



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

**„Evaluierung von Vektorformaten zur Visualisierung am
Hyperglobus“**

verfasst von / submitted by

Sara Lena Kordasch, BSc

angestrebter akademischer Grad / in partial fulfillment of the requirements for the degree of

Master of Science (MSc)

Wien, 2019 / Vienna, 2019

Studienkennzahl lt. Studienblatt /

degree programme code as it appears on
the student record sheet:

A 066 856

Studienrichtung lt. Studienblatt /

degree programme as it appears on
the student record sheet:

Kartographie und Geoinformation

Betreut von / Supervisor:

Ass.-Prof. Mag. Dr. Andreas Riedl

Inhalt

INHALT	III
ABBILDUNGSVERZEICHNIS.....	VI
TABELLENVERZEICHNIS	X
FORMELVERZEICHNIS.....	XI
ABKÜRZUNGSVERZEICHNIS	XII
KURZFASSUNG.....	XIII
ABSTRACT	XIV
VORWORT	XV
1 EINLEITUNG	1
1.1 PROBLEMSTELLUNG.....	1
1.2 ZIELSETZUNG.....	2
1.3 FRAGESTELLUNG	2
1.4 METHODIK	2
1.5 MOTIVATION	3
1.6 AUFBAU DER ARBEIT	3
2 GEOVISUALISIERUNG AM SPHÄRISCHEN DISPLAY	5
2.1 SPHÄRISCHE DISPLAYS IN DER GLOBENFAMILIE.....	5
2.1.1 Virtueller Hyperglobus	6
2.1.2 Taktiler Hyperglobus	8
2.2 ANFORDERUNGEN AN EINE VISUALISIERUNG AM SPHÄRISCHEN DISPLAY	9
2.2.1 Kartographische Gestaltung.....	9
2.2.1.1 Punkt	10
2.2.1.2 Linie	10
2.2.1.3 Fläche.....	11
2.2.1.4 Schrift.....	12
2.2.1.5 Legende.....	13
2.2.1.6 Generalisierung	13
2.2.2 Verzerrungsproblematik am sphärischen Display.....	13
2.2.3 Globenwürdigkeit	15
2.2.4 Thematiken	16
2.2.5 Animation	17
2.2.6 Interaktion	17
3 DATEIFORMATE	19
3.1 DAS RASTERMODELL.....	19
3.1.1 Rasterformate am sphärischen Display.....	20

3.1.1.1	JPEG	21
3.1.1.2	PNG	22
3.1.1.3	DDS	22
3.2	DAS VEKTORMODELL	24
3.2.1	Vektormodelle.....	25
3.2.2	Skalenspektrum & Variabilität	26
3.3	DATEIFORMATE IM VERGLEICH	28
3.3.1	Dateiformate am sphärischen Display	28
4	METHODEN DER GEOVISUALISIERUNG AM SPHÄRISCHEN DISPLAY	30
4.1	DIREKTE RASTERVISUALISIERUNG	31
4.2	INDIREKTE RASTERVISUALISIERUNG	32
4.3	DIREKTE VEKTORVISUALISIERUNG.....	33
4.4	VERGLEICH DER VISUALISIERUNGSMETHODEN	35
5	ANALYSE UMGESETZTER GLOBAL STORIES	36
5.1	EARTHQUAKES 1914 – 2014	36
5.2	GLOBAL SEA LEVEL RISE	37
5.3	TIEFSEEKABEL	38
5.4	DISKUSSION DER ANALYSIERTEN GLOBAL STORIES.....	39
6	VEKTORFORMATE IN DER GEOINFORMATION.....	40
6.1	SIMPLE FEATURE GEOMETRIEN	40
6.1.1	Punkt / Multipunkt [Point/MultiPoint]	40
6.1.2	Linie [Curve]	41
6.1.3	Flächen [Surface].....	41
6.1.4	Methoden der Simple Feature Specification.....	42
6.1.5	Vektorformate im Rahmen der Simple Feature Specification.....	43
6.2	VERBREITUNG VON VEKTORFORMATEN AUF OPEN-DATA PLATTFORMEN	45
7	THEORETISCHE FORMATANALYSE.....	47
7.1	FORMATGRUNDLAGEN.....	47
7.1.1	XML	47
7.1.2	JSON.....	48
7.2	GRAPHISCHE AUFBEREITUNG VON VEKTORFORMATEN	49
7.2.1	CSS	50
7.3	KOMMERZIELLE VEKTORFORMATE	50
7.3.1	SHP.....	50
7.4	OFFENE VEKTORFORMATE	54
7.4.1	GML.....	54
7.4.2	KML/KMZ.....	57
7.4.3	GeoJSON	61
7.4.4	SVG.....	64
7.4.5	WKT/WKB Geometry.....	66
7.4.6	OmniSuite Punkt-Format.....	68

7.5	BEURTEILUNG DER VEKTORFORMATE	70
8	PRAKTISCHE FORMATANALYSE	65
8.1	UMSETZUNG EINER GLOBAL STORY NACH DEM AKTUELLEN ANSATZ	65
8.1.1	Der Datensatz	65
8.1.2	Umsetzung einer Global Story für Global Flood Events	66
8.1.2.1	Der Punktdatensatz	67
8.1.2.2	Der Polygon-Datensatz	72
8.1.3	Erstellen der Global Story für den Testdatensatz	78
8.1.4	Analyse der Erstellung einer Global Story nach dem aktuellen Ansatz	79
8.2	PRAKTISCHE ANALYSE DER GEWÄHLTEN FORMATE	80
8.2.1	KML	80
8.2.1.1	Graphische Aufbereitung - KML	81
8.2.1.2	Animation und Interaktion – KML	85
8.2.1.3	Formatbibliotheken in C++ – libkml	86
8.2.2	SVG	87
8.2.3	GeoJSON	88
8.2.3.1	GeoJSON – CSS	88
8.2.3.2	GeoJSON - Mapbox	91
9	DISKUSSION DER FORMATANALYSEN	98
10	ZUSAMMENFASSUNG	103
11	LITERATURVERZEICHNIS	105
12	ANHANG	110
12.1	XML	110
12.2	JSON	111
12.3	STYLESHEETS	111
12.4	SHAPEFILE	116
12.5	GML	117
12.6	GEOJSON	118
12.7	SVG	121
12.8	OMNISUITE PUNKTFORMAT	121
12.9	KML	123
12.10	MAPBOX	136
	ERKLÄRUNG	141

