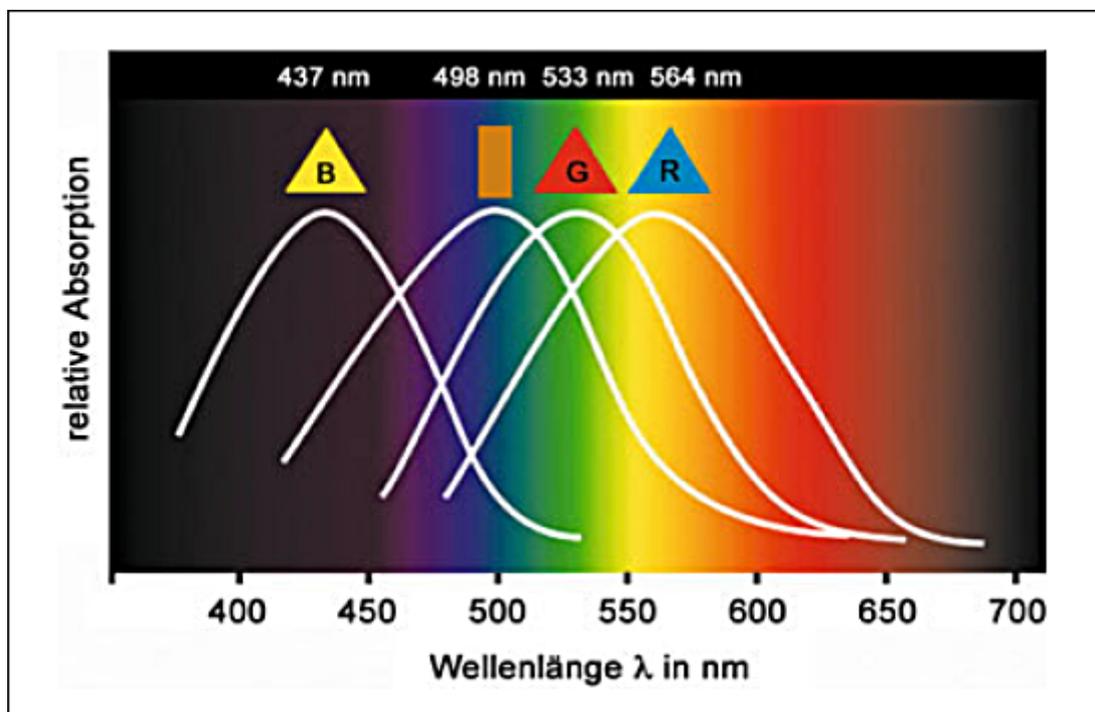


Abbildung 13: Absorptionsspektren der Sehzelle. Quelle: verändert nach Welsch & Liebmann[49] S.241

roten Zapfen im Wellenlängenbereich des sichtbaren Lichts. Die Maxima der Spektren liegen dabei bei 437, 533 und 564 nm.

Mithilfe dieser Fotorezeptoren ist es gesunden Menschen möglich, Farben unter Tageslichtbedingungen zu unterscheiden.

Sind die Zapfen durch eine organische Störung beeinträchtigt spricht man von Farbenfehlsichtigkeit. Sie kann in unterschiedlichen Formen vorkommen, die sich in zwei Gruppen aufteilen lassen, bei denen die Reizzentren der Netzhaut entweder teilweise oder ganz ausfallen.

In der Wissenschaft spricht man bei der ersten Gruppe von anomalen Trichromaten, bei der zweiten Gruppe von Dichromaten. Die Bezeichnung der genauen Krankheit ist abhängig von der Zapfenart, die durch die Farbenfehlsichtigkeit beeinträchtigt ist. Folgende Arten lassen sich unterscheiden:

1. Protoanomale (Rotsensitive Zapfen gestört.)
2. Deuteranomale (Grünintensive Zapfen gestört.)
3. Tritanomale (Blauintensive Zapfen gestört.)

4. Protanopen (Rotsensitive Zapfen ausgefallen.)
5. Deuteranopen (Grünintensive Zapfen ausgefallen.)
6. Tritanopen (Blauintensive Zapfen ausgefallen.)

Die Krankheit kann dabei angeboren oder erworben sein und betrifft ca. 8% der Männer und ca. 0,5% der Frauen(vgl. Luebbe[50] S.64).

Für die vorgestellten Krankheitstypen der Dichromaten werden in Kapitel 7 Mapboxstyles entwickelt, die die Besonderheiten der Krankheit berücksichtigen. Für jeden Nutzer des Prototypen soll der Kartenstyle so angepasst werden, dass die unterschiedlichen Landbedeckungen innerhalb der Karte unterscheidbar bleiben. Die Anpassung erfolgt dabei anhand der WCAG, eines internationalen Standards zur barrierefreien Gestaltung von Internetangeboten.

4.10 Web Content Accessibility Guidelines

Die WCAG sind ein internationaler Standard, welcher vom Wide Web Consortium entwickelt wird. Sie liegen momentan in Version 2.1 vor und bestehen aus unterschiedlichen Empfehlungen, die bezwecken sollen, dass Entwickler Webinhalte erreichbarer gestalten(vgl Cooper[51] S.1). Für die europäische Union ist dieser Standard aufgrund der EU Richtlinie 2016 / 2102 vom 26.Oktober 2016 verbindlich¹⁶(vgl. Europäische Union[52] o.S.). Für jede Empfehlung gibt es unterschiedliche Erfolgskriterien, die als Stufen der Konformität zu verstehen und als A, AA und AAA deklariert sind.

Der für diese Arbeit maßgebliche Teil der WCAG 2.1 ist Abschnitt 1.4 und hier insbesondere 1.4.3 Contrast(Minimum), 1.4.6 Contrast (Enhanced) und 1.4.11 Non-text Contrast.

Der Standard fordert in diesen Abschnitten für das Text zu Hintergrundverhältnis eine Mindestkontraste von 3:1(A), 4,5:1(AA) und 7:1(AAA). Zudem soll der Kontrast von nichttext-Elementen bei 3:1(AA) liegen.

Auf Basis dieser Richtlinien werden für Protanopen, Deuteranopen und Tritanopen Styles entwickelt, die dem WCAG entsprechen.

¹⁶ Für öffentliche Stellen ab 23.9.2019 für neue, ab dem 23.9.2020 für alte Webseiten. Ab 23.6.2021 für mobile Anwendungen.

4.11 Color Blindness Simulator

Der Color Blindness Simulator (Coblis) bietet eine Möglichkeit, Bilder von der trichromatischen Sicht eines gesunden Menschen in die dichromatische Sicht eines Farbenfehlsichtigen zu transformieren. Er wird in dieser Arbeit dazu verwendet, um die Weiterentwicklung des Prototypen zu bewerten. Dazu werden die in der Weiterentwicklung erstellten Farbpaletten mithilfe von Coblis in das dichromatische Pendant transformiert, um die Unterscheidbarkeit der unterschiedlichen Farben zu prüfen. Coblis basiert dabei auf einer JavaScript-Datei, welche im Webbrowser läuft¹⁷.

Aus Gründen der Übersichtlichkeit soll hier nur kurz auf die Art der Transformation eingegangen werden.

Das Script transformiert die RGB-Farbwerte des Bildes dabei in den XYZ Farbraum. Dieser Schritt wird in vielen unterschiedlichen Arbeiten zur Simulation von Farbenfehlsichtigkeit angewandt¹⁸. Danach werden die Farbwerte je nach Typ der Farbenfehlsichtigkeit geändert und wieder in den RGB-Farbraum zurück transferiert. Das so entstandene Bild nähert die Wahrnehmung eines Farbenfehlsichtigen an und kann so zur Evaluation der Farbenblindstyles dienen.

5 Prototyp

In diesem Kapitel soll der Aufbau und die Funktionsweise des Prototypen erläutert und anhand von Schaubildern dargestellt werden. Grundsätzlich kann man den Aufbau des Prototypen mit der im folgenden Abschnitt gezeigten Abbildung vergleichen. Die 2017 auf der Freie und Open Source Software für Geoinformationssysteme (FOSSGIS)-Konferenz veröffentlichte Grafik zeigt den Aufbau eines typischen Vektor Tile Stacks für Custom-Data.

Systemarchitektur und Aufbau der Entwicklungsumgebung des Prototypen mithilfe von Docker

Der ganze Stack des Prototypen befindet sich in einer virtuellen Umgebung in unterschiedlichen Docker Containern. Im ersten Schritt wird dazu mithilfe des Befehls `docker volume create -name bavi-atkis` Speicherplatz für die Daten des Containers reserviert. Danach wird der Befehl `docker compose-up` ausgeführt, der alle wichtigen Bestandteile des Prototypen in eigenen Containern erzeugt und durch eine YAML-Datei aufeinander abstimmt. Dieser Prozess läuft in mehreren Schritten ab, die in folgendem Abschnitt einzeln beschrieben werden.

¹⁷ Die Dokumentation und die Berechnung und deren Parameter sind unter https://github.com/MaPePeR/jsColorblindSimulator/blob/master/hcirn_colorblind_simulation.js ersichtlich.

¹⁸ Beispielsweise bei Muttaqin & Suwandi [53]

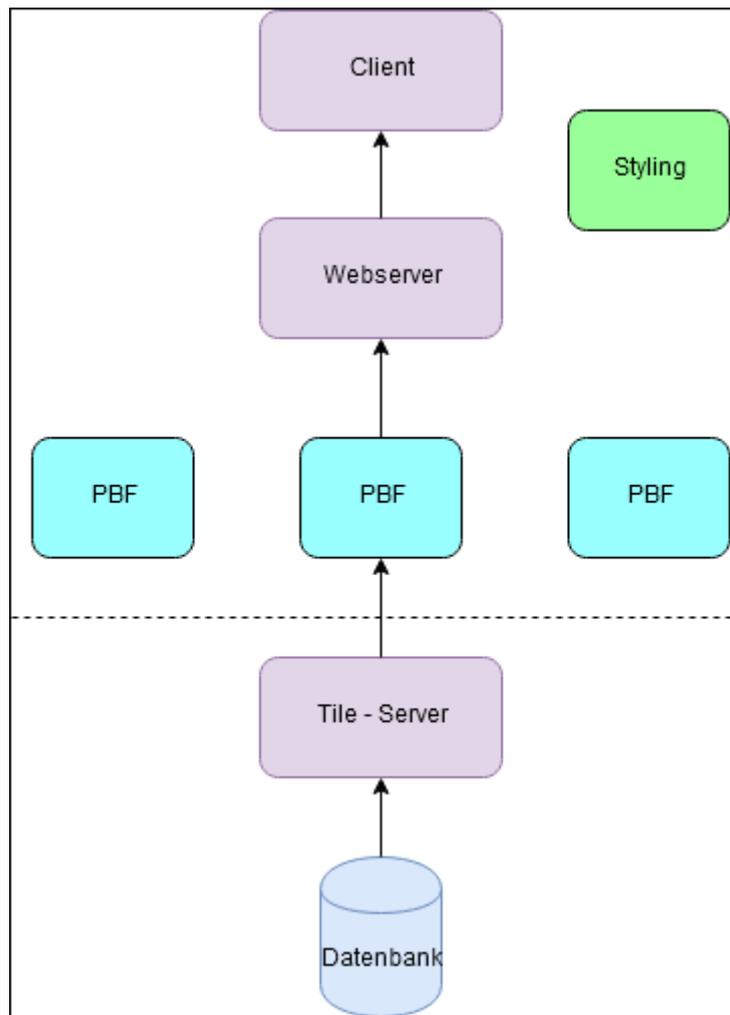


Abbildung 14: Aufbau eines Vektor Tile Stack für Custom-Data. Quelle: verändert nach Kalberer[45] o.S.

Im ersten Schritt wird eine PostgreSQL-Datenbank mit PostGIS -Erweiterung erzeugt. Sie kann über den Port 5432 angesprochen und mithilfe von pgAdmin verwaltet werden. Die Datenbank

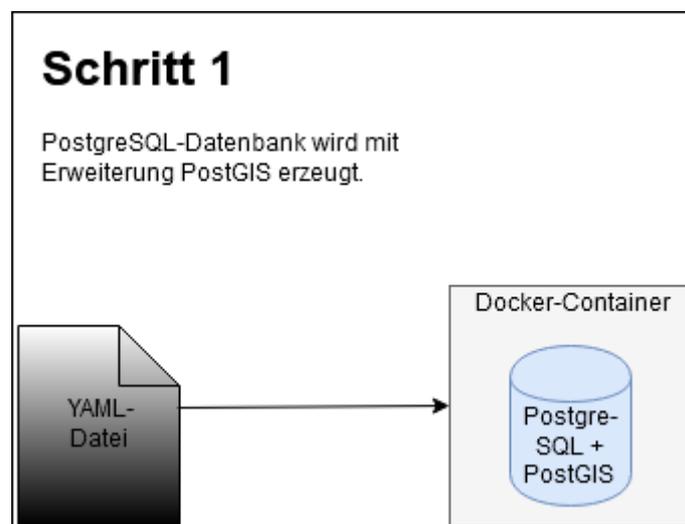


Abbildung 15: Schritt 1: Aufbau der Datenbank innerhalb von Docker. Quelle: Eigene Abbildung.

hält im ersten Schritt die Datengrundlage ATKIS vor. In ihr sind Informationen zu Gebäuden, Siedlungen, Verkehr, Vegetation, Wasserflächen, Relief und Gebieten¹⁹ gespeichert.