Abbildungsverzeichnis

Abbildung 1: Digitaler Behaim-Globus ([DOR-96: 22])	6
Abbildung 2: "Earth Browser" ("Planet Earth") ([DIG-08])	7
Abbildung 3: Hyperglobus ([RIE-11: 155])	7
Abbildung 4: Taktiler Hyperglobus ([DGF-12])	8
Abbildung 5: Linienobjekte ([HAK-02: 120])	11
Abbildung 6: Flächenobjekte ([HAK-02:121])	11
Abbildung 7: Schrift am Hyperglobus ([KIE-18])	12
Abbildung 8: Legende am Hyperglobus ([KIE-18])	13
Abbildung 9: Generalisierung von Punktoobjekten am Hyperglobus (Datengrundlage: [EAR-17])	13
Abbildung 10: Visualisierung am sphärischen Display ohne Berücksichtigung der Verzerrung (2D → 3D) ([LIE-12: 2])	14
Abbildung 11: Visualisierung am sphärischen Display unter Berücksichtigung der Verzerrung (2D -> 3D) ([LIE-12: 2])	15
Abbildung 12: Rasterformat ([WAS-16])	20
Abbildung 13: Qualitätsunterschiede durch Komprimierung beim *.jpeg Format ([WIC-02])	21
Abbildung 14: PNG ohne/mit Transparenz (eigene Darstellung)	22
Abbildung 15: In der Globensoftware verwendete Bildformate ([BIL-17: 5])	23
Abbildung 16: Originalbild vs. komprimiertes Bild ([BIL-17: 4])	24
Abbildung 17: Geometrische Grundprimitiva (eigene Darstellung)	25
Abbildung 18: Verlustfreie Skalierung Vektorformat (Datengrundlage: [NAT-18])	25
Abbildung 19: Vektormodelle ([HAK-02:158-159])	26
Abbildung 20: Methoden zur Datenintegration am Hyperglobus ([LIE-12: 45])	30
Abbildung 21: Interne, synchrone Datenintegration am sphärischen Display ([LIE-12: 47])	31
Abbildung 22: Direkte Rastervisualisierung OmniSuite (Datengrundlage: [NAT-18])	32
Abbildung 23: Indirekte Rastervisualisierung OmniSuite (eigene Darstellung)	33
Abbildung 24: Vektorintegration am sphärischen Display - Variante 1 ([LIE-12: 50])	34

Abbildung 25: Vektorintegration am sphärischen Display -Variante 2 ([LIE-12: 50]).....	34
Abbildung 26: Vektorintegration am sphärischen Display -Variante 3 ([LIE-12: 50]).....	35
Abbildung 27: Earthquakes 1914 – 2014 ([KRI-14]).....	37
Abbildung 28: Global Story Sea Level Rise ([KRI-18]).....	38
Abbildung 29: Global Story Tiefseekabel ([KIE-18]).....	39
Abbildung 30: Korrekter vs. inkorrekt Tagaufbau in XML (eigene Darstellung).....	48
Abbildung 31: JSON vs. XML Datenstruktur (eigene Darstellung).....	49
Abbildung 32: CSS Beispiel (eigene Darstellung).....	50
Abbildung 33: Geometrietypen eines Shapefiles (eigene Darstellung).....	51
Abbildung 34: Shapefile im Texteditor (eigene Darstellung).....	51
Abbildung 35: PRJ Datei eines Shapefiles (eigene Darstellung).....	52
Abbildung 36: Layer File ArcMap (*.lyr) (eigene Darstellung).....	52
Abbildung 37: Layerdefinitionsdatei QGIS (*.qrl) (eigene Darstellung).....	53
Abbildung 38: Animationseinstellungen in ArcGIS ([HOW-17]).....	53
Abbildung 39: GML Objektstruktur ([BUR-06: 180]).....	55
Abbildung 40: GML Objekt-Attributstruktur ([BUR-06: 180]).....	55
Abbildung 41: Codebeispiel von „Remote“ GML-Objekten ([LAK-04: 85]).....	56
Abbildung 42: Definition des Koordinatensystems eines GML-Objekts (eigene Darstellung).....	57
Abbildung 43: KML Code "RoadSegment" (eigene Darstellung).....	59
Abbildung 44: Externe graphische Stilisierung von KML-Objekten (eigene Darstellung).....	60
Abbildung 45: Syntax der KML-Prefix <gx:AnimatedUpdate> ([KML-o.J.]).....	60
Abbildung 46: TimeSpan und TimeStamp Syntax in KML ([KML-o.J.]).....	61
Abbildung 47: GeoJSON Datenstruktur (eigene Darstellung).....	62
Abbildung 48: GeoJSON Stildefinition nach der <i>simplestyle</i> Spezifikation ([SIM-17]).....	63
Abbildung 49: GeoJSON kategorisierte graphische Aufbereitung (eigene Darstellung).....	63
Abbildung 50: CSS Datei zur Stilisierung der GeoJSON Objekte (eigene Darstellung).....	63
Abbildung 51: SVG Code Ellipse ([SVG-18]).....	65
Abbildung 52: Identer path id von SVG-Elementen (eigene Darstellung).....	66

Abbildung 53: Beispiel eines Textfiles eines Punktdatensatz ([GLO-18a: 93]).....	69
Abbildung 54: OmniSuite Punktformat am sphärischen Display ([GLO-18a: 88])	69
Abbildung 55: CSS-Datei eines Punktdatensatzes (eigene Darstellung).....	70
Abbildung 56: Ausgangsdatsätze der praktischen Formatevaluation ([GLO-18b]).....	66
Abbildung 57: EarthByNight Plattkarte ([EAR-17]).....	67
Abbildung 58: Tabelle des aufbereiteten Datensatzes (Finale Form) (eigene Darstellung)	69
Abbildung 59: CSV-Export in Libre Office (eigene Darstellung).....	69
Abbildung 60: Ausschnitt der *.txt Datei (eigene Darstellung)	69
Abbildung 61: Ausschnitt einer Kategorie des CSS-Dokuments der *.txt Datei (eigene Darstellung)	70
Abbildung 62: Import der *.txt Datei in OmniSuite (eigene Darstellung)	71
Abbildung 63: *.txt Dateivisualisierung in OmniSuite (eigene Darstellung).....	71
Abbildung 64: Export der Polygone der Flood Events im Jahr 2017 (Datengrundlage: [GLO-18b])	72
Abbildung 65: Überlappende Polygone Monat Mai 2017 ([GLO-18b])	72
Abbildung 66: ArcGIS Pro Rasterexport Einstellungen (eigene Darstellung)	73
Abbildung 67: Rasterbild ohne/mit Transparenz (eigene Darstellung)	73
Abbildung 68: Polygone aufbereitet in Layer-Ebenen (eigene Darstellung).....	74
Abbildung 69: Animation der einzelnen Polygone in Photoshop (eigene Darstellung)	75
Abbildung 70: Render-Einstellung in Photoshop (eigene Darstellung).....	76
Abbildung 71: DDS-Konvertierung in OmniSuite (eigene Darstellung).....	76
Abbildung 72: Rasterimport in OmniSuite (eigene Darstellung)	76
Abbildung 73: Legende der Global Story (eigene Darstellung)	77
Abbildung 74: Definition der Legendenausdehnung in OmniSuite (eigene Darstellung)	77
Abbildung 75: Erstellung der Global Story im StoryEditor (eigene Darstellung).....	78
Abbildung 76: Global Story zu verschiedenen Zeitpunkten (eigene Darstellung)	78
Abbildung 77: Ausschnitt aus der KML-Datei des Testdatensatzes (eigene Darstellung)	81
Abbildung 78: KML Style-Definition im Testdatensatz nach Konvertierung (eigene Darstellung)	81
Abbildung 79: KML 8-Digit Farbcode im Vergleich ([KML-12])	81
Abbildung 80: <styleURL> im KML Objektcode (eigene Darstellung)	82