Im zweiten Schritt wird der Vektor-Tile-Server angebunden. Dies geschieht ebenfalls durch

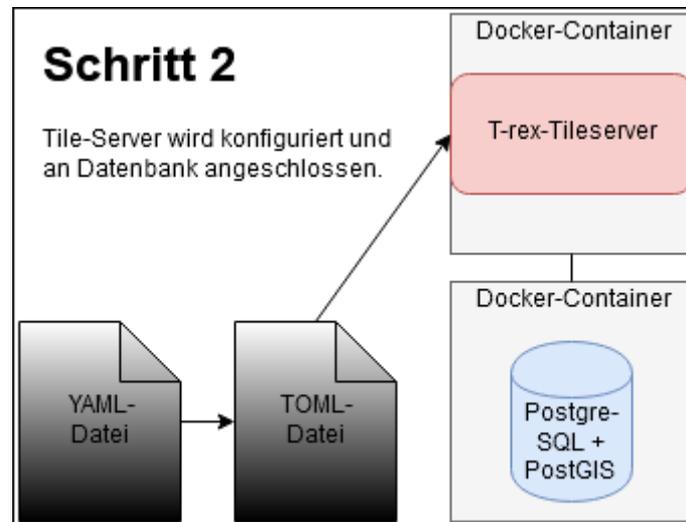


Abbildung 16: Schritt 2: Anbindung des Vektor-Tile-Server. Quelle: Eigene Abbildung.

die YAML-Datei, die die Konfigurationsdatei des T-rex-Servers ausführt. Der Aufbau der dafür benötigten TOML-Datei beinhaltet wichtige Parameter für die Bereitstellung von Vektor-Tiles:

1. Speicherort der Datenbank, um auf diese zuzugreifen.
2. Projektion und Ausdehnung des Datensatzes.
3. Layer, die als Vektor-Tiles geliefert werden sollen. Für jedes dieser Layer sind Name, Name innerhalb der Datenbank, Geometrie und Projektion abgespeichert.
4. Anbindung an den Webserver.
5. Ort des Cache für Vektor-Tiles.

Um einen Eindruck eines Layers zu vermitteln, aus dem für den Prototypen Vektor-Tiles entstehen, ist im Folgenden ein kurzes Beispiel abgebildet. Der abgebildete TOML-Codeabschnitt zeigt die Definition des Layers `bui01_f`, den Gebäuden, die im Prototypen dargestellt werden. Die Darstellung des Multipolygons erfolgt dabei im Universal Transverse Mercator (UTM)-Koordinatensystem Zone 32 N. Die Geometrien sind vereinfacht und werden erst ab Zoomstufe 13 auf der Karte angezeigt.

¹⁹ Beispielsweise Informationen zu Verwaltungsgebieten in Niedersachsen.

```

[[ tileset.layer ]]
name = "bui01_f"
table_name= "atkis.bui01_f"
geometry_field = "geom"
geometry_type = "MULTIPOLYGON"
srid = 3044
buffer_size = 10
simplify = true
minzoom = 13

```

Die TOML-Datei besitzt für jeden ATKIS-Layern in der Datenbank eine solche Beschreibung der Eigenschaften, die dafür sorgt, dass der T-rex-Server die Tiles in einer vom Entwickler festgelegten Art und Weise bereitstellen kann.

Im dritten und letzten Schritt wird der Webserver des Prototypen durch die YAML-Datei angesprochen. Durch diesen Aufruf wird das Frontend erzeugt, welches der Nutzer angezeigt bekommt. Dabei greift es zur Visualisierung der Daten auf die Vektor-Tiles des T-rex zurück. Die Grafik auf der nächsten Seite verdeutlicht diesen Ablauf. Hinzu kommt das Styling der Vektor-Tiles, das im Prototypen mithilfe von Mapbox GL realisiert ist. Die so entstandene

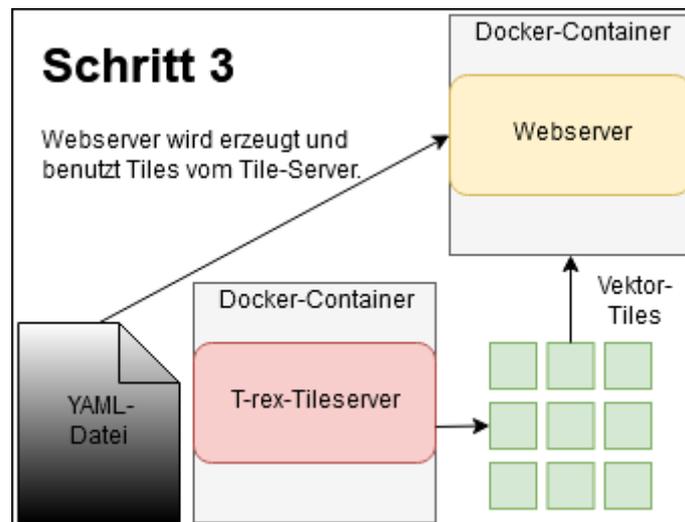


Abbildung 17: Schritt 3: Funktion und Lage des Webserver. Quelle: Eigene Abbildung.

Systemarchitektur ähnelt dem Aufbau stark der Abbildung 14 und nutzt dabei das Konzept der Replikation von Entwicklungsumgebungen²⁰ um eine einfache, ressourcensparende Entwicklung zu ermöglichen.

²⁰ vgl. 4.5

5.1 Webserver

Nach dem Aufbau der Entwicklungsumgebung wird in diesem Abschnitt der Webserver des Prototypen beschrieben. Er beeinflusst durch seine Komponenten maßgeblich die Visualisierung der Geodaten und ist demnach von besonderem Interesse.

Die Entwicklung des LGLN basiert auf dem Webapplikationframework Angular und wurde mit TypeScript, Syntactically Awesome Stylesheets (Sass) und HTML realisiert. Um einen Ein-

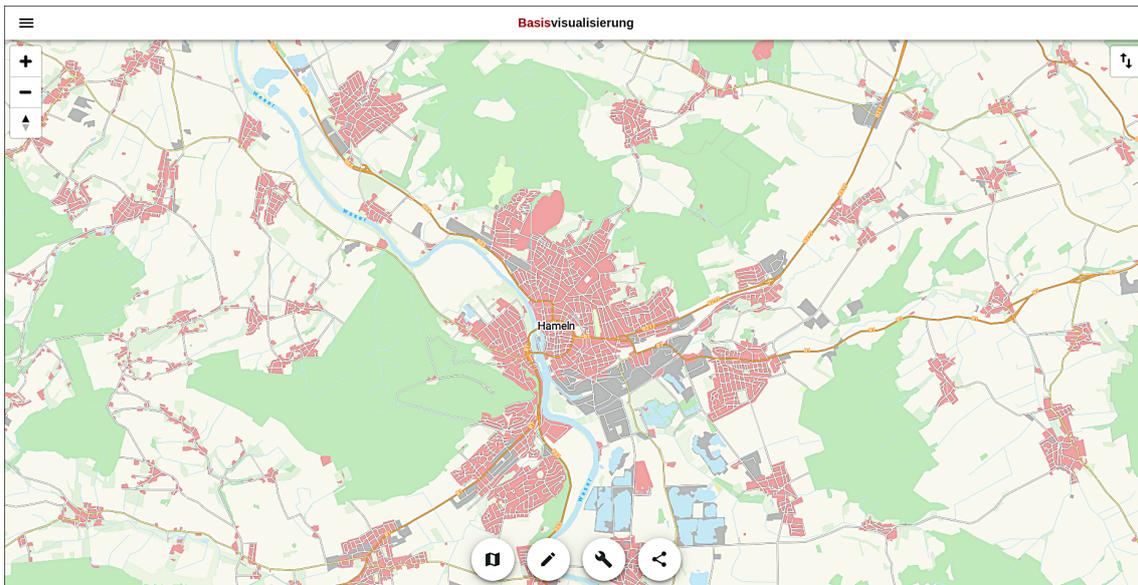


Abbildung 18: Frontend des Webservers. Quelle: Eigene Abbildung.

druck der Applikation vermitteln zu können, ist hier der Startbildschirm abgebildet. Er bietet unterschiedliche Möglichkeiten, die zumeist mithilfe von TypeScript realisiert sind. Die wichtigsten Funktionen werden in folgendem Abschnitt beschrieben.

Hauptmenü

Das Hauptmenü befindet sich in der linken oberen Ecke des Startbildschirms. Es besitzt noch keine Funktion und führt den Nutzer zurück zum Hauptbildschirm. Hier sollen in Zukunft Informationen zum Datenschutz, das Impressum, eine Bewertungsfunktion und Informationen zur Applikation hinterlegt werden.

Kartendarstellung

Die Karte wird in der `mapbox-gl.component`-Datei erzeugt und mithilfe von `MapboxGL` visualisiert. Das Styling der Daten entspricht dabei dem aktuellen Styling des `MapStylingService`, einer Klasse, die das aktuelle Styling der Karte vorhält. Bei einer Änderung²¹ wird das aktuelle Styling angepasst.

²¹ vgl. Kartenfunktionalitäten

Neben der Karte umfasst der Startbildschirm in der rechten und linken oberen Ecke interaktive Elemente, um die Kartendarstellung zu verändern. Mit diesen Elementen ist es für den Nutzer möglich, zu zoomen und die Karte zu neigen. Für die Implementation dieser Funktionalität wurden die Mapboxfunktionen *NavigationControl()* und *PitchControl()* zum Kartenelement hinzugefügt²².

Kartenfunktionalitäten

Am unteren Kartenrand sind vier Knöpfe implementiert, mit denen der Nutzer unterschiedliche Möglichkeiten hat, das Erscheinungsbild der Karte zu verändern.

Mithilfe des ersten Knopfs von links ist es möglich, das Styling der Karte on-the-fly zu ändern. Dazu stehen verschiedene Stylings zur Verfügung, die für unterschiedliche Einsatzzwecke gedacht sind. Die Applikation bietet sieben Styles an, die zum Teil an andere Geoviewer angelehnt sind und zum Teil Eigenentwicklung des LGLN darstellen.

Auf der Programmierenebene wird der Client vom Server durch den Tile-Server mit den notwendigen Vektor-Tiles beliefert. Danach rendert der Client die Daten mithilfe eines MapboxGL-Stylings, welches vom Server geliefert wird. Das aktuelle Styling des MapStylingService wird durch Interaktion mit den dargestellten Styles verändert und das Aussehen der Karte ändert sich. Die folgende Grafik verdeutlicht dabei die beteiligten Komponenten der Applikation: Über

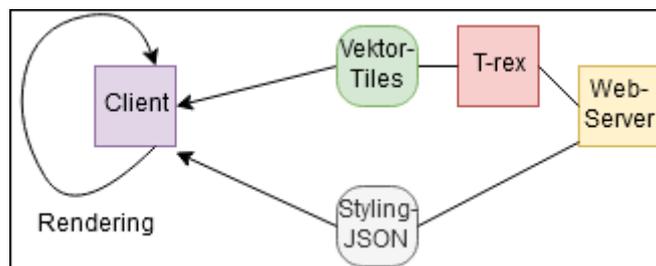


Abbildung 19: Renderingprozess Quelle: Eigene Abbildung.

den zweiten Knopf lassen sich die Layer der Karte verändern. Für die in der Karte dargestellten Klassen Verkehr, Nutzungsflächen, Vegetation, Gewässer und die Schrift ist es so möglich, den Detailgrad zu bestimmen, der in der aktuellen Version von 0 (stark generalisiert) bis 3 (alle Details) aufgeteilt ist.

Um diese Funktion zu ermöglichen, wird die Methode *changeGroupDetailLevel()* verwendet, die aufgerufen wird, wenn sich der Slider einer Klasse verändert. Über die Styling-Datei des momentanen Styles ist das Aussehen je nach aktueller Detailstufe festgelegt.

Neben dieser Funktionalität, die es dem Nutzer erlaubt Einfluss auf vorhandene Styles zu nehmen, ist eine Zusatzfunktion unter diesem Knopf geplant. Sie soll eine vollständige Anpassung jeder einzelnen Klasse der Karte bieten, sodass komplett eigene Kartenstyles erstellt werden können. Diese Funktion befindet sich aber noch im Aufbau und ist noch nicht vollständig implementiert.

²² Zur weiteren Erklärung der Funktionen sei auf die Dokumentation von Mapbox verwiesen: <https://docs.mapbox.com/mapbox-gl-js/api/>

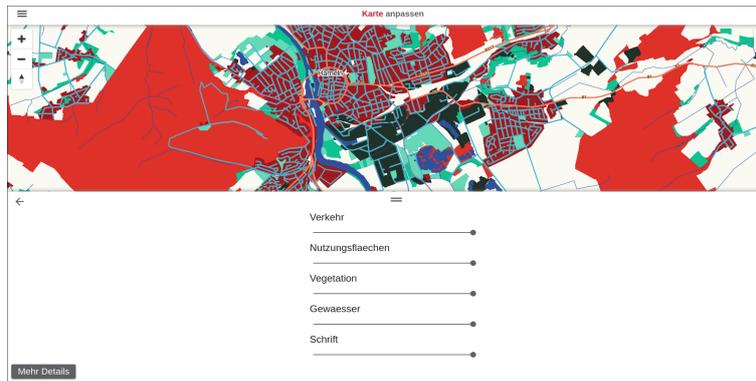


Abbildung 20: Kartengeneralisierung Quelle: Eigene Abbildung.

Der dritte Knopf besitzt noch keine Funktionalität, soll aber Werkzeuge zur Kartenerstellung bereitstellen. Um die eigenen angepassten Kartenstyles zu exportieren ist Knopf vier implementiert. Mit ihm ist es möglich, den aktuellen Style als JSON unter einer bestimmten Uniform Resource Locator (URL) zu veröffentlichen. Momentan ist diese Adresse nur lokal zu erreichen, für den Release ist jedoch ein Server für die Styles der Nutzer geplant.

Der Prototyp ist, wie aus den Kartenfunktionalitäten und dem Menü ersichtlich, noch nicht fertig und befindet sich in der Aufbauphase. Im folgenden Abschnitt wird er evaluiert, um die Möglichkeiten der Visualisierung mithilfe der Vektor-Tiles zu verdeutlichen.

In einem zweiten Schritt wird der Prototyp weiterentwickelt, indem die Nutzbarkeit der Applikation mithilfe von neuen Kartenstyles erhöht wird.

6 Evaluierung

In diesem Kapitel soll der Einsatz von Vektor-Tiles als eine Möglichkeit der Visualisierung von Geobasisdaten bewertet werden. Dazu wird im ersten Schritt die Leistung des Prototypen analysiert und mit einem anderen Geoviewer, der auf Raster-Tiles basiert verglichen.

Die Leistungsbewertung der Applikationen wird dabei mithilfe von Google Lighthouse(vgl. Kapitel 4.8) realisiert. In diesem Kontext werden die Leistung, die Barrierefreiheit, der Einsatz moderner Webstandards und die Sichtbarkeit der Webentwicklungen ermittelt und miteinander verglichen.

Im zweiten Schritt werden die Funktionen analysiert, die der Prototyp für den Nutzer im Vergleich zu etablierten Geoviewern anbietet. Die Vergleichsbasis ist hier der Geolife-Viewer des LGLN.

Testumgebung

Getestet werden beide Webentwicklung auf einem Dell XPS 13 9370. Der Laptop ist dabei mit einem Intel Core i7-8550U ausgestattet, der über 4 Kerne verfügt. Neben dem Prozessor stehen dem System 16 Gigabyte Arbeitsspeicher und eine UHD Graphics 620 Prozessorgrafikkarte zur Verfügung.

Die Testreihe findet unter dem Betriebssystem Ubuntu Linux Version 18.04 statt und nutzt dabei den Browser Google Chrome²³ im Inkognitomodus. Der Prototyp wird dabei in der un-kompilierten Entwicklungsumgebung von Angular getestet.

Testreihe

Um Ergebnisse zu erzielen, die einen Aussagewert für die Bewertung der unterschiedlichen Datenformate liefern, wird der Test für jede Entwicklung 30 mal durchgeführt. Dies soll statistische Schwankungen im Rahmen der Testreihe minimieren und für zuverlässige Aussagen der Evaluierung sorgen.

Neben einer Testreihe unter normalen Bedingungen werden die Webseiten auch mit vierfach reduzierter Leistung des CPU getestet. Dies soll Aussagen über die Leistungsfähigkeit der Applikationen bei älteren Geräten ermöglichen.

Google Lighthouse beurteilt die im Kapitel 4.8 beschriebenen Kriterien mit Punktwerten von 0-100. Drei der vier Kategorien²⁴ werden dabei von Lighthouse in jeder Iteration gleich bewertet, da die Tests auf sogenannten “audits“ basieren. Diese können entweder bestanden oder nicht bestanden werden und beruhen auf der Analyse des Quellcodes.

²³ Version 75.0.3770.100-1.

²⁴ Barrierefreiheit, moderne Webstandards und Sichtbarkeit der Webseite.

Geolife- Viewer

Nachdem der Prototyp in Kapitel 5 ausreichend besprochen wurde, soll hier kurz auf den Geolife- Viewer eingegangen werden, der als klassischer Rasterdaten - Geoviewer in der Evaluierung als Vergleichsobjekt dient.

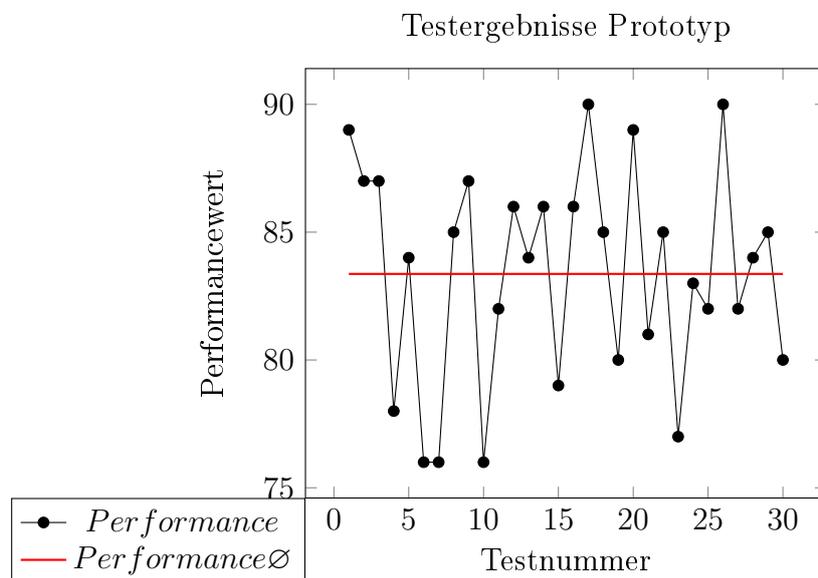
Die Webentwicklung ist dabei auf Freizeitangebote²⁵ spezialisiert und kann diese auf Basis der Geobasisdaten des Landes Niedersachsen visualisieren. Momentan dient die Applikation zudem als Ersatz zum NiedersachsenViewer, einem Geodateninfrastrukturviewer, der zur Zeit nicht vom Land angeboten wird[54].

Testergebnisse Prototyp

Der abgebildete Graph zeigt die Performance des Prototypen in den 30 durchgeführten Tests. Die Messwerte sind dabei als schwarze Punkte im Graphen abgebildet und reichen von 76 bis 90. Die Varianz der Messreihe, die durch die Formel

$$\frac{\sum(x - \bar{x})^2}{(n - 1)}$$

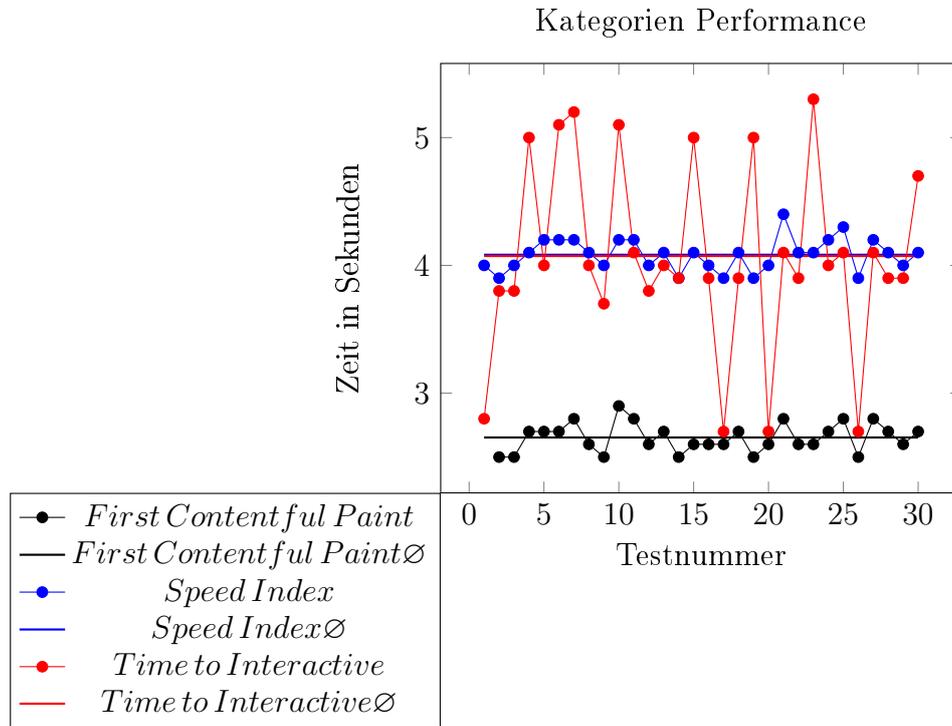
berechnet wird, beträgt $\approx 17,55$. Dieser Wert ist für eine solche Messreihe recht hoch, bestätigt aber den von Google geschätzten Einfluss der Varianz auf das Messergebnis(vgl. Hulce [55] S.3-5). So haben das lokale Netzwerk, die Hardware des Clienten und der Nichtdeterminismus des Browsers und weitere Einfluss auf das Messergebnis und Schwankungen innerhalb der Messungen sind normal.



Aus diesem Grund wurde sich hier für 30 Testläufe entschieden, um den Einfluss der zufällige Schwankung einzelner Messergebnisse auf das Gesamtergebnis zu verringern. Zu diesem Zweck ist die durchschnittliche Leistung(in der Abbildung rot abgebildet) des Prototypen mithilfe des Mittelwerts berechnet worden und beträgt $\approx 83,37$. Für die maximale Leistung des Testlaptops ist dieser Wert nicht optimal, aber für einen Prototypen in diesem Stadium der Entwicklung

²⁵ Angeboten werden beispielsweise Fuß-, Rad- und Wanderwege.

annehmbar. Um einen genaueren Blick auf die Leistung des Prototypen zu werfen, werden die einzelnen Leistungsindikatoren, die für die Gesamtpunktzahl verantwortlich sind abgebildet. Da die Werte für First Contentful Paint und First Meaningful Paint, sowie Time to Interactive und First CPU Idle identisch sind, wird nur eine der beiden Möglichkeiten gezeigt. Eine Sonderstellung unter den Indikatoren nimmt die Estimated Input Latency ein, da sie sich nicht verändert. Sie nimmt in dieser Testreihe konstant den Wert 30 Millisekunden an und wird aufgrund dessen ebenfalls nicht in die Grafik aufgenommen.



Auffällig bei den Indikatoren sind die hohen Schwankungen bei den Messwerten für Time to Interactive und First CPU Idle. Hier beträgt die Varianz $\approx 0,54$, einzelne Messwerte weichen um bis zu 1,3 Punkten vom Mittelwert ab. Die restlichen Indikatoren sind relativ konstant und weisen nur geringe Schwankungen um den Mittelwert auf.

Die dargestellten Einzelbewertung der Kategorien entsprechen im Scoring Modell von Lighthouse ungefähr dem Mittelwert der Performancebewertung(Punktwerte um 83) und weichen nur marginal voneinander ab.

Um die Leistung des Prototypen zu verbessern sind im Report unterschiedliche Möglichkeiten aufgeführt, die in folgendem Abschnitt vorgestellt werden.

1. Enable Text compression - Textbasierte Ressourcen können durch Kompression²⁶ verkleinert werden, um so weniger Netzwerktraffic zu erzeugen. In diesem Beispiel könnten so 9435 Kilobyte gespart werden, was die Seitenladezeit um geschätzte 1,26 Sekunden verbessern würde.

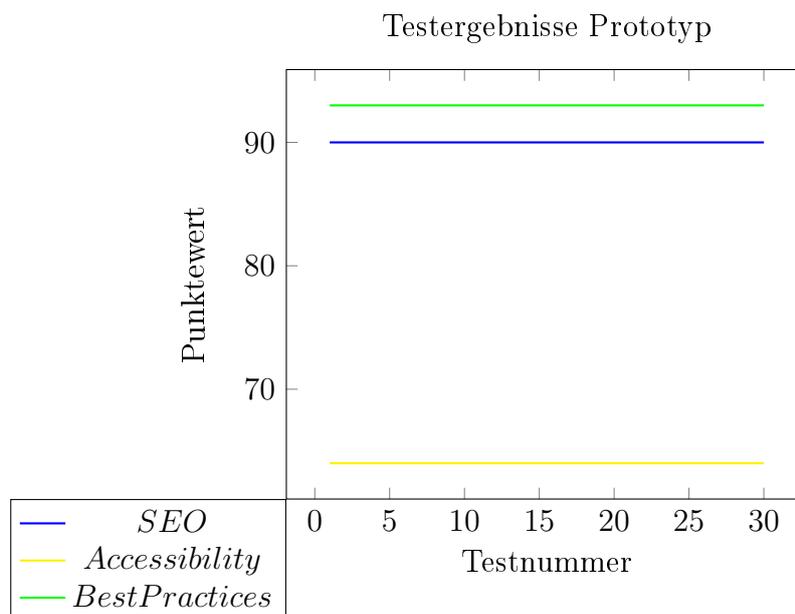
²⁶ Beispielsweise mithilfe von Programmen wie Gzip oder deflate.

2. Minify JavaScript - JavaScript-Dateien können durch Minify um ungenutzten Code oder redundante Informationen verringert werden. Im Prototypen entspricht die geschätzte Einsparung 3428 KB und ca. 0,46 Sekunden Ladezeit.

Neben diesen Verbesserungen werden weitere Probleme der Webentwicklung genannt, die durch Eingriff des Programmierers verbessert werden können. Probleme beim Prototypen sind:

1. Aufgrund der Vektor-Tiles wird das Netzwerk stark beansprucht. Beim Testlauf werden 11,739 KB verschickt, was zu langen Ladezeiten führen kann.
2. Das Cachen der Daten ist für die Anwendung nicht optimiert. Für eine Veröffentlichung sollte die Applikation eine effiziente Cachestrategie besitzen.
3. Die Programmierung der Typescript-Dateien des Projekts sind ineffizient, sodass die Ladezeit der Seite verlängert wird.

Die Bewertung von Barrierefreiheit, Einsatz moderner Webstandards und Sichtbarkeit der Webentwicklung ist bei jedem Testlauf gleich. Der folgende Graph zeigt die Messergebnisse für diese Kategorien. Die abgebildeten Werte gelten dabei sowohl für die Testreihe mit als auch ohne Verlangsamung der CPU.

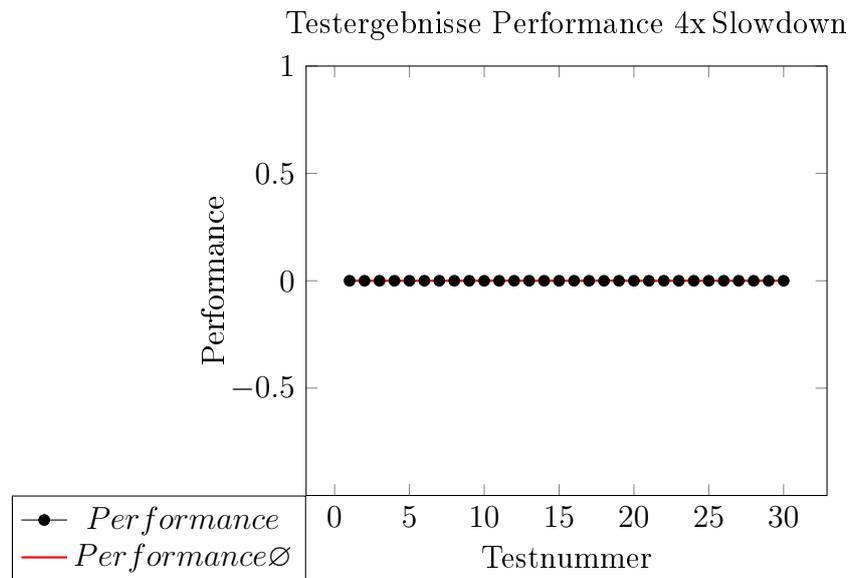


Für die Sichtbarkeit der Webentwicklung und den Einsatz moderner Webstandards werden Punktwerte von 93 bzw. 90 erzielt. Lighthouse bewertet Punktzahlen ≤ 90 als sehr gutes Ergebnis, im Bezug auf die beiden Kategorien gibt es für den Prototypen wenig Verbesserungspotential.