Abbildung 81: Klassifiziertes KML-File (exportiert aus QGIS/visualisiert in Google Earth)	83
Abbildung 82: KML externe Stildefinition (eigene Darstellung).....	83
Abbildung 83: Definition zum Ein- und Ausblenden von KML-Objekten (Datengrundlage: [KML-o.J.])	84
Abbildung 84: FadeIn von KML-Objekt in Google Earth Pro (eigene Darstellung).....	84
Abbildung 85: FadeIn durch Zoom in Google Earth Pro (eigene Darstellung)	85
Abbildung 86: Zeit- und Datumseinstellungen Windows 10 (eigene Darstellung)	85
Abbildung 87: KML Objektanimation in Google Earth Pro (eigene Darstellung)	85
Abbildung 88: C++ KML Parsing Beispiel ([LIB-08]).....	87
Abbildung 89: Ausschnitt aus der SVG-Datei des Testdatensatzes (eigene Darstellung)	87
Abbildung 90: Ausschnitt aus der GEOJSON-Datei des Testdatensatzes (eigene Darstellung)	88
Abbildung 91: Kartographische Aufbereitung von GeoJSON per Geoobjekt (eigene Darstellung)	89
Abbildung 92: Gruppierung von GeoJSON Geoobjekten (eigene Darstellung).....	89
Abbildung 93: CSS File zur Stilisierung der gruppierten Geoobjekte (eigene Darstellung)	90
Abbildung 94: C++ RapidJSON Example ([RAP-o.J.]).....	91
Abbildung 95: Flutpolygone in Mapbox Studio Classic (eigene Darstellung)	92
Abbildung 96: Variation von Geoobjekten in Mapbox Studio Classic mit CartoCSS (eigene Darstellung)	92
Abbildung 97: Ausschnitt aus dem Quellcode der Mapbox WebMap (Datengrundlage: [ADD-o.J.])	93
Abbildung 98: Interne CSS-Stilisierung einer HTML Mapbox Datei (Datengrundlage: [ADD-o.J.]).....	93
Abbildung 99: GeoJSON Objekte des Testdatensatzes in einer Mapbox Webmap (eigene Darstellung)	94
Abbildung 100: Data-Driven Styling in der Mapbox Webmap (Datengrundlage: [ADD-o.J.]).....	94
Abbildung 101: Attributbasierte Kategorisierung von Flood-Polygonen mit Mapbox (eigene Darstellung) ...	95
Abbildung 102: Definition eines FadeIn von Flood-Objekten in Mapbox GL JS (eigene Darstellung).....	95
Abbildung 103: Sichtbarkeit der Flood-Polygone zur Sekunde 0 und 3 (eigene Darstellung)	95
Abbildung 104: Slider-basierte Animation in Mapbox GL JS (eigene Darstellung).....	96

Tabellenverzeichnis

Tabelle 1: Globenkategorien ([RIE-11: 153])	5
Tabelle 2: Punktoobjekte am Hyperglobus ([OPT-16: 4])	10
Tabelle 3: Schrift am Hyperglobus ([OPT-16: 2])	12
Tabelle 4: Auszug aus Themenbereichen für den Hyperglobus ([RIE-00: 88-89])	16
Tabelle 5: Skalenniveaus ([LAN-13: 165],[HOR-04: 71])	27
Tabelle 6: Rasterformat vs. Vektorformat ([LAN-13: 336])	28
Tabelle 7: Vor- und Nachteile für Dateiformate in OmniSuite (eigene Darstellung)	29
Tabelle 8: Vergleich der Visualisierungsmethoden am Hyperglobus (eigene Darstellung)	35
Tabelle 9: LineString/MultiLineString nach ISO 90107 ([OGC-10: 23-25])	41
Tabelle 10: Polygon/MultiPolygon nach ISO 90107 ([OGC-10: 27-32])	42
Tabelle 11: Auswahl räumlicher Beziehungen/Analysen nach ISO 90107 ([OGC-10: 36-40])	43
Tabelle 12: OGR Vektorformate konform der Simple Feature Spezifikation (blau hinterlegt) ([OGR-o.J.]) ..	44
Tabelle 13: Vektorformate auf der Plattform data.gv.at ([DAT-o.J.])	45
Tabelle 14: Vektorformate auf der Plattform datacatalog.worldbank.org ([DAT-18])	45
Tabelle 15: Vor- und Nachteile von XML ([EVJ-07: 5])	48
Tabelle 16: WKT Struktur verschiedener Features ([OGC-10: 61])	67
Tabelle 17: OmniSuite Punktdatensatz Auszug ([GLO-18a: 89])	68
Tabelle 18: Vektordatenkonvertierung in ArcGIS (Pro) und QGIS (eigene Darstellung)	71
Tabelle 19: Vergleich der analysierten Vektorformate (eigene Darstellung)	64
Tabelle 20: Tabelle des Datensatzes (Ausgangsform) (eigene Darstellung)	67
Tabelle 21: Kategorien der Überflutungsursachen (eigene Darstellung)	68
Tabelle 22: Tabelle mit berechneten Frames per Second (eigene Darstellung)	75
Tabelle 23: Aspekte der Umsetzung der Global Story (eigene Darstellung)	79
Tabelle 24: Praktische Formatanalyse der Vektorformate (eigene Darstellung)	97

Formelverzeichnis

Formel 1: Formel zur Berechnung der Breite eines Objektes ([PUN-17: 1])	15
Formel 2: Formel zur Berechnung der Animationszeiten ([GLO-18a: 92])	68
Formel 3: Formel zur Berechnung der Polygonanimation	75

Abkürzungsverzeichnis

BSD	Berkeley Software Distribution
CSG	Constructive Solid Geometry
CSS	Cascadian Style Sheet
DDS	Direct Draw Surface
DFX	Drawing Interchange Format
DOM	Dokumentobjektmodell
DXT	S3 Texture Compression
GeoJSON	Geo JavaScript Object Notation
GML	Geography Markup Language
GIS	Geoinformationssystem
FME	Feature Manipulation Engine
HG	Hyperglobus
HRG	Hyperglobe Research Group
JPEG	Joint Photographic Expert Group
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
OGC	Open Geospatial Consortium
OGR	OpenGIS Simple Features Reference Implementation Simple Features Library
OpenGL	Open Graphics Library
PNG	Portable Network Graphics
RGB	Rot-Grün-Blau
SAX	Simple API for XML
SHP	Shapefile
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics
STL	C++ Standard Template Library
WKT	Well-Known Text
WWW	World Wide Web

Kurzfassung

Als sphärisches Display dient der taktile Hyperglobus zur realitätsnahen Informationswiedergabe räumlicher Information. Durch Projektionsverfahren im Inneren der Kugel werden verschiedene globale Thematiken und Phänomene auf diesem visualisiert. Der Umsetzung einer Global Story liegen jedoch arbeitsintensive Prozesse der Datenbeschaffung, -Aufarbeitung und- Visualisierung zu Grunde. Formate, welche hierfür verwendet werden sind Raster- und Vektordaten.

Die vorliegende Masterarbeit untersucht geeignete Vektorformate für eine geplante, direkte Visualisierung am Hyperglobus. Zum momentanen Forschungsstand werden globale Vektordaten in einem Geoinformationssystem aufbereitet, in einem Bildbearbeitungsprogramm entzerrt und als Platkarte im Rasterformat in die Globensoftware OmniSuite importiert. Dieser Prozess ist mit einem hohen Arbeitsaufwand verbunden. Zusätzlich ist der Speicherbedarf durch die gerenderten Bilddaten enorm.

Die direkte Implementierung von Vektordaten im Punktformat wurde bereits entwickelt und in OmniSuite realisiert. Ziel dieser Arbeit ist es, gängige Vektorformate, speziell für Linien- und Polygondaten, zu evaluieren und eine Auswahl für eine praktische Analyse zu treffen. In dieser werden die Vektorformate hinsichtlich ihrer Vor- und Nachteile, bezogen auf ihre Geometrie, Raumbezug, kartographische Stilisierung, Animation, Interaktion, Formatbibliotheken in C++ und den Speicherbedarf untersucht.

Basierend auf dieser Analyse wird eine Empfehlung für (ein) Vektorformat(e) gegeben, welche in der Zukunft für die Entwicklung einer direkten Visualisierung am sphärischen Display herangezogen werden können.

Abstract

As a spherical display, the tactile hyperglobe is used for the reproduction of realistic information of spatial data. Through projection methods inside the spherical display, various global topics and phenomena are implemented and visualized. The implementation of a global story, however, is based on labor-intensive processes of data acquisition, processing and visualization. Formats used therefore are raster and vector.

This thesis investigates suitable vector formats for a planned, direct visualization on the hyperglobe. To the current state of research, global vector datasets are processed in a geoinformation system, rectified in an image processing program and imported as a flat map in raster format into the globe software OmniSuite. This process includes a lot of work. In addition, the memory required by the rendered image data is enormous.

The direct implementation of global vector data for point datasets has already been developed and realized in OmniSuite. The aim of this work is to evaluate common vector formats, especially for line and polygon data, and to select formats for a practical analysis. There, the vector formats will be examined regarding their advantages and disadvantages in terms of their geometry, spatial reference, cartographic styling, animation, interaction, format libraries in C++ and the memory requirements.

Based on this analysis, a recommendation for a (one) vector format(s) is given. In the future, the format(s) can be used to develop a direct implementation for line and polygon datasets on the spherical display.

Vorwort

An dieser Stelle möchte ich all jenen danken, die mich während meiner Studienzeit und der Umsetzung dieser Masterarbeit unterstützt haben.

Zuerst bei meiner Familie, weil Ihr mich dabei unterstützt habt, durch mein Studium einen neuen Lebensabschnitt in Wien zu beginnen.

Bei meinen Freunden und Unikollegen, welche die Universitäts- und Studentenzeit zu einer unvergesslichen Zeit gemacht haben. Danke Lara, dass wir gemeinsam auch in harten Prüfungsphasen nie den Humor verloren haben. Danke Renée und Katrin, dass wir immer über verschiedene Bereiche unseres Studiums diskutieren und so immer eine neue Begeisterung für dieses entwickeln konnten.

Besonders danken möchte ich auch Ass.-Prof. Mag. Dr. Andreas Riedl und Mag. Jürgen Kristen. Danke für Eure Zeit, die konstruktive Kritik und Inputs während der Entstehung dieser Masterarbeit. Ich konnte mich bei Fragen immer an Euch wenden und bin dafür sehr dankbar!

Bedanken möchte ich mich zuletzt auch bei den Korrekturlesern dieser Masterarbeit.

Ihr alle habt dies möglich gemacht. Danke Euch dafür!

1 Einleitung

Der Wissensvermittlung globaler Phänomene wird in der modernen Welt eine immer wichtigere Rolle zu Teil. Als Medium, oder besser gesagt „Erdkugel“ zur realitätsnahen Informationswiedergabe von räumlicher Information dient der Hyperglobus, ein sphärisches Display, welcher seit dem Jahr 2005 von der Hyperglobe Research Group (HRG) erforscht und entwickelt wird. Durch diese Möglichkeit der Präsentation können globale Zusammenhänge wie auch Phänomene global realisiert und visualisiert werden. Doch dem Prozess der Geovisualisierung am Hyperglobus liegen aufwendige Wege der Datenbeschaffung, Datenaufbereitung und Darstellung am sphärischen Display zu Grunde.

Diese Masterarbeit beschäftigt sich daher mit einem Datenformat, welches für den Hyperglobus eine wichtige Quelle zur Visualisierung globaler räumlicher Information ist: den Vektorformaten. In den nächsten Unterkapiteln wird auf die Relevanz dieses Formates im Rahmen der Geovisualisierung eingegangen und es werden die Problemstellungen wie auch Forschungsfragen der Masterarbeit vorgestellt. Dadurch soll ein Überblick über die Herangehensweise wie auch Ziele dieser Forschungsarbeit entstehen.

1.1 Problemstellung

Plattkarten im Rasterformat, welche von Vektordaten abgeleitet wurden, sind nicht fehlerfrei zur Darstellung thematischer Inhalte am Globus geeignet. Dies liegt an der zunehmenden Verzerrung mit ansteigender geographischer Breite. Um eine korrekte Darstellung am Hyperglobus (HG) zu erreichen, muss die Verzerrung in Plattkarten berücksichtigt werden. Wird dies nicht beachtet, werden Inhalte am sphärischen Display fehlerhaft dargestellt. Im breiten Spektrum der Vektorformate im Bereich räumlicher Geoinformationsvisualisierung, und im speziellen jenem des sphärischen Displays, spielt somit die Wiedergabe von Punkt-, Linien- und Flächengeometrien eine entscheidende Rolle. (vgl. [LIE-12: 2-3]) Verschiedene Träger von Geometrie werden zur Visualisierung von räumlicher Geoinformation herangezogen, zu verbreiteten Formaten dieser Art zählen Shapefiles, wie auch KML (Keyhole Markup Language) - oder SVG (Scalable Vector Graphics) - Dateien.

Doch auch andere Vektorformate rücken immer mehr in den Fokus. Aufgrund der programmunabhängigen Datenstruktur werden Formate wie (Geo)JSON (Geo JavaScript Object Notation), durch ihr ideales, programmunabhängiges Austauschformat, als weitere, vielseitig einsetzbare Vektorformate in der Geoinformation gehandelt.

Da Vektorformate im aufstrebenden Gebiet der Multimediatechnologie und Geokommunikation das zentrale Format zur globalen Wissensvermittlung darstellen, müssen geeignete Vektorformat(e) festgelegt werden, um den Prozess der direkten räumlichen Geometrievisualisierung auf dem Globenmedium der Zukunft zu optimieren.

1.2 Zielsetzung

Wie bereits im vorherigen Unterkapitel erläutert, ist das Ziel dieser Masterarbeit Vektorformate, hinsichtlich ihrer Vor- und Nachteile, für eine Visualisierung am sphärischen Display zu evaluieren. Diese zählen in der Geoinformation, neben Rasterdaten, zu den Hauptträgern räumlicher Information. Gewählte Vektorformate werden, basierend auf einer praktischen Formatanalyse, für eine zukünftige Visualisierung am Hyperglobus bewertet.