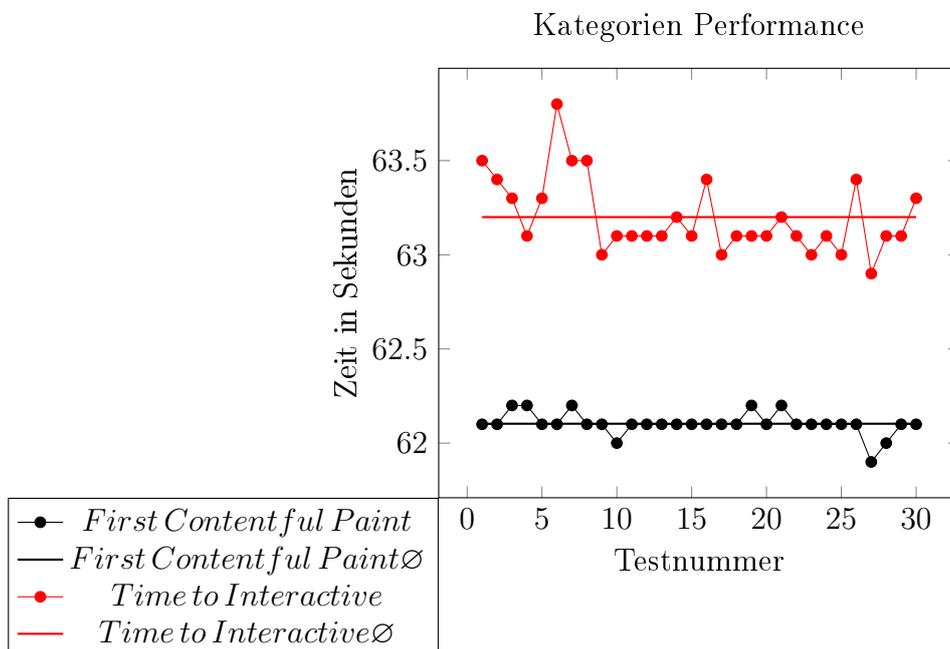
Um 100 Punkte zu erzielen, sollte jede Ressource des Projekts HTTP/2 benutzen und das Dokument mit einer Metabeschreibung ausgestattet werden.

Die Barrierefreiheit der Webseite für Menschen mit Einschränkungen wird im Test mit einer Punktzahl von 64 Punkten durchschnittlich bewertet.

Verbessert werden können hier die Navigation des Prototypen, sowie das Buttondesign. Die Anforderungen der Tests auf Bypass und Button name, die in Kapitel 4.8 genannt werden, werden vom Projekt nicht erfüllt und bedürfen Änderungen. Die Testergebnisse des Prototypen ohne CPU Verlangsamung zeigen, dass die Applikation auf modernen Endgeräten eine gute Performance aufweist. Einige Anpassungen im Bereich moderne Webstandards und Sichtbarkeit der Webentwicklung müssen getätigt werden, um die volle Punktzahl im Test von Google Lighthouse zu erzielen. Im Bereich Barrierefreiheit sind noch viele Möglichkeiten zur Verbesserung. Im zweiten Schritt wird der Prototyp nun unter vierfacher Verlangsamung der CPU getestet. Diese Testumgebung simuliert die Leistung von Low-End-Rechnern oder Smartphones, auf denen die Webseite ebenso laufen soll.



Das Testergebnis für Performance zeigt, dass sich der Prototyp im derzeitigen Status nicht für den Einsatz auf langsamen Rechnern oder Smartphones eignet. Der Performancewert liegt bei jedem Test bei 0, wichtige Inhalte der Seite werden nur sehr langsam dargestellt.

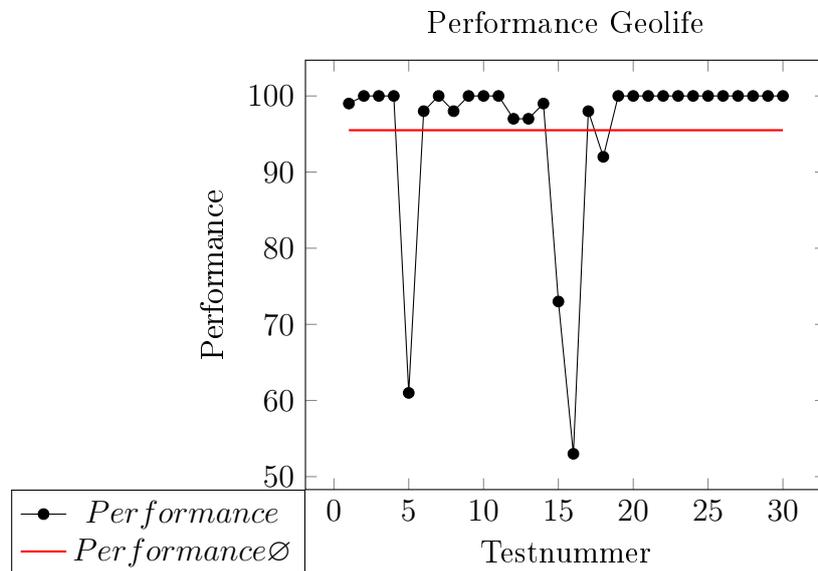


Diese Bewertungen lässt sich durch die einzelnen Kategorien bestätigen. Die im Graph dargestellten Rechenzeiten, die benötigt werden um Inhalte darzustellen oder eine interaktive Oberfläche zu bieten, sind für Endnutzer nicht akzeptabel²⁷. Der Übersicht halber wird der Wert für Estimated Input Latency nicht in der Grafik angezeigt. Sie beträgt durchschnittlich 1380 ms und bewegt sich von 1260 - 1450 ms. Da die Rechenzeiten für First Contentful Paint, Speed Index, First Meaningful Paint und First CPU Idle identisch sind, ist nur eine der Messreihen im Graph abgebildet. Die Gründe dieser langsamen Berechnung werden im Vergleich von Geolifeviewer und Prototyp diskutiert.

Zur Verbesserung der Webseite schlägt das Testprogramm wie beim Test mit voller CPU-Leistung Textkompression und JavaScript Minimierung vor. Die geschätzten Zeitgewinne durch diese Operationen sind im Vergleich zur ersten Testreihe aber bedeutend. Mithilfe der Textkompression können geschätzte 47,88 Sekunden eingespart werden. Zudem bringt die Reduktion von JavaScript - Anteilen der Entwicklung ca. 17,26 Sekunden Zeitersparnis. Damit können zusammen fast 65 Sekunden Ladezeit eingespart werden. Da beide Techniken aber JavaScript - Dateien verkleinern, sind diese Zeitgewinne nicht additiv, sondern in der Praxis etwas geringer.

Testergebnisse Geolife

Um einen Vergleich zu ermöglichen wird der Geolife - Viewer des LGLN mithilfe von Lighthouse getestet. Dabei gelten die gleichen Voraussetzungen, wie beim Test des Prototypen.

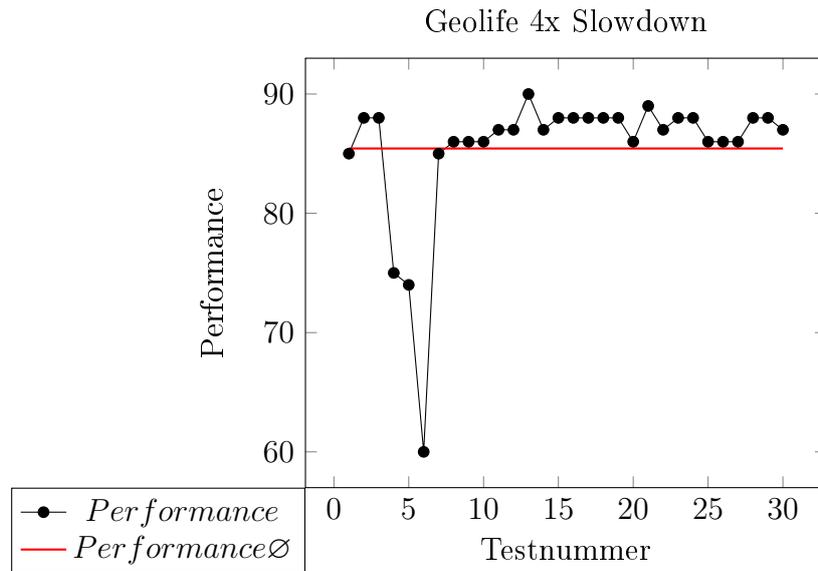


Der Graph zeigt die Performance der Webseite bei voller CPU - Leistung. Die Varianz der Messreihe beträgt dabei $\approx 34,73$. Der Grund dafür sind vier Testläufe, die eine deutlich geringere Performance, wie der Rest der Testreihe messen. Die durchschnittliche Leistung der Webentwicklung liegt dabei bei 95,5, was in Lighthouse einem sehr gutem Wert entspricht.

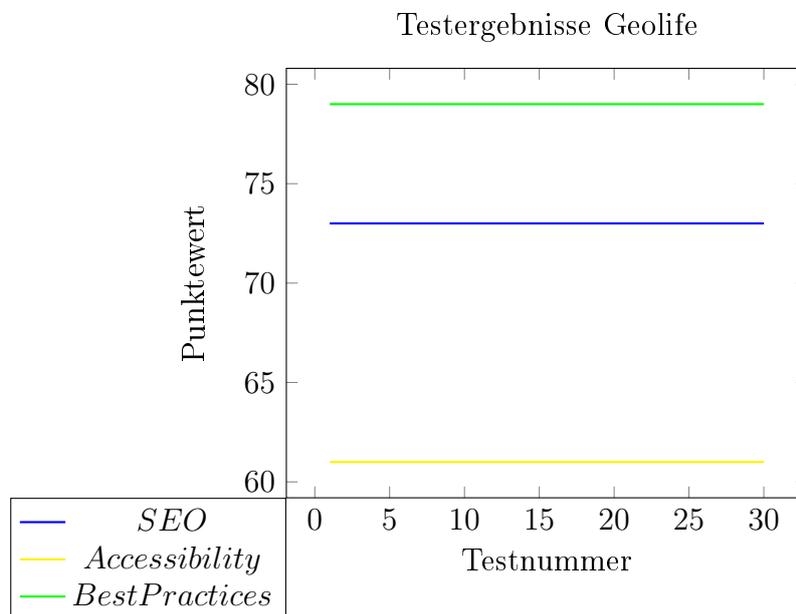
In den Unterkategorien schneidet der Geolife - Viewer dementsprechend ebenfalls besser als der

²⁷ Im Durchschnitt $\approx 62,1$ Sekunden für First Contentful Paint und $\approx 63,2$ Sekunden für Time to Interactive.

Prototyp ab^{28 29}. Auch unter vierfacher CPU Verlangsamung liegt die Performance im Durchschnitt bei $\approx 85,43$ und damit höher als der Prototyp.



Auffällig ist die hohe Varianz der Daten, der Geolife-Viewer weist in Einzelfällen deutliche Schwankungen in seiner Leistung auf. Anders als der Prototyp kann die Webseite aber auch mit weniger CPU-Leistung zuverlässig und schnell dargestellt werden.



Die Werte der übrigens Test von Lighthouse zeigen, dass der Prototyp in Bezug auf Barrierefreiheit, moderner Webstandards und Sichtbarkeit besser als Geolife abschneidet. Die Bewertungen des Geoviewers³⁰ liegen teilweise deutlich unter dem Ergebnis des Prototypen.

²⁸ Mittelwerte der Tests: First Contentful Paint: $\approx 0,54$, Speedindex: $\approx 2,86$, Time to Interactive: $\approx 3,51$, First Meaningful Paint: $\approx 0,54$, First CPU Idle: $\approx 1,23$, Estimated Input Latency: $\approx 20,66$.

²⁹ Die Verbesserungsmöglichkeiten vom Geolife-Viewer werden hier nicht behandelt, da sie nicht Thema der Masterarbeit sind.

³⁰ Barrierefreiheit: 61, Moderne Webstandards: 79 und Sichtbarkeit der Webseite: 73

Vergleich und Analyse der Ergebnisse

Die Ergebnisse der Evaluation auf technischer Ebene zeigen, dass einige Anpassungen notwendig sind, um den Prototypen für Endnutzer zu veröffentlichen. Die Applikation liefert im Rahmen der Testreihe mithilfe einer modernen CPU akzeptable Ergebnisse (vgl. Seite 36 - 40). Da die Webseite auch auf Lowend - Rechnern laufen soll, wird der Prototyp unter Reduktion der CPU - Leistung getestet. Die daraus resultierenden Testergebnisse zeigen, dass der Prototyp für diesen Einsatz noch nicht ausgelegt ist. Im Vergleich dazu liefert der Geolife - Viewer in beiden Testumgebungen gute Ergebnisse im Lighthouse - Test (vgl. Seite 40f.). Um den Prototypen für eine Veröffentlichung bereit zu machen, sind dazu mehrere Schritte notwendig, die in folgendem genannt werden und als Handlungsempfehlung für das LGLN verstanden werden können:

1. Textkompression der Javascript - Daten - Insbesondere unter Reduktion der CPU - Leistung sorgen große, textbasierte Ressourcen für langsame Lade - und Reaktionszeiten der Webseite. Diese müssen mithilfe der verfügbaren Technologien komprimiert werden, um die Größe der Netzwerkdaten zu reduzieren.
2. Cachepolitik - Da sich die Applikation noch in der Testphase befindet, ist noch keine Cachestrategie vorhanden. Dementsprechend werden Ressourcen, die für die Webseite notwendig sind nicht zwischengespeichert und müssen bei jedem Aufruf neu geladen werden. Dies sorgt ebenfalls für höhere Netzwerkauslastung, die zu langen Ladezeiten führt.
3. Optimierung der Vektor - Tiles - Die für den Prototypen genutzten Vektor - Tiles sind in ihrer Größe und Abdeckung noch nicht für den Einsatz optimiert. Dieser Zustand führt dazu, dass die für gewöhnlich kleinen Tiles (vgl. 4.3) größer sind als notwendig und für längere Ladezeiten sorgen. Für den Release der Webseite sollte die Landesbehörde die Tiles an den Anwendungszweck anpassen und so Datentransfer einsparen.