1.3 Fragestellung

Aus der erläuterten Problematik, Relevanz und Zielsetzung dieser Masterarbeit ergibt sich die folgende Forschungsfrage, welche durch folgend definierte Arbeitsfragen bearbeitet und untersucht werden soll:

Welche Vektorformate eignen sich anhand ihrer Vor- und Nachteile für eine Visualisierung am sphärischen Display?

- Welche Features sind für eine Visualisierung am sphärischen Display **essenziell**?
- Wie werden **kartographische** Stilinformationen im jeweiligen Format mitgespeichert?
- Welche **Animations- und Interaktionsmöglichkeiten** bieten sich im Rahmen der Vektorformate?
- Werden die Vektorformate von **Programmen unterstützt** und gibt es (open-source) **Programmbibliotheken** in Programmiersprachen für diese?
- Welche Unterschiede gibt es im **Speicherbedarf** der jeweiligen Formate?
- Welche Vektorformate **eignen** sich und können diese Features **mitspeichern**?

1.4 Methodik

Im Rahmen einer Literaturrecherche wird im ersten Teil der Masterarbeit kurz eine Einführung in sphärische Displays, die Verzerrungsproblematik auf jenen und die Möglichkeiten der Geovisualisierung gegeben. Zusätzlich werden auch die momentan bestehenden drei Methoden der Geovisualisierung am Hyperglobus beleuchtet. Auch wird durch die Literaturanalyse untersucht, welche Vektorformate in der Geoinformation relevant sind und im Rahmen der praktischen Evaluation untersucht werden sollen.

In einem zweiten Schritt findet die praktische Evaluierung und Analyse der verschiedenen Formattypen, sowohl offen als auch kommerziell, statt. Diese Vektorformate werden auf Basis ihrer Features, welche für eine Visualisierung am sphärischen Display essenziell sind, ihrer kartographischen Stilinformationen und weiterer, in Kapitel 1.2 definierter Arbeitsfragen, untersucht und gegenübergestellt. Basierend auf den Ergebnissen der Evaluation wird ein Werturteil der untersuchten Formate gegeben.

In einem letzten Schritt werden die Vektorformate, welche sich im Rahmen der Untersuchung als geeignet für eine Darstellung erwiesen haben, für eine zukünftige Anwendung und Visualisierung am sphärischen Display geprüft. Diese Untersuchung wird anhand einer theoretischen und praktischen Formatanalyse durchgeführt.

1.5 Motivation

Die persönliche Motivation zur Bearbeitung dieser Fragestellung im Rahmen dieser Masterarbeit liegt am großen Interesse am Hyperglobus und der Darstellung globaler Thematiken. Dieses wurde durch die Teilnahme an der Lehrveranstaltung „Multimediatechnologie und Geokommunikation“ geweckt. Da es im Rahmen der Umsetzung des Semesterprojekts „Tiefseekabel – Die Welt hängt am Draht“ Probleme bei der Integration der Vektordaten in diverse Bildbearbeitungsprogramme gab, wurde der Anreiz geweckt, die eigene Masterarbeit in diesem Bereich zu verfassen.

1.6 Aufbau der Arbeit

Der Aufbau der Masterarbeit gliedert sich in zehn Abschnitte. Kapitel 1 gibt eine generelle Einführung in die Problemstellung, Zielsetzung und Motivation dieser Masterarbeit.

Die Geovisualisierung am sphärischen Display ist Thema des zweiten Kapitels. Zum Einstieg wird kurz die Entstehungsgeschichte von sphärischen Displays behandelt. Darauf folgend wird geklärt, welche Anforderungen für eine Visualisierung am Hyperglobus erfüllt werden müssen.

In Kapitel 3 werden Raster- und Vektordaten, welche momentan für eine Visualisierung am sphärischen Display verwendet werden, untersucht. Am Ende wird ein Überblick über die Vor- und Nachteile der Formate für eine Visualisierung am sphärischen Display gegeben.

Verschiedene Methoden der Geovisualisierung am sphärischen Display werden in Kapitel 4 beleuchtet. Dabei werden die direkte-, indirekte Raster- wie auch direkte Vektorvisualisierung am sphärischen Display erläutert. Durch einen Vergleich der Methoden sollen die Vor- und Nachteile der jeweiligen Prozesse verdeutlicht werden.

Um eine Grundlage für den praktischen Teil der Masterarbeit zu bilden, werden zusätzlich umgesetzte Global Stories in Kapitel 5 analysiert und hinsichtlich eingesetzter Features untersucht.

Kapitel 6 beschäftigt sich mit der weiten Breite an Vektorformaten, welche in der Geoinformation Einsatz finden. Es wird geklärt, welche Anforderungen ein Vektorformat nach der Simple Feature Specification besitzen muss, um als ein geographisches Vektorformat zu gelten. Um eine Auswahl von gängigen Vektorformaten zu treffen, werden eine internationale und österreichische Datenplattform auf das Vorkommen jener untersucht. Basierend darauf wird eine Auswahl an Vektorformaten für die theoretische Formatanalyse in Kapitel 7 gebildet.

In dieser wird anhand diverser Vorgaben untersucht, ob die gewählten Formate für die praktische Analyse in Kapitel 8 geeignet sind. Das Kapitel wird mit einer Übersichtstabelle, welche die Vor- und Nachteile der theoretischen Formatanalyse abbildet, geschlossen.

Nach der Auswahl von Vektorformaten in Kapitel 7, werden diese in Kapitel 8 praktisch untersucht. Um eine Grundlage für die Untersuchung zu bilden, wird zuerst eine Global Story nach dem momentanen Ansatz der indirekten Raster- und direkten Vektorvisualisierung umgesetzt. Durch die Dokumentation der Arbeitsschritte soll die Bedeutsamkeit einer zukünftigen, direkten Vektorvisualisierung von Linien- und Polygondaten am sphärischen Display hervorgehoben werden. Die praktische Analyse beschäftigt sich mit Aspekten der kartographischen Stilisierung, Animation, Interaktion und Formatbibliotheken der jeweiligen Vektorformate in C++. Basierend auf dieser praktischen Analyse werden die Vor- und Nachteile einer direkten Implementierung der finalen Vektorformate unterstrichen. Am Ende des Kapitels steht eine Formatempfehlung für die zukünftige Implementierung von Vektordaten durch die Globensoftware am Hyperglobus.

In den Kapiteln 9 und 10 werden die Erkenntnisse dieser Masterarbeit diskutiert und zusammengefasst. Es gibt zusätzlich einen Ausblick auf Vorteile, welche sich durch die direkte Implementierung von Vektorformaten am sphärischen Display ergeben.

2 Geovisualisierung am sphärischen Display

Globen, ob analog oder digital, geben ein dreidimensionales reales Abbild der Welt wieder. Sie sind die einzige kartographische Darstellungsform, welche originäre, dreidimensionale Körpergestalten verzerrungsfrei abbilden. (vgl. [RIE-00: 16])

„Ein Globenkörper präsentiert ein maßstabsgebundenes und strukturiertes Modell eines Himmelskörpers (bzw. der scheinbaren Himmelskugel) in seiner unverzerrten dreidimensionalen Ganzheit“ ([RIE-00: 17])

Dem sphärischen Display, auch Hyperglobus genannt, wie er heute besteht, liegt eine lange Entwicklungsphase zugrunde. Das folgende Kapitel möchte daher beleuchten, seit wann sphärische Displays in der Familie der Globen einen Platz einnehmen und welche Entwicklungen es im Bereich von jenen gab.