Der erste Punkt der Analyse ist dabei unter Vorbehalt zu betrachten. Da der Prototyp im Entwicklungsmodus getestet wurde, sind die Ergebnisse der Performancetests in Zukunft deutlich besser. Eine Kompilierung, die vor jeder Installation auf einem Server standardmäßig erfolgt, verringert die Größe des Frontends auf ca. 2MB. Dieser Schritt hätte maßgeblichen Einfluss auf die Performance, vor allem im Low - End - Test der Applikation.

Da sich beide Geoviewer auf technisch unterschiedlichen Niveaus³¹ befinden, liefert der erste Teil der Evaluation der Leistung keine eindeutigen Pro - oder Kontraargumente für Vektor - Tiles.

Im Gegenzug zeigt die Evaluation aber, dass die Applikation auf moderne Webstandards zurückgreift und eine hohe Sichtbarkeit im Web besitzt.

Die Barrierefreiheit der Webseite wird durch die Tests aber deutlich schlechter bewertet und benötigt Verbesserung. Aus diesem Grund wird im Zuge der Weiterentwicklung des Prototypen eine Möglichkeit ausgearbeitet, die Applikation auch für Menschen mit eingeschränktem

³¹ Optimierte Webentwicklung gegenüber einem Prototypen in der Entwicklung.

Sehvermögen benutzbar zu machen. Dazu werden unterschiedliche Mapbox-Styles entwickelt, die auf verschiedene Typen von Farbenfehlsichtigkeit zugeschnitten sind.

Im zweiten Schritt der Evaluierung werden die Funktionen der Geoviewer miteinander verglichen. Dieser Vergleich dient dazu einschätzen zu können, welches Tileformat sich besser für diesen Einsatzzweck eignet.

Evaluation Funktionalität der Geoviewer

Da in der Wissenschaft zur Zeit noch kein Bewertungsrahmen für Geoviewer existiert, wird der Prototyp im Bereich Funktionen anhand eines Vergleiches bewertet. Als Vergleichsobjekt dient der in der Arbeit vorgestellte Geolife-Viewer, der auf Raster-Tiles basiert.

Dabei sollen die Kategorien Kartendarstellung, Geoverarbeitung, Kartenstyling und sonstige Angebote bewertet werden. Sie spiegeln generelle Funktionen eines Geoviewers wider und sind in vielen unterschiedlichen Geoviewern³² präsent.

Kartendarstellung

Die Kartendarstellung im Prototypen ist mithilfe von Vektor-Tiles implementiert. Dieser Sachverhalt impliziert die Vor- und Nachteile, die mit der Verwendung von Vektor-Tiles einhergehen. Im Vergleich dazu benutzt Geolife Raster-Tiles, deren Verwendung ebenfalls zu unterschiedliche Vor- und Nachteile(vgl. Kapitel 4.3) führt. In der praktischen Umsetzung kann der Nutzer dies beispielsweise am Format und an der Größe der Kacheln erkennen, die im Browser visualisiert werden³³.

Das Tileformat ist dabei die Grundlage für die unterschiedlichen Möglichkeiten, die ein Geoviewer dem Nutzer bieten kann. Es beeinflusst direkt die Funktionen der Webseite, sodass sie nicht losgelöst von der Tilewahl betrachtet werden können.

Der Prototyp visualisiert ATKIS-Daten und umfasst dabei die Bodenbedeckung, Gebäude und Straßen Niedersachsens. In Zukunft sollen weitere Fachdaten und 3D Gebäudedaten in die Darstellung aufgenommen werden.

Im Vergleich dazu ist das Angebot des Geolife-Viewers zur Zeit größer. Neben den Features des Prototypen werden zudem Points of Interest angezeigt, die in der Karte mithilfe des ATKIS Signaturenkatalogs visualisiert werden. Zudem ist es möglich, die Hintergrundkarte³⁴ zu ändern und weitere Informationslayer, wie beispielsweise Schutzgebiete oder Verwaltungsgrenzen hinzuzufügen.

Der Vergleich macht deutlich, dass beide Geoviewer auf einer unterschiedlichen Basis fußen, die unterschiedliche Funktionen ermöglicht.

Der aktuelle Stand zeigt, dass der Prototyp noch vergleichsweise wenig Kartenmaterial anbietet.

³² Beispiele hierfür sind unter anderem der Bundesanstalt für Geowissenschaften und Rohstoffe Geoviewer, das Geoportal Nordrhein-Westfalen oder der IÖR-Monitor.

³³ Der Browser lädt im Prototypen Google Protobufs, während er in der Geolife-Anwendung PNG-Kacheln benutzt.

³⁴ Kartenmaterial wie Luftbilder oder historische Karten.

Da das Land aber Zugang zu den notwendigen Geobasisdaten besitzt, ist eine Implementation im Verlauf des Prototyping aber vergleichsweise einfach und wird in Zukunft forciert.

Kartenstyling

Zum Visualisierungsprozess(vgl. Kapitel 4.1) gehört auch das Styling der zugrunde liegenden Datenbasis. Der Prototyp bietet dem Nutzer unterschiedliche Kartenstyles, mit denen das Aussehen der Karte on-the-fly verändert werden kann. Jeder Style beruht dabei auf der Mapbox Style Specification(vgl. Kapitel 4.4.3) und liegt als JSON-Datei vor. Da die Vektor-Tiles beim Clienten verfügbar sind, können die Symbole und Label der Karte ohne Erhöhung der Bandbreite skaliert werden.

Das Aussehen der einzelnen Strukturelemente, wie Straßen, Schriftzüge oder Gebäude basiert dabei auf keiner Spezifikation, sondern wurde vom LGLN neu entwickelt.

Im Gegensatz dazu basiert die Darstellung des Geolife-Viewers auf den Signaturenkatalogen der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland(AdV)³⁵. Für jede Auflösungen und jede Objektart sind festgelegte Signaturen zu verwenden, die das Styling der Karte bestimmen.

Die Gegenüberstellung beider Stylingmöglichkeiten zeigt, dass Vektor-Tiles Vorteile gegenüber Raster-Tiles besitzen. So beeinträchtigen Kartenelemente, wie beispielsweise Schriftzüge oder Points of Interest die Bandbreite nicht, da sie beliebig skaliert werden können. Bei Raster-Tiles muss das jeweilige Kartenelement für die einzelnen Maßstäbe angepasst werden, was zu erhöhter Bandbreite führt.

Eine weitere Beobachtung, die sich aus dem Vergleich beider Geoviewer ergibt, ist der Paradigmenwechsel, der mit der Entwicklung des Prototypen beim LGLN eingesetzt hat.

Etablierte Geoviewer des Landes, hier beispielhaft vertreten durch den Geolife-Viewer, setzen auf Signaturkataloge zum Styling der Kartenelemente der DTK. Durch diese Wahl ist die Interoperabilität der Daten innerhalb der Arbeit auf Ebene unterschiedlicher Landesbehörden gewährleistet.

Der Prototyp verfolgt aber einen eigenen Ansatz. Hier gibt es verschiedene Styles, die vom Standard des Signaturkatalogs abweichen. Daraus ergeben sich auf der einen Seite Freiheiten für die Entwicklung neuer eigener Signaturen, aber auf der anderen Seite auch Probleme. Durch eigene Signaturen ist die Interoperabilität der Daten beispielsweise auf Bundesebene nicht mehr gewährleistet. Verschiedene Landesbehörden besitzen nach diesem System unterschiedliche Signaturen für die gleiche Objektart.

Da das Land Niedersachsen aber als einer der ersten Landesbehörden das Thema Visualisierung von Geobasisdaten mithilfe von Vektor-Tiles umsetzt, kann der Prototyp Signalwirkung für die anderen Landesbehörden haben. Wenn sich die Basisvisualisierung mit Vektor-Tiles auf

³⁵ Für die DTK 10 bis DTK 100 bietet die AdV Signaturenkataloge für Geobasisdaten an.

Landesebene durchsetzt, können andere Länder ihre Signaturen für Objektarten anpassen und so die Interoperabilität wiederherstellen.

Geoverarbeitung

Geoviewer bieten in der Regel keine ausgeprägten Geoverarbeitungsschritte an, da es auf Basis von Raster-Tiles verhältnismäßig schwierig ist, diese zu visualisieren. Ausschlaggebend für diesen Sachverhalt ist das Prerendering. Tiles werden für jede Zoomstufe vorgehalten und bei Bedarf abgerufen. Im Vergleich dazu werden Vektordaten on-the-fly erzeugt, sodass weitere Objekte in diesem Prozess hinzugefügt werden können. Zudem beinhaltet die Vektor-Tiles auch die Objektgeometrien (vgl. Kapitel 4.3), die für die Geoverarbeitung benötigt werden.

Vor diesem Hintergrund ist dieser Vergleich beider Geoviewer zu sehen, Geolife bietet als einziges Werkzeug der Geoverarbeitung das Messen von Strecken oder Flächen an. Weitere Werkzeuge sind aufgrund des Ausgangsformats der Daten nicht implementiert und würden die Leistung der Applikation deutlich reduzieren.

Der Prototyp bietet bisher noch keine Möglichkeit der Geoverarbeitung. In Zukunft können aber einfache Werkzeuge entwickelt werden, da die Objektgeometrien beim Clienten vorliegen. So bietet das Tile-Format die Möglichkeit für erfahrene Benutzer einfache Geoverarbeitung im Browser zu tätigen. Für eine Behörde, die beispielsweise auf kostenpflichtige Software zurückgreifen muss, kann dies zu Kosteneinsparungen führen.

Sonstige Angebote

Aufgrund der Ausrichtung des Geolife-Viewers besteht im Geoviewer die Möglichkeit, sich das Freizeitangebot Niedersachsens visualisieren zu lassen. Die Visualisierung der ausgewählten Tour wird dabei über ein Overlay und ein Informationsfenster realisiert, welches die grundlegenden Informationen zur Freizeitaktivität beinhaltet. Neben der Ausrichtung auf Freizeit ist es möglich nach gastronomischen Einrichtungen, Unterkünften oder nach Kultur zu suchen. Die dafür implementierte Umkreissuche ist über das Startmenü zu erreichen und zeigt die Suchergebnisse auf der Karte an³⁶.

Für einige Layer der Kartendarstellung sind weiterführende Informationen hinterlegt, die der Benutzer abfragen kann. Dazu ist eine Informationsabfrage implementiert, die neben den Informationen noch die Koordinaten, das Bezugssystem und die Höhe über Normalnull in einem Informationsfenster darstellt.

Die so erstellte Karte kann vom Benutzer zu jedem Zeitpunkt über eine Druckfunktion ausgegeben werden.

Im Gegensatz dazu liefert der Prototyp andere Funktionen, die eng mit der Wahl des Datentypen zusammenhängen.

Momentan ist es möglich den aktuellen Style der Karte als JSON zu exportieren, um ihn persistent speichern zu können. Gleichzeitig liefert die Webseite die Option, eine solche Basiskarte zu

³⁶ Die Form der Visualisierung gleicht dabei der des Freizeitangebots.

laden. Dies sind vorbereitende Funktionen, die dem Nutzer die Gelegenheit auf ein interaktives Kartenerlebnis bieten. In Zukunft soll jedes Element der Karte in Bezug auf die Farbgebung veränderbar sein. Der Benutzer kann dann mithilfe eines Informationsfensters die Farbe verändern und sich so eine eigene Karte auf Basis des vorhandenen Kartenmaterials erstellen. Die dafür notwendige Funktion wird derzeit im LGLN entwickelt und ist im derzeitigen Stand noch nicht komplett implementiert. Darauf aufbauend soll eine Druckfunktion entwickelt werden, die die Basiskarte dann für den Benutzer physisch verfügbar macht.

Die Eventualität das Styling on-the-fly zu verändern, spielt in dieser Funktionskette eine entscheidende Rolle: Ohne diese Eigenschaft der Vektortiles ist es nicht möglich, eine solche Funktionalität anzubieten.

Mithilfe dieser Veränderung vereinigt der Prototyp in naher Zukunft die Funktionalitäten eines Geoviewers mit Web-Mapping 2.0 Elementen³⁷.

Neben dieser Funktionalität sind weitere Veränderungen geplant, die dem Benutzer ein umfassenderes Erlebnis der Daten gewähren sollen³⁸:

1. Navigation - Implementation einer Navigation mithilfe des OpenRouteService der Universität Heidelberg.
2. 3D-Modell - Visualisierung von 3D-Daten aus Punktwolken Cesium.

Die Gegenüberstellung der sonstigen Angebote zeigt, dass sich der Prototyp noch in einer Anphangsphase befindet, in der die Grundlage für komplexere Funktionalitäten gelegt werden. Die zukünftige Umsetzung ist dabei stark an die Datengrundlage Vektor-Tiles angepasst und bedient sich an deren Vorteilen. Besonders im Bereich Styling sind diese Vorteile so groß, dass der Einsatz dieser Grundlage notwendig ist.

Die geplanten Funktionalitäten des Prototypen erweitern den Geoviewer in der Form, dass er, wenn alle Neuerungen, wie beispielsweise die Implementation einer Visualisierung von 3D-Daten, erfolgt sind, einem digitalen Globus für Niedersachsen ähnelt(vgl. Seite 9).

Die Evaluierung hat insgesamt gezeigt, dass sich die im Prototyp verwendeten Vektor-Tiles für die Entwicklung eines Geoviewers besser eignen als Raster-Tiles. Die Vorteile sind insbesondere im Vergleich der Funktionalitäten zu sehen, im Bereich Styling und der Erstellung von Karten ist die on-the-fly-Prozessierung dem Prerendering vorzuziehen. Aufgrund des technologischen Unterschieds der Geoviewer konnten einige Vorteile der Vektor-Tiles nicht in der Praxis bestätigt werden(vgl. Seite 21 Vorteile Vektor-Tiles 4.). Dafür sind drei Handlungsempfehlungen für das LGLN entstanden, welche zu Performancegewinnen im Bereich Lade- und

³⁷ Web-Mapping bezeichnet dabei“die Erstellung von Karten im Internet mit Hilfe von Web-2.0-Technologien“(de Lange[13]S.273).

³⁸ Um unnötige Längen zu vermeiden hier nur die wichtigsten Ideen.

Reaktionszeit der Webseite führen. Die darin genannten Anpassungen des Quellcodes sind vor allem vor Veröffentlichung der Smartphoneanwendung durchzuführen.

7 Weiterentwicklung des Prototypen

Die Weiterentwicklung des Prototypen basiert auf den Ergebnissen der zuvor durchgeführten Evaluation. Das Ergebnis zur Erreichbarkeit der Webseite ist mit 64 Punkten im Vergleich zu anderen Kriterien als schlecht zu bewerten. Hier setzt die Weiterentwicklung an und versucht das Styling des Prototypen an geltende Standards anzupassen. Dies ist insbesondere relevant, da öffentliche Stellen die WCAG bei neuen Webseiten und mobilen Anwendungen ab spätestens 2021 zwingend berücksichtigen müssen. Die Entwicklung ist demnach notwendig, sollte der Prototyp die Testphase verlassen und veröffentlicht werden.

Die Weiterentwicklung ist dabei in zwei Schritte aufgeteilt. Im ersten Schritt wird die Neuentwicklung beschrieben und die zugrundeliegenden Konzepte erklärt. Im zweiten Schritt wird die Wahrnehmung jedes Styles unter Normalbedingungen und unter Einfluss der jeweiligen Krankheit getestet und miteinander verglichen.

7.1 Styleentwicklung auf Basis von WCAG

Der Fokus der Styleentwicklung liegt auf drei Styles für die in Kapitel 4.9 vorgestellten Ausprägungen der Farbenfehlsichtigkeit Protanopie, Deutanopie und Tritanopie. Die Seheinschränkungen gehen mit unterschiedlichen Farbschwächen einher, die beim Styling der Kartenelemente berücksichtigt werden müssen. Zudem soll laut WCAG der Kontrastunterschied zwischen angrenzenden grafischen Objekten wie beispielsweise Landebedeckungen mindestens 3:1 betragen.

Berechnung des Kontrast

Die Berechnung des Kontrast von zwei Farben ist von WCAG festgelegt und basiert auf Arbeiten der International Electrotechnical Commission(IEC) und International Organization for Standardization(ISO) Standards³⁹. Die Formel um den Kontrast zu ermitteln lautet:

$$(L1 + 0,05)/(L2 + 0,05)$$

L1 bezeichnet dabei die relative Lumineszenz der helleren Farbe, L2 die relative Lumineszenz der dunkleren Farbe.

Diese werden aus den sRGB Farbwerten der beteiligten Farben nach folgender Formel errechnet:

$$L = 0.2126 * R + 0.7152 * G + 0.0722 * B$$

L bezeichnet in dieser Formel die relative Lumineszenz, R, G und B müssen aus den rot-, grün- und Blauwerten der Farbe berechnet werden. Für den RGB-Farbraum sind dafür alle Werte

³⁹ IEC/4WD 61966-2-1 IEC-4WD und ISO-9241-3.

in den sRGB Raum zu überführen. Dazu teilt man den 8Bit Farbanteil durch 255:

$$XsRGB = X8bit/255$$

Der letzte Schritt der Berechnung ist dann die Überführung der sRGB Werte in R, G und B nach folgendem Ausdruck[56]:

$$f(x) = \begin{cases} X = XsRGB/12,92 & \text{if } XsRGB \leq 0,03928 \\ X = ((XsRGB + 0,055)/1,055)^{2,4} & \text{otherwise} \end{cases}$$

Um die Kontrastberechnung zu veranschaulichen, soll hier ein Beispiel für die Berechnung des Kontrast zwischen Schwarz (RGB(0,0,0)) und Weiß (RGB(255,255,255)) gegeben werden. Das Beispiel illustriert zudem auch den maximalen Kontrast im RGB-Farbraum.

$$RsRGB_{\text{schwarz}} = 0/255 = 0$$

$$GsRGB_{\text{schwarz}} = 0/255 = 0$$

$$BsRGB_{\text{schwarz}} = 0/255 = 0$$

$$RsRGB_{\text{schwarz}} \leq 0,03928 \Rightarrow R = 0/12,92 = 0$$

$$GsRGB_{\text{schwarz}} \leq 0,03928 \Rightarrow G = 0/12,92 = 0$$

$$BsRGB_{\text{schwarz}} \leq 0,03928 \Rightarrow B = 0/12,92 = 0$$

$$L_{\text{schwarz}} = 0,2126 * 0 + 0,7152 * 0 + 0,0722 * 0 = 0$$

$$RsRGB_{\text{weiß}} = 255/255 = 1$$

$$GsRGB_{\text{weiß}} = 255/255 = 1$$

$$BsRGB_{\text{weiß}} = 255/255 = 1$$

$$RsRGB_{\text{weiß}} > 0,03928 \Rightarrow ((1+0,055)/1,055)^{2,4} = 1$$

$$GsRGB_{\text{weiß}} > 0,03928 \Rightarrow ((1+0,055)/1,055)^{2,4} = 1$$

$$BsRGB_{\text{weiß}} > 0,03928 \Rightarrow ((1+0,055)/1,055)^{2,4} = 1$$

$$L_{\text{weiß}} = 0,2126 * 1 + 0,7152 * 1 + 0,0722 * 1 = 1$$

$$\text{Kontrast}_{\text{weiß-schwarz}} = (1+0,05)/(0+0,05) = 21$$

Mithilfe dieser Berechnung ist es möglich für eine beliebige Farbkombination zu ermitteln, ob sie dem WCAG entspricht oder nicht. Sie wird in der Arbeit dazu genutzt, um für die Kartenelemente sicherzustellen, dass sie einen ausreichenden Kontrast bieten. Zudem sollten für die Entwicklung der Styles Farbkombination berücksichtigt werden, die von dem jeweiligen Krankheitstyp nicht einwandfrei zu unterscheiden sind. Bei Deuteranopen führen die Farben Rot, Gelb, Braun und Grün zur Verwechslung. Protanopen besitzen die gleiche Schwäche, zudem bereitet die Unterscheidung zwischen Dunkelrot und Schwarz Probleme.

Tritanopen verwechseln Blau und Grün, sowie Gelbgrün und Grau(vgl. Welsch & Liebmann[49] S.269f.).

Kartenvoraussetzungen

Die derzeit implementierten Styles des Prototypen umfassen unterschiedliche Kartenelemente, die direkt aus den Geobasisdaten des Landes abgeleitet sind. Für die neu entwickelten Styles wurden die Daten auf 15 Informationsklassen und deren Textbeschreibung reduziert. Dieser Schritt ist notwendig, da der RGB-Farbraum nur eine gewisse Zahl an Klassen erlaubt, die die notwendigen Kontrastbedingungen erfüllen.

Die eigentliche Farbwahl der Styles ist dabei von im Internet verfügbaren Farbpaletten inspiriert. So liefert die in folgendem Abschnitt abgebildete Farbpalette Beispiele für unterscheidbare Farben für die unterschiedlichen Krankheitsbilder.

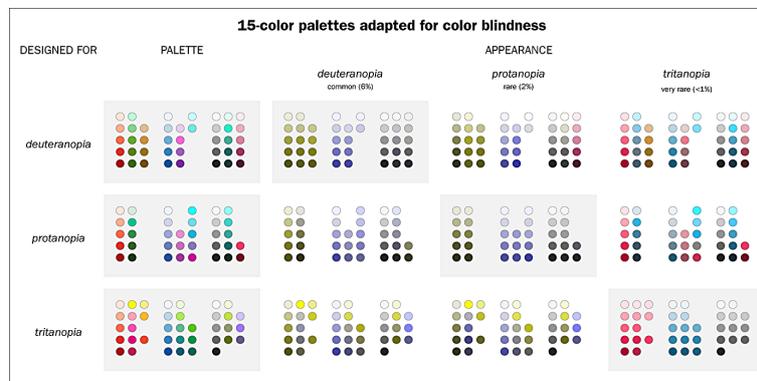


Abbildung 21: Farbpalette für Farbfeldsichtige Quelle: verändert nach Krzywinski[57].

Für jede Farbe in der Abbildung wird die relative Lumineszenz berechnet und die Farbkombination ausgewählt, die für das jeweilige Krankheitsbild möglichst keine Problemfarbe aufweist und einen hohen Kontrast zwischen den Farben gewährleistet. In der Praxis wird dabei die Kontrastbedingungen vom WCAG teilweise verletzt, da Kontraste unter 3:1 in der Farbwahl vorkommen.

Protanopiestyle

Die im Rahmen der Weiterentwicklung erstellte Farbenpalette zur Protanopie umfasst 15 Landebedeckungsklassen, die entsprechend der Grafik gefärbt sind.

Dabei berücksichtigt die Farbwahl einen möglichst hohen Kontrast zwischen den Farben. Zudem werden bestimmte Farbkombinationen gesondert behandelt (vgl. Seite 50), da sie zu Unterscheidungsproblemen führen können.

Ein Beispiel für eine solche Farbkombination ist für Protanopen Grün und Rot. Für Siedlungsflächen und Ortlagen wurde ein besonders dunkles Rot benutzt, damit sich die Farbe trotz dieser Problematik von den Grün- und den Naturschutzflächen der Karte abhebt. Dieser Sachverhalt

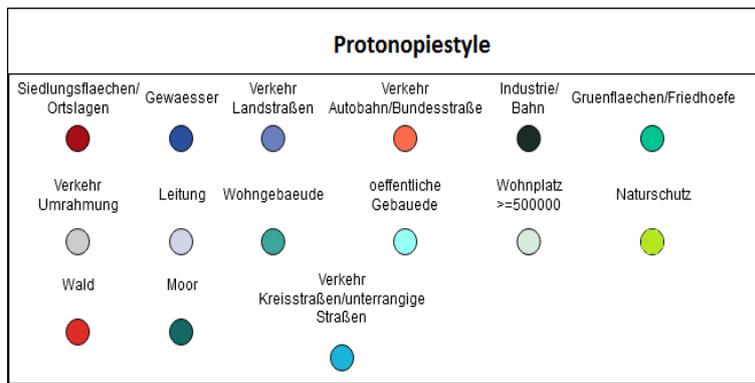


Abbildung 22: Farbpalette für Protonopen Quelle: Eigene Abbildung.

wird nochmal in Kapitel 7.2 untersucht, indem der Style unter unterschiedlichen Sehverhältnissen getestet wird. Zudem ist bei Berechnung der Kontraste zu erkennen, dass die WCAG für Karten nur schwer durchsetzbar ist. Da potentiell jede Informationsklasse adjazent zu jeder anderen Informationsklasse liegen kann, ist die konsequente Durchsetzung bei der Anzahl an Landbedeckungen unmöglich. In den entwickelten Styles wird versucht, insbesondere den Kontrastunterschied bei Problemklassen möglichst hoch zu gestalten.

Deuteranopiestyle

Die Farbgebung des Deuteranopiestyle muss ähnliche Farbkombination, wie der Protonopiestyle vermeiden. Hier sind die Kontraste zwischen den roten Siedlungsflächen und Wald und Grünflächenelementen aber so groß, dass Deuteranopen zwischen den Landbedeckungen unterscheiden können.

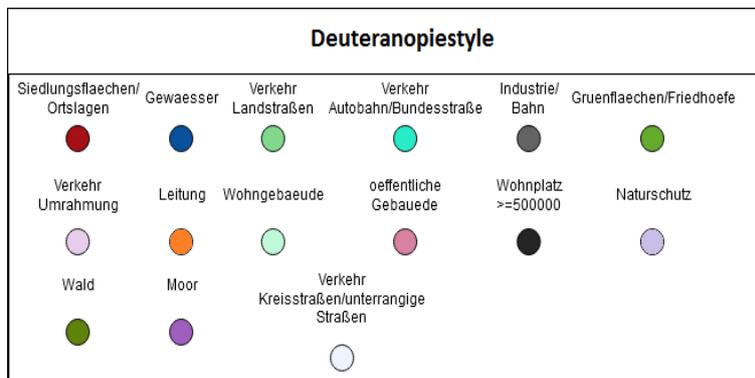


Abbildung 23: Farbpalette für Deuteranopen Quelle: Eigene Abbildung.

Aufgrund der Menge an Informationsklassen ist hier das gleiche Verfahren bezüglich der Kontrastunterschiede von Problemklassen, wie beim Protonopiestyle verwendet worden.

Tritanopiestyle

Der Tritanopiestyle hebt sich in der Farbgebung deutlich von den beiden anderen Styles ab. Grund dafür sind potentielle Verwechslungen von Blau und Grün und Gelbgrün und Grau, die in diesem Style vermieden werden sollen. Die gewählten Grün und Blautöne des Styles sind aber

so kontrastreich (Kontrast $< 4:1$), dass die Erkennbarkeit von adjazenten Informationklassen trotz dieser Problematik bestehen bleibt.

Tritanopie Style					
Siedlungsflaechen/ Ortslagen	Gewaesser	Verkehr Landstraßen	Verkehr Autobahn/Bundesstraße	Industrie/ Bahn	Gruenflaechen/Friedhoeefe
●	●	●	●	●	●
Verkehr Umrahmung	Leitung	Wohngebaeude	oeffentliche Gebaeude	Wohnplatz >=500000	Naturschutz
●	●	●	●	●	●
Wald	Moor	Verkehr Kreisstraßen/unterrangige Straßen			
●	●	●			

Abbildung 24: Farbpalette für Tritanopen Quelle: Eigene Abbildung.

Umsetzung der Styles im Prototypen

Für jeden der Styles ist eine JSON-Datei erstellt worden, die den Mapbox Spezifikation entspricht. Um nicht vom Basisstyle des LGLN abzuweichen, sind einige Eigenschaften der anderen Styles, wie beispielsweise die Größe der Straßen oder die Darstellung bei bestimmten Zoomstufen übernommen worden. Die so erstellten JSON-Dateien werden in einem bestimmten Ordner gespeichert⁴⁰ und können dann vom Frontend benutzt werden.