Da es sich bei vorliegender Masterarbeit um eine Thematik für ein taktilen sphärisches Display (Hyperglobus) handelt, werden analoge Globen und deren Entwicklungsgeschichte nicht näher beleuchtet. Für nähere Informationen zu diesem Themenbereich wird auf Kapitel 2 der Dissertation *“Virtuelle Globen in der Geovisualisierung – Untersuchungen zum Einsatz von Multimediatechniken in der Geopräsentation”* von Ass.-Prof. Mag. Dr. Andreas Riedl verwiesen.

2.1 Sphärische Displays in der Globenfamilie

Im Rahmen technologischer Weiterentwicklungen der letzten Jahrzehnte ist der Globus, nach einer mehr als 500-jährigen Geschichte im Bereich analoger Globen, in eine neue Entwicklungsetappe eingetreten, auch die „Renaissance“ der Globen genannt. Auch in jenem Bereich digitaler Globen gibt es Differenzierungen hinsichtlich ihrer Beschaffenheit, der räumlichen Ausprägung und der Art des Globenbildes. (vgl. [RIE-10: 153])

So lassen sich durch die genannten Parameter nach Riedl A. die folgenden drei Kategorien digitaler Globen festlegen. Der Begriff „Hyper“ bei jenen Globenarten bezieht sich auf den Begriff Hypermedia, welcher ein interaktives und vernetztes Informationsmedium beschreibt. (vgl. [RIE-11: 153])

	Abbild	Raum	Globenkörper
Analoger Globus	analog	real	materiell
Virtueller Hyperglobus	digital	virtuell	virtuell
Taktiker Hyperglobus	digital	real	materiell
Hologlobus	digital	real	virtuell

Tabelle 1: Globenkategorien ([RIE-11: 153])

2.1.1 Virtueller Hyperglobus

Unter jener Globenart versteht man digitale Globenbilder, welche globale Information auf einem nicht materiellen Globenkörper im virtuellen Raum visualisieren. Bereits im Jahr 1992 gab es das erste Produkt, dabei handelte es sich um eine digitale Form des Behaim Erdglobus, der ältesten erhaltenen Erddarstellung in Kugelform. Die digitale Version des Behaim-Globus wurde im Rahmen des 500. Geburtstages der analogen Version als Reproduktion des Originals umgesetzt. (vgl. [DOR-96: 20])

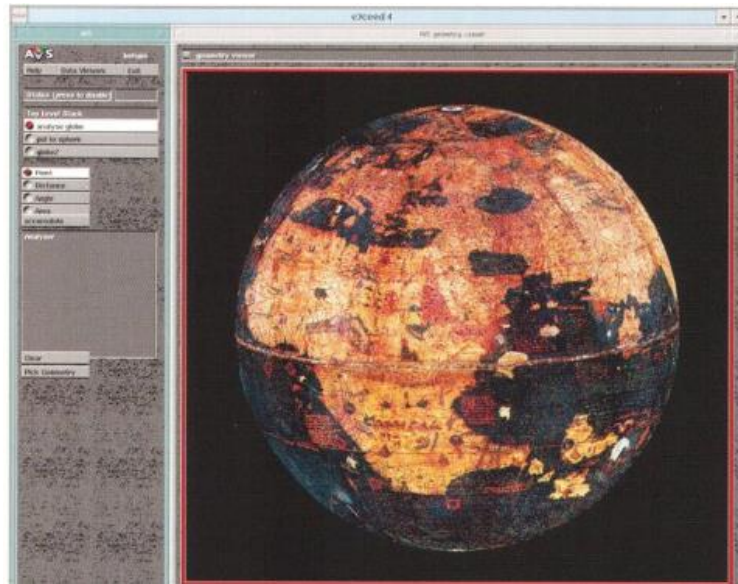


Abbildung 1: Digitaler Behaim-Globus ([DOR-96: 22])

Im Zuge der Digitalisierung des analogen Behaim-Globus wurde ein 3D-Modell erstellt. Dadurch konnte der digitale Globus in Echtzeit am Monitor verschoben, vergrößert, verkleinert und auch gedreht werden. Weitere Funktionen waren die Generalisierung der dargestellten Geoinformation je nach Zoomstufe, die digitale Vermessung des Globus oder auch der Vergleich verschiedener Globen oder Kartenwerke durch Ebenen. Dies bot Nutzerinnen und Nutzern eine Möglichkeit der Interaktivität, welche beim Einsatz analoger Globen in dieser Form nicht gegeben ist. Somit ist der Behaim-Globus als wichtiger Meilenstein in der Entwicklungsgeschichte digitaler sphärischer Displays zu handeln. (vgl. [DOR-96: 22-24])

Als weiterer Vorreiter im Bereich virtueller Hypergloben ist „Planet Earth“ zu nennen. Hierbei handelt es sich um den ersten digitalen Hyperglobus, welcher Real-Time Daten visualisiert hat. Umgesetzt im Jahr 1996 wurden auf diesem aktuelle, globale Wetterdaten digitalisiert. Bis heute wurde „Planet Earth“ mit vielen weiteren Funktionen erweitert und in „Earth Browser“ umbenannt. (vgl. [RIE-99: 260])



Abbildung 2: "Earth Browser" ("Planet Earth") ([DIG-08])

In seiner Dissertation hat sich Ass.-Prof. Mag. Dr. Andreas Riedl, Leiter der Hyperglobe Research Group (HRG), mit digitalen Globen, der Verwendung von Multimediatechnologie und der Vorteile dieser Darstellungsform von raumbezogenen Sachverhalten gegenüber analogen Globen auseinandergesetzt.

Im Rahmen dieser wissenschaftlichen Forschung wurde der „Hyperglobus“ entwickelt. Als zentraler Wissensvermittler globaler Thematiken können diverse globale Fragestellungen auf ihm visualisiert und beantwortet werden. (vgl. [RIE-11: 155])

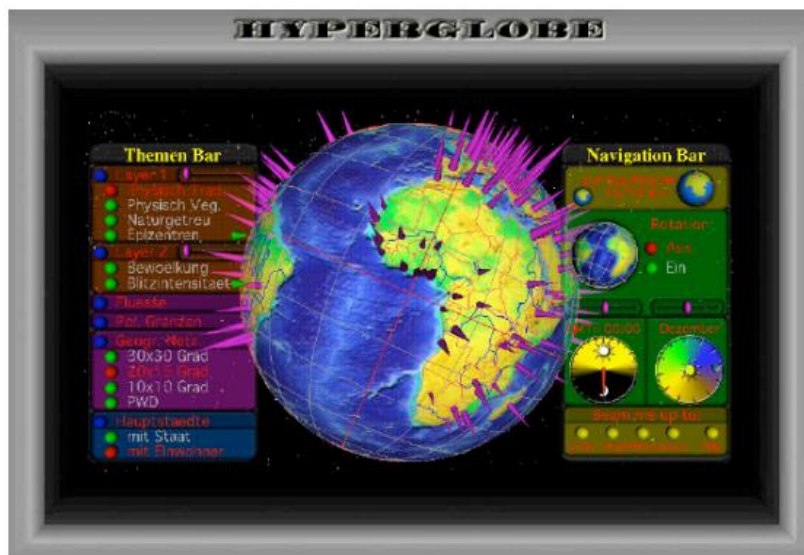


Abbildung 3: Hyperglobus ([RIE-11: 155])

2.1.2 Taktiler Hyperglobus

Im Gegensatz zu virtuellen Hypergloben werden taktile sphärische Displays als Nachkommen analoger Globen angesehen. Hierbei wird einem realen Globenkörper ein digitales (Globen)Bild aufprojiziert. Durch die Beschaffenheit und die Bildschirmgeometrie im Raum wird die dreidimensionale Wahrnehmung erzeugt. Eine weitere Gemeinsamkeit taktiler und analoger Globen ist, dass die Kugel des sphärischen Displays sowohl zur Visualisierung als auch als Globenmedium genutzt wird. Der Begriff „taktil“ bezieht sich im Namen der Hyperglobenart darauf, dass die Möglichkeit der Berührung des Globenmediums besteht. (vgl. [RIE-11: 156-157])

Die ersten Vorreiter und Entwicklungen taktiler Hypergloben gab es schon Anfang der 1990er, wobei es sich hier meist einerseits um Entwicklungsergebnisse zwischen virtuellem und taktilem Globus oder um Kombinationen aus diesen handelte. Seit dem Wendejahr 2002 erfuhr der taktile Hyperglobus hingegen einen großen Aufschwung. So wurden mehrere Varianten des Globenmediums entwickelt, wobei sich diese durch jeweils eigene patentierte Projektionsverfahren unterscheiden. Insgesamt sind seit dem Jahr 2002 mehr als 250 taktile einsatzbereite Hypergloben in den Umlauf gekommen, davon im Jahr 2009 alleine fast ein Drittel. (vgl. [RIE-11: 157-164])

Generell wird taktile Hypergloben, aufgrund ihrer realen Existenz und der Vielfältigkeit ihrer globalen Darstellungs- und Visualisierungsmöglichkeiten, ein großes Potential zugeschrieben. Jedoch befinden sich Anschaffungskosten der taktile sphärischen Displays in einem hohen Bereich befinden und es bedarf eines fachkundigen Personals zur Instandhaltung des sphärischen Displays. Daher sind taktile Hypergloben momentan mehrheitlich in Forschungsstätten, wie dem HRG Labor, oder auch Museen und Schauräumen vorzufinden. In Anbetracht der Digitalisierung und mit fortschreitender Entwicklung im Bereich der Größe, der Materialkosten wie auch der Instandhaltung jener werden taktile Hypergloben in der Zukunft Stück für Stück analoge Globen, auch im privaten Gebrauch, ersetzen. (vgl. [WIN-12: 8])



Abbildung 4: Taktiler Hyperglobus ([DGF-12])

Für detailliertere Informationen bezüglich der Vorteile und Eigenschaften des taktilen Hyperglobus wird auf Kapitel 2.2 in der Diplomarbeit von Mag. rer. nat. Sebastian Wintner mit dem Arbeitstitel „*Digital Storytelling angewandt auf einen taktilen Hyperglobus*“ verwiesen.

2.2 Anforderungen an eine Visualisierung am sphärischen Display

Um eine Visualisierung von globalen Inhalten am sphärischen Display umzusetzen, gilt es mehrere wichtige Aspekte zu beachten. Auch wenn die Datengrundlage alle Voraussetzungen für eine Visualisierung am Hyperglobus erfüllt, spielt die Qualität der Darstellung und Umsetzung für Endnutzer die zentrale Rolle. Die Qualität einer Darstellung am Hyperglobus ergibt sich aus den folgenden, zentralen Elementen:

- **Kartographische Gestaltung**
- **Verzerrungsproblematik**
- **Globenwürdigkeit**
- **Thematik**
- **Animation**
- **Interaktion**

Die folgenden Unterkapitel beschäftigen sich nun mit den eben genannten Elementen und Anforderungen, welche es für das sphärische Display umzusetzen gilt.

2.2.1 Kartographische Gestaltung

„Als kartographische Gestaltungsmittel gelten die Grundelemente Punkt, Linie und Fläche sowie die zusammengesetzten Zeichen Signatur, Diagramm, Halbton und Schrift. Für diese Zeichen und ihre graphischen Variationen (...) geht es um eine allgemeine systematische Zuordnung zu den Objektmerkmalen (...), die sich mit den Zeichen jeweils darstellen lassen.“ ([HAK-02:118])

Wie bei analogen oder digitalen Karten handelt es sich beim Hyperglobus um ein digitales, dreidimensionales kartographisches Produkt. Daher müssen Visualisierungen im Zuge von Global Stories den Grundregeln der kartographischen Gestaltung folgen und dementsprechend graphisch aufbereitet werden. Nach Hake zählen zu jenen Elementen Punkt, Linie, Fläche und daraus folgend Signaturen, Diagramme als auch Schrift. Jene Objekte lassen sich durch differenzierte Darstellungen nach Aspekten der Quantität und Qualität visualisieren. (vgl. [HAK-02: 107]) Dabei lassen sich Objektmerkmale durch die folgenden Variablen beschreiben:

- **Größe:** Durch Bewertungen wie Betonung oder Abschwächung (*quantitativ*)
- **Form:** Unterschiede der *Qualität* (bild- und symbolhafte Zeichen)
- **Füllung:** *Qualitative* (Farbe) und *quantitative* (z.B. Intensität) Unterschiede

- **Tonwert:** *Quantitative* Unterschiede durch Helligkeitsstufen (Dichtewerte)
- **Richtung (Orientierung):** Hinweis auf *zeitliches Verhalten* wie die Gliederung von Merkmalen
- **Farbe:** *Qualität* (Farbton) und *Quantität* (Farbsättigung) (vgl. [HAK-02:108])

2.2.2 Punkt

Punkte können, basierend auf den in Kapitel 2.2.1 genannten Variablen, visualisiert werden. Qualitative Informationen werden bei Punktobjekten durch Unterschiede in der Farbgebung visualisiert, quantitativ durch die Variation der Objektgrößen. Um eine gute Lesbarkeit am Globenmedium zu garantieren, muss eine gewisse Mindestgröße von Punktobjekten am Hyperglobus eingehalten werden. Dieser ist bei analogen kartographischen Produkten mit mindestens 0,3 mm angegeben. (vgl. [HAK-02: 118]) Da es sich beim Hyperglobus um eine andere Größenordnung handelt, ändert sich die Mindestgröße der Punktgeometrien. Im Rahmen der Lehrveranstaltung „Multimediavisualisierung und Geokommunikation“ haben sich Studentengruppen mit optimalen Darstellungen am sphärischen Display beschäftigt und die Größe einfärbiger Kreise und Rechtecke am Hyperglobus untersucht. Tabelle 2 gibt einen Überblick über gewonnene Erkenntnisse.