Über die in Kapitel 5.1 vorgestellte Kartenstyling-Funktion kann man das Styling der Karte auf die Farbenblindstyles umstellen. Auf Programmiererebene wurde die Liste mit Kartenstyles um die drei Farbenblindstyles durch drei weitere Themes erweitert. Ein Theme ist dabei eine Klasse in Typescript, die aus einer ID, einem Namen, sowie einem Bild und einem Styling besteht. Neben dem Style ist ein Bild erstellt worden, welches den jeweiligen Style im Menü des Prototypen repräsentiert. Es besteht für die entwickelten Farbenblindstyles, wie bei den vorhandenen Styles aus 256x256 Pixeln und zeigt die Innenstadt Hamelns. Der Speicherort muss dabei im Theme genannt werden, um auf die Ressource zugreifen zu können.

⁴⁰ Als Assets unter `themes\farbenblindstyle1 – 3.json`.

7.2 Evaluation der entwickelten Styles

Um die Leistungsfähigkeit der Neuentwicklung im Bereich Visualisierung von Geobasisdaten für Farbenblinde zu testen, ist zunächst eine Testreihe mit Probanden geplant gewesen. Da die Krankheit in der Bevölkerung aber relativ selten⁴¹ ist, sind nicht genügend Testpersonen gefunden worden, um eine statistisch signifikante Aussage zu tätigen.

Aus diesem Grund werden die Styles mithilfe einer Webseite ausgewertet, die die unterschiedlichen Krankheitstypen der Farbenfehlsichtigkeit anhand eines RGB-Bilds simulieren kann.

Das Programm Coblis (vgl. Kapitel 4.11) kann RGB-Bilder in die dichromatische Sicht eines Farbenblinden umwandeln. Das Programm wird im Folgenden dazu benutzt, um die in Kapitel 7.1 entwickelten Kartenstyles unter dichromatischen Bedingungen zu testen.

Protanopiestyle

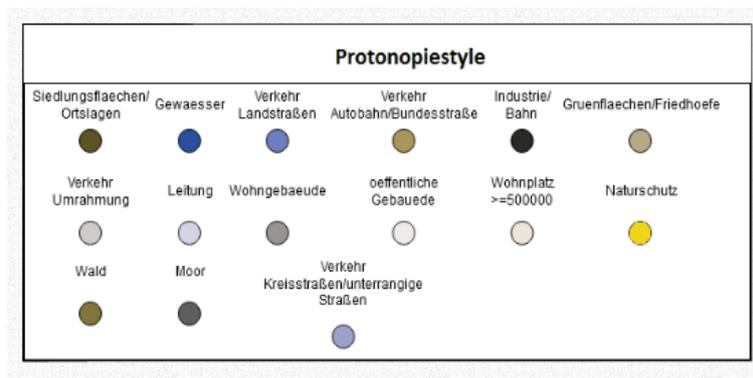


Abbildung 25: Farbpalette aus Sicht eines Protanopen Quelle: Eigene Abbildung.

Die dichromatische Sicht zeigt, dass es möglich ist einzelne Landbedeckungen voneinander zu unterscheiden. Farblich gestaltet sich die Karte für Protanopen in unterschiedlichen Braun- und Blautönen. Die Auswahl von Türkis statt Grün macht es möglich Ortslagen von Grünflächen, Mooren und Wohngebäuden zu unterscheiden. Hellgrün hebt sich aufgrund des hohen Lumineszenzwert so deutlich von den Rottönen ab, dass beide die Farben von Protanopen unterschieden werden können.

Deutanopiestyle

Die Farbpalette für an Deutanopie erkrankte Personen ist in Braun- und Blautönen gehalten. In der RGB-Kartendarstellung ähnelt Dunkelrot dabei den Grüntönen, besitzt aber eine geringere Lumineszenz und kann deswegen aufgrund des Kontrast von Grünflächen, Wald und Landstraßen unterschieden werden. Die Farbpalette zeigt, dass es keine Probleme mit adjazenten Farben gibt, sodass sich dieser Style für Deutanopen eignet.

⁴¹ Ca. 8% der männlichen und 0,5% der weiblichen Bevölkerung sind Farbenfehlsichtig[50].

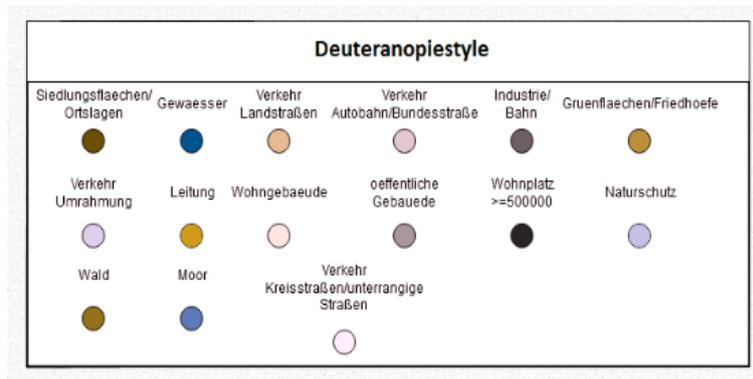


Abbildung 26: Farbpalette aus Sicht eines Deuteranopen Quelle: Eigene Abbildung.

Tritanopiestyle

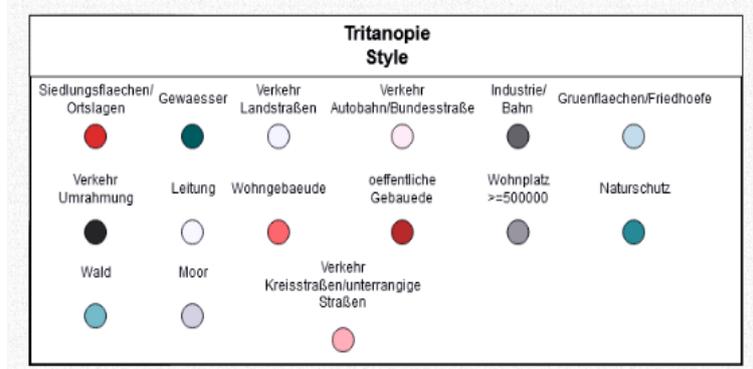


Abbildung 27: Farbpalette aus Sicht eines Tritanopen Quelle: Eigene Abbildung.

Auch der Tritanopenstyle zeigt, dass die Landbedeckungsklassen unterscheidbar sind. Die Farbtöne stellen sich für Tritanopen als Rot-, Blau- und Grautöne da. Problematische Kombinationen für diese Art der Dichromatie sind durch den hohen Kontrastunterschied von Grün und Blau, sowie Gelb und Grau nicht relevant für die Unterscheidbarkeit.

Die Tests aus Sicht von Dichromaten haben gezeigt, dass jeder der drei verfügbaren Styles sich für die Verwendung im Prototypen eignet. So hat die Weiterentwicklung im Bereich des Stylings dafür gesorgt, dass Farbenfehlsichtige die Kartenfunktion der Applikation sinnvoll nutzen können. Dabei hat sich die Weiterentwicklung primär mit dem Mapping des Visualisierungsprozess beschäftigt (vgl. Kapitel 4.1). In der Diskussion soll unter anderem besprochen werden, in wie weit diese Entwicklung Relevanz in der Praxis besitzt.

8 Diskussion & Ausblick

Zu Beginn der Arbeit sind zwei Forschungsfragen aufgestellt worden, die im Verlauf der Diskussion besprochen und beantwortet werden sollen.

Frage 1.: Wie ist die Nutzung des Vektor-Tile-Formats im Vergleich zu anderen in der Wissenschaft existierenden Standards zu bewerten?

Die Literaturrecherche hat ergeben, dass Vektor-Tiles als eine Grundlage der Visualisierung von Geodaten viele Vorteile bieten, die die sonst häufig benutzten Raster-Tiles nicht besitzen. Vektor-Tiles werden on-the-fly gerendert, sodass die Daten ohne Netzwerkbelastung in ihrer Art der Präsentation verändert werden können. Diese Eigenschaft ist auch in der Evaluation des Prototypen als ein Unterschied zwischen Geoviewern auf Rasterbasis und Geoviewern auf Vektorbasis gezeigt worden (vgl. Kapitel 6).

Weitere Vorteile, wie die vergleichsweise kleine Größe der Tiles oder die Möglichkeit simple Geoverarbeitungsschritte durch den Clienten durchzuführen sind zwar in der Literatur beschrieben (vgl. Kapitel 4.3), können aber durch die Evaluation des Prototypen nicht belegt werden, da der technische Stand dies zur Zeit nicht zulässt.

Falls der Prototyp in Zukunft durch die Landesbehörde veröffentlicht wird, kann der Evaluationsprozess bei fortschreitender technischer Umsetzung nochmals durchgeführt werden, um auch diese Vorteile an einem konkreten Beispiel zu zeigen.

Ein weiterer interessanter Aspekt der weiteren Forschung Bedarf, ist die Entwicklung von clientseitigen Geoverarbeitungsschritten auf Basis von Vektor-Tiles. In der Praxis liegen die Geodaten mit ihren Objektgeometrien beim Clienten, sodass eine Geoverarbeitung vergleichsweise einfach möglich ist. Problematisch dabei sind Grafikkonflikte und Informationsverluste, die an den Kachelgrenzen der Tiles auftreten können. In diesem Spannungsfeld können neue Erkenntnisse über die Möglichkeiten von Vektor-Tiles als Datenbasis für die Visualisierung von Geodaten gesammelt werden.

Frage 2.: Welche Methoden werden im Rahmen der Weiterentwicklung des Prototypen hinzugefügt? Welche Aussagen können dabei über die Visualisierung von Geobasisdaten abgeleitet werden?

Im Rahmen der Weiterentwicklung des Prototypen wurden drei neue Mapboxstyles entwickelt, die die Erreichbarkeit der Seite mithilfe einer alternativen Visualisierung der Geodaten erhöhen sollen.

Für jede der dichromatischen Farbfeldsichtigkeiten (vgl. Kapitel 4.9) ist ein Style implementiert worden, der sich durch Unterscheidbarkeit der Landbedeckungen auszeichnet. Diese Fähigkeit der Neuentwicklung wurde mithilfe von CobliS getestet und für alle Styles bestätigt.

Die so entwickelten Kartenstyles zeigen, dass mithilfe von Visualisierung von Geodaten auch ei-

ne Grundlage für barrierefreien Zugang zu Internetinhalten gewährt werden kann. Für beliebige Karten lässt sich dieses Konzept aber nicht erweitern, da sich mit der Zahl der Informationsklassen auch die Anzahl an möglichen Farbpaarungen vervielfältigt. Vor dem Hintergrund eine kontrastreiche Karte anzubieten, ist der Grad zwischen Anzahl der Informationsklassen und Eignung für Farbenfehlsichtige schmal.

Da die Styles aber als JSON-Daten vorliegen können sie in anderen Projekten, die ATKIS-Daten als Grundlage haben verwendet werden und bieten deshalb eine barrierefreie Möglichkeit der Visualisierung von Geobasisdaten.

Die Masterarbeit hat zudem weitere kontroverse Themen in Bezug auf Visualisierung von Geobasisdaten zum Inhalt, die in folgendem Abschnitt diskutiert werden sollen.

Der Prototyp basiert auf Styles, die nicht mehr durch den ATKIS-Signaturenkatalog definiert werden, indem jedes Kartenelement eine feste Art der Visualisierung zugewiesen bekommt. Dies kann man als eine Art Paradigmenwechsel der Landesbehörde sehen, da das Styling der Karte nun vom Entwicklerteam vorgenommen wird und relativ frei gestaltet ist. Um diesen Sachverhalt einzuordnen ist hier nochmal die These des VKV aufgeführt, mit der sich die Arbeitsgruppe Basisvisualisierung beschäftigt:

Basisvisualisierungen ersetzen die Digitale Topographische Karte (DTK) und sind Grundlage für sämtliche (Standard-) Präsentationen.

Die digitale topographische Karte und damit auch ihre Art der Visualisierung soll durch Basisvisualisierungen ersetzt werden. Basisvisualisierungen sind nicht näher definiert, bestehen aber im Prototypen aus Vektor-Tiles, die durch unterschiedliche Styles visualisiert werden. Demnach ist die These ein klarer Bruch zum vorherigen Visualisierungsprozess, der auf Signaturenkataloge gestützt ist.

Dieser Prozess kann Vor- und Nachteile mit sich bringen, wird aber von der Landesbehörde als positive Entwicklung beschrieben. So sind Karten nicht mehr nur an eine Form der Visualisierung gebunden und können frei gestaltet werden.

Auf der anderen Seite sind die geltenden Signaturenkataloge für Basisvisualisierung obsolet und werden nicht mehr benötigt. Dies kann unter Umständen zu Problemen bei der Interoperabilität der Daten führen, wenn jede Landesbehörde eigene Styles entwickelt und so kein einheitlicher Standard verfügbar ist. Da das Land Niedersachsen sich aber als eine der ersten Landesbehörden mit der Visualisierung von Vektor-Tiles beschäftigt, kann sich die Art der Visualisierung als eine Art de facto Standard durchsetzen und Signalwirkung für andere Landesbehörden haben.

In der Literatur ist eine Entwicklung von Geoviewern zu digitalen Globen, also intelligenten multidimensionalen Geoviewern zu beobachten (vgl. Kapitel 3.1). Dabei werden mehrdimensionale Daten in den Geoviewer eingepflegt, um dem Nutzer eine besser Art der Präsentation (vgl. Map use cube Kapitel 2.1) zu bieten.

Die geplanten Änderungen am Prototypen⁴² legen nahe, dass der Prototyp im Stadium der Veröffentlichung einem solchen digitalen Globus ähnelt. Mit den Daten können potentielle Kunden angesprochen und Dienstleistung angeboten werden.

Auf diesem Weg müssen aber noch einige Änderungen erfolgen, wie beispielsweise die Berücksichtigung rechtlicher Aspekte oder ein Test auf Nutzbarkeit der Applikation.

Die Grundlage für den Prototypen, Basisvisualisierung mithilfe von Vektor-Tiles ist aber bereits gelegt und kann von den Vorteilen, die in dieser Arbeit behandelt wurden profitieren.

⁴² Zum Beispiel die Implementation von 3D-Modellen oder der Navigation vom ORS.

Erklärung zur selbstständigen Abfassung der Masterarbeit

Name:.....
Geburtsdatum:.....
Matrikelnummer:.....
Titel der Masterarbeit:.....

Ich versichere, dass ich die eingereichte Masterarbeit selbstständig und ohne unerlaubte Hilfe verfasst habe. Anderer als der von mir angegebenen Hilfsmittel und Schriften habe ich mich nicht bedient. Alle wörtlich oder sinngemäß den Schriften anderer Autoren entnommenen Stellen habe ich kenntlich gemacht.

Ort, Datum

Unterschrift

Literatur

- [1] Weber, W. et al.: Interaktive Infografiken. 2013. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2013. S.39.
ISBN: 978-3-642-15453-9
- [2] O'Reilly, T.: Was ist Web 2.0? Entwurfsmuster und Geschäftsmodelle für die nächste Software Generation. Online unter:
<http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=2>
(abgerufen am 07.05.2019)
- [3] Mittlböck, M.: Task-orientierte Web-Maps zur kompakten Visualisierung kartographischer Inhalte. Angewandte Geoinformatik 2012: Beiträge zum 24. AGIT - Symposium Salzburg. Hrsg. Strobl, J. et al.. Heidelberg: Wichmann, 2012. 333-338. Print. S.333.
ISBN: 978-3-879-07520-1
- [4] Blankenbach, J. et al.: Über Web 2.0 und GDI zur ePartizipation 2.0 Ein GIS-basierter Ansatz zur Unterstützung der Partizipation in Verwaltungs- und Planungsprozessen von Kommunen. Tuprints. S.3f. Online unter:
<http://tuprints.ulb.tu-darmstadt.de/2692/>
(abgerufen am 07.05.2019)
- [5] Landesamt für Geoinformation und Landesvermessung Niedersachsen: Jahresbericht 2017. 2017. S.18. Online unter:
http://www.lgln.niedersachsen.de/download/132062/Jahresbericht_2017.pdf
(abgerufen am 04.06.2019)
- [6] Rausch, S.: Fachliches Zukunftskonzept für die Vermessungs- und Katasterverwaltung Zieljahr 2025. In: Nachrichten der Niedersächsischen Vermessungs- und Katasterverwaltung-1+2 2017. Hrsg. Landesamt für Geoinformation und Landesvermessung Niedersachsen. S.7f. Online unter:
http://www.lgln.niedersachsen.de/download/128681/NaVKV_2017_1-2.pdf
(abgerufen am 04.06.2019)
- [7] Schumann, H., Müller, W.: Visualisierung : Grundlagen und allgemeine Methoden. 2000. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2000. S. VI., S.15-17, S.255.
<http://doi.org/10.1007/978-3-642-57193-0>
- [8] MacEachren, A. M., Kraak, M.-J.: Research Challenges in Geovisualization. Cartography and Geographic Information Systems, 28. 2001. S.3.
<http://doi.org/10.1559/152304001782173970>
- [9] Schiewe, J.: Was ist Geovisualisierung? Online unter:
http://geovisualisierung.net/geovis_defs.htm
(abgerufen am 27.05.2019)

- [10] MacEachren, A. M., Kraak, M. - J.: Exploratory cartographic visualization: Advancing the agenda. *Computers & Geosciences* 23(4): 335-43. 1997. S.337, S.339.
<http://doi.org/10.1002/9780470979587.ch11>
- [11] Landesamt für Geoinformation und Landesvermessung Niedersachsen: Geodaten & Karten. Online unter:
http://www.lgln.niedersachsen.de/geodaten_karten/geodaten--karten-51666.html
 (abgerufen am 28.05.2019)
- [12] Behnisch, M. et al.: Flächeninanspruchnahme in Deutschland: Auf dem Wege zu einem besseren Verständnis der Siedlungs- und Verkehrsflächenentwicklung. 1.Aufl.. Berlin Heidelberg New York: Springer-Verlag, 2018. S. 28f.
ISBN: 978-3-662-50305-8
- [13] Lange, N. de: Geoinformatik: in Theorie und Praxis. 3. Aufl.. Berlin Heidelberg New York: Springer-Verlag, 2013. S.26, S.218, S.273.
ISBN: 978-3-642-34807-5
- [14] Wiebrock, I.: Zur kontextbasierten Visualisierung von Geodaten auf Basis von standardisierten Webdiensten. Unv. Diss., Universität der Bundeswehr München 2011. S.10, S.20, S.31, S.33, S.113f., S.123.
<http://d-nb.info/1013350650/34>
- [15] Meinel, G.: Monitoring of Settlement and Open Space Development on the Basis of Topographical Spatial Data- Concept, Realization and first Results. In: *Core Spatial Databases- Updating, Maintenance and Services from Theory to Practice*, Haifa, Israel. ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2010. S.1 - 5.
- [16] Brovelli, M. A. et al.: New Generation Platforms for Exploration of Crowdsourced Geo-Data. *Earth Observation Open Science and Innovation*. Hrsg. Mathieu, P. - P. et al.. Berlin, Heidelberg: Springer, 2018. S.219, S.220f., S.225 - 227.
ISBN: 978-3-319-65633-5
- [17] Menninghaus, M.: Mobile Geodatenvisualisierung mit Android. In: Bick, M. et al. (Hrsg.), *Mobile und Ubiquitäre Informationssysteme*. Bonn: Gesellschaft für Informatik e.V.
- [18] Steffen, D. et al.: GeoVisualizer – ein neuer Ansatz zur Visualisierung von Sensordaten im raumplanerischen Kontext. *Angewandte Geoinformatik 2013: Beiträge zum 25. AGIT-Symposium Salzburg*. Hrsg. Strobl, J. et al.. Heidelberg: Wichmann, 2013.
ISBN: 978-3-879-07533-1

- [19] Fernandez, A. et al.: Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology* 53(8). 2011. S.2., S.7, S.9.
<http://dx.doi.org/10.1016/j.infsof.2011.02.007>
- [20] Abran, A. et al: Consolidating the ISO Usability Models. *Proceedings of 11th International Software Quality Management Conference and the 8th Annual INSPIRE Conference*, 2003. S.7.
- [21] Lew, P. et al.: Quality, quality in use, actual usability and user experience as key drivers for web application evaluation. *LNCS*. 6189. 2010. S.231.
http://doi.org/10.1007/978-3-642-13911-6_15
- [22] Nayebi, F. et al.: The state of the art of mobile application usability evaluation. *CCECE*. 2012. S.1f.
<http://doi.org/10.1109/CCECE.2012.6334930>
- [23] Ivory, M. Y., Hearst, M. A.: The state of the art in automating usability evaluation of user interfaces. In: *ACM Computing Surveys* 33. 2001. S.473.
<http://doi.org/10.1145/503112.503114>
- [24] Kaur, S. et al.: An empirical performance evaluation of Universities Website. In: *International Journal of Computer Applications* 146(15). 2016. S.11.
<http://doi.org/10.5120/ijca2016910922>
- [25] Krol, K.: Comparative analysis of the performance of selected raster map viewers. In: *Geomatics, Landmanagement and Landscape Vol.2*. 2018. o.S.
<http://doi.org/10.15576/GLL/2018.2.23>
- [26] Litoiu M. & Barna, C.: A performance evaluation framework for web applications. In: *Journal of Software: Evolution and Process*. Vol.25. 2012. S. 872f.. S.888f.
<http://doi.org/10.1002/smr.1563>
- [27] Sample, J. T., Ioup, E.: *Tile-Based Geospatial Information Systems : Principles and Practices*. Aufl.. Berlin Heidelberg: Springer Science & Business Media, 2010. S.1, S.2, S.27, S.81-83, S.88f., S.194f.
<http://doi.org/10.1007/978-1-4419-7631-4>
- [28] Wikimedia Commons contributors, 'File:Tiled web map Stevage.png', Wikimedia Commons, the free media repository. Online unter:
https://commons.wikimedia.org/wiki/File:Tiled_web_map_Stevage.png
 (abgerufen am 04.07.2019)
- [29] Garcia, R. et al.: Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization. In: *Cartography - A Tool for Spatial Analysis Infortech*, 2012, S.26f.
<http://doi.org/10.5772/2704>

- [30] Stefanakis, E.: Web meicator and raster tile maps: two cornerstones of online map service providers. In: *Geomatica* 71(2), 2017, S.8f.
<https://doi.org/10.5623/cig2017-203>
- [31] Christiansen, T.: Lexikon der Geographie: Rasterdaten. Online unter:
<https://www.spektrum.de/lexikon/geographie/rasterdaten/6416>
 (abgerufen am 17.07.2019)
- [32] Bartelme, N.: *Geoinformatik : Modelle, Strukturen, Funktionen*. 4. Aufl.. Berlin Heidelberg New York: Springer-Verlag, 2005. S.65.
ISBN: 978-3-540-27201-4
- [33] ESRI: Vector Tile Coverage Map vs. Raster Tile Coverage Map. Online unter:
<https://telecomrc.maps.arcgis.com/apps/StorytellingSwipe/index.html?appid=2dcc25643b3b49739cf090f1770cd741&webmap=d2debce328d44a1b90f2a2f90228c532#>
- [34] Gaffuri, J.: Towards Web Mapping with Vector Data. In: *Geographic Information Science*, edited by X. Ningchuan, M.-P. Kwan, M. F. Goodchild, and S. Shekhar, 87–101. Berlin Heidelberg: Springer. S.88.
http://doi.org/10.1007/978-3-642-33024-7_7
- [35] Cavazzi, S.: OGC Testbed- 13: Vector Tiles Engineering Report. OGC. S.11.
- [36] Ingensand, J. et al.: Implementation of tiled vector services: a case study. *GIScience* 2016. S.1f.
http://doi.org/10.1007/978-3-642-10601-9_5
- [37] O.A.: Vector tiles tutorial. *GeoServer 2.17.x User Manual*. Online unter:
<https://docs.geoserver.org/latest/en/user/extensions/vectortiles/tutorial.html>
 (abgerufen am 06.09.2019)
- [38] Yu, E. G. et al.: Performance improvement on a Web Geospatial service for the remote sensing flood-induced crop loss assessment web application using vector tiling. In: *2017 6th international conference on agro- geoinformatics*. Fairfax, VA: IEEE. 2017. S.2.
<http://doi.org/10.1109/Agro-Geoinformatics.2017.8047053>
- [39] Esri: *Comparing Vector and Raster Mapping for Internet Applications*, An ESRI White Paper. 2006. S.2.
- [40] Google: *Developer Guide Protocol Buffers*. Online unter:
<https://developers.google.com/protocol-buffers/docs/overview>
 (abgerufen am 15.09.2019)

- [41] Mapbox: Vector tile specification. Online unter:
<https://docs.mapbox.com/vector-tiles/specification/>
(abgerufen am 15.09.2019)
- [42] Github: Vector Tile Specification. Online unter:
<https://github.com/mapbox/vector-tile-spec/tree/master/2.1>
(abgerufen am 15.09.2019)
- [43] Mapbox: Style Specification. Online unter:
<https://docs.mapbox.com/mapbox-gl-js/style-spec/#glyphs>
(abgerufen am 16.09.2019)
- [44] Anderson, C.: Docker [Software engineering]. In: IEEE Software, Vol.32. No.3, 2015. S.102f..
<http://doi.org/10.1109/MS.2015.62>
- [45] Kalberer, P.: Von WMS zu WMTS zu Vektor-Tiles. In: FOSSGIS - Konferenz 2017 Passau 22.05 - 25.05. Online unter:
<https://av.tib.eu/media/30549>
(abgerufen am 17.09.2019)
<http://doi.org/10.5446/30549>
- [46] Momjian, B.: PostgreSQL Introduction and Concepts. In: Addison-Wesley, 2000. S.4.
ISBN: 978-0-201-70331-3
- [47] Strobl, C.: PostGIS. In: Encyclopedia of GIS., Shekhar, S., Xiong, H., Eds. Springer 2008. S.891
https://doi.org/10.1007/978-0-387-35973-1_1012
- [48] Google Chrome: Lighthouse Scoring. Online unter: <https://github.com/GoogleChrome/lighthouse/blob/d2ec9ffbb21de9ad1a0f86ed24575eda32c796f0/docs/scoring.md>
(abgerufen am 11.10.2019)
- [49] Welsch, N., Liebmann, C. Chr.: Farben: Natur, Technik, Kunst. 3. Aufl.. Berlin Heidelberg New York: Springer-Verlag, 2012. S.227, S.241, S.269f.
ISBN: 978-3-8274-2846-2
- [50] Luebbe, E.: Farbemfindung, Farbbeschreibung und Farbmessung: Eine Formel für die Farbsättigung. 2013. Aufl.. Berlin Heidelberg New York: Springer-Verlag, 2012. S.64
ISBN: 978-3-834-81801-0
- [51] Cooper, M. et al.: Web Content Accessibility Guidelines (WCAG) 2.1. S.1. Online unter:
<https://www.w3.org/TR/WCAG21/>
(abgerufen am 21.9.2019)

- [52] Europäische Union: Richtlinie 2016/2102 des europäischen Parlaments und des Rates. 2016. Online unter:
<https://eur-lex.europa.eu/legal-content/DA/TXT/PDF/?uri=CELEX:32016L2102&from=DE>
(abgerufen am 13.10.2019)
- [53] Muttaqin, G. F., Suwandi, I. S.: Simulation System of Color Blind Glasses by Image Processing. In: ICEEI 2011 Indonesia, 2011, o.S.
<http://doi.org/10.1109/ICEEI.2011.6021616>
- [54] Landesamt für Geoinformation und Landesvermessung Niedersachsen: Niedersachsen-Viewer. Online unter:
https://www.geodaten.niedersachsen.de/viewer_dienste/niedersachsenviewer/niedersachsenviewer-25442.html
(abgerufen am 19.09.2019)
- [55] Hulce, P.: Lighthouse Metric Variability and Accuracy. S.3-5. Online unter:
https://docs.google.com/document/d/1BqtL-nG53rxW0I5R00pItSRPowZVnYJ_gBEQCJ5EeUE/edit
(abgerufen am 19.09.2019)
- [56] W3C: Web Content Accessibility Guidelines (WCAG) 2.1. Online unter:
<https://www.w3.org/TR/WCAG21/#dfn-relative-luminance>
(abgerufen am 20.09.2019)
- [57] Krzywinski, M.: Color Palettes for Color Blindness. Online unter:
<http://mkweb.bcgsc.ca/colorblind/>
(abgerufen am 02.10.2019)