Kreis:

	Weißer Hintergrund		Bunter Hintergrund	
	min	opt	min	opt
Schwarz gefüllt - Größe	4-5 px	ab 10 px	5 px	ab 12 px
Schwarz Linie - Größe	4 px	ab 7 px	4 px	ab 8 px
Schwarz Linie - Linienstärke	0.2 px	0.3 – 1.0 px	0.5 px	0.8 – 1.4 px
Farbig gefüllt - Größe	4 px	ab 8 px	5 px	ab 8 px
Farbig Linie - Linienstärke	0.2 px	0.3 – 1.0 px	0.5 px	0.8 – 1.4 px

Rechteck:

	Weißer Hintergrund		Bunter Hintergrund	
	min	opt	min	opt
Schwarz gefüllt - Größe	5 px	ab 8 px	5 px	ab 10 px
Schwarz Linie - Größe	5 px	ab 7 px	5 px	ab 9 px
Schwarz Linie - Linienstärke	0.1 px	ab 0.5 px	0.6 px	ab 0.9 px

Tabelle 2: Punktobjekte am Hyperglobus ([OPT-16: 4])

2.2.2.1 Linie

Die graphische Variabilität von Linien (nicht unterbrochene Striche) ist durch ihre Farbwahl und ihre Linienbreite zu erreichen. Durch diese Variationen ist es möglich, anhand von

Linien quantitative und qualitative Aussagen zu treffen. Es ist die Abgrenzung von diskreten Objekten wie auch die Verbindung identer Objekte (Kontinuum, z.B. Höhenlinie) möglich. Als zusätzliche Gestaltungsmittel für die Qualität der Datenaussage nennt Hake unter anderem Schrift und Signatur. (vgl. [HAK-02: 120])

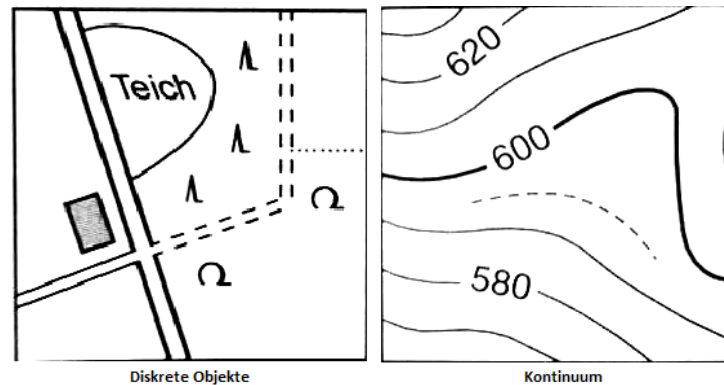


Abbildung 5: Linienobjekte ([HAK-02: 120])

2.2.2.2 Fläche

Flächen besitzen in ihrer Ausdehnung einen konstanten Farbton und Helligkeit. Durch die Variation der soeben genannten Flächenattribute kann eine graphische Variation und Differenzierung von Flächen vorgenommen werden. Hingegen lassen sich quantitative flächenhafte Darstellungen durch die Variation des Tonwerts (z.B. Flächenkartogramm) umsetzen. (vgl. [HAK-02: 121])

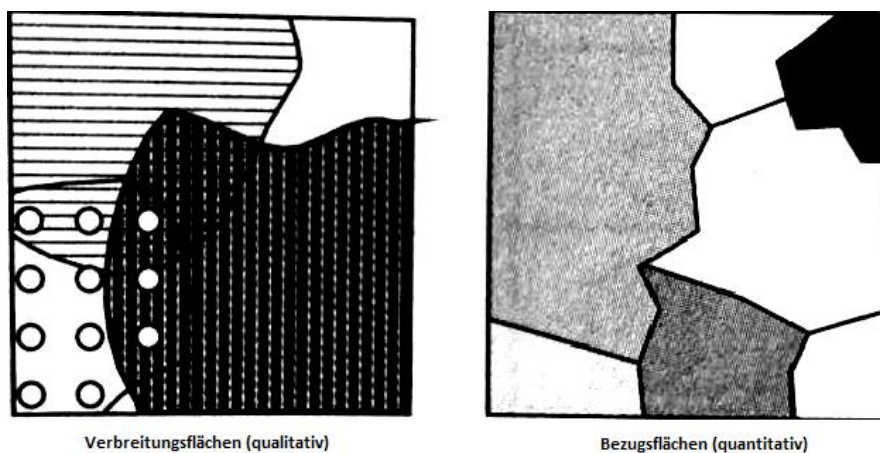


Abbildung 6: Flächenobjekte ([HAK-02:121])

Wie Liem J. bereits erwähnt hat, reichen die genannten drei kartographischen Grundelemente nicht aus, um eine vollständige Aussagekraft zu erreichen. Abstrakte, sowie geometrische Signaturen in der Kartographie sind oft eine Kombination aus Kreis und

Dreieck (punkthafte Signatur), gepunkteten Linien (linienhafte Signaturen) oder Schraffuren und geometrisch-rasterförmigen Mustern (flächenhafte Signaturen). Kombiniert mit den genannten Methoden der qualitativen und quantitativen Differenzierung wird damit eine hohe Variabilität und Kombinationsfähigkeit erreicht. Jedoch ist es natürlich vom eingesetzten Vektorformat abhängig, welche Geometrien es verarbeiten kann. (vgl. [LIE-12: 29])

2.2.2.3 Schrift

In jedem kartographischen Produkt ist die Schrift ein besonderer Teil. Verglichen mit den soeben besprochenen Grundelementen besitzt sie die geringste geometrische Aussagekraft. Jedoch ist sie ein zentrales, erläuterndes Element in einer kartographischen Visualisierung. Qualitäten lassen im Zuge der Beschriftung durch die Farb- und Formvariation und Quantitäten durch Änderungen der Größe abbilden. (vgl. [HAK-02: 137]) Auch für den Bereich der Schriftvisualisierung wurden am Hyperglobus Versuche durchgeführt. Folgende in Tabelle 3 genannte Schriftgrößen wurden als geeignet für eine Visualisierung am sphärischen Display festgelegt.

Schriftart	Rotation	Minimum	Gut Lesbar
Arial	-	8-9px	11px
Arial	45°	10px	11px
Calibri	-	9px	11px
Calibri	45°	11px	12px
Georgia	-	9-10px	11px
Georgia	45°	10-11px	12px
Times New Roman	-	10px	11-12px
Times New Roman	45°	11px	13px

Tabelle 3: Schrift am Hyperglobus ([OPT-16: 2])

Zusätzlich gilt es immer kritisch zu hinterfragen, auf welchem Hintergrund Schrift am Hyperglobus visualisiert wird. Handelt es sich hierbei um farbtintensive, unruhige Hintergründe, so kann die Schrift, wie in Abbildung 7 ersichtlich, durch einen Rahmen oder eine semitransparente Freistellung in einer (Kontrast)-Farbe hervorgehoben werden. (vgl. [OPT-16: 2])



Abbildung 7: Schrift am Hyperglobus ([KIE-18])

2.2.2.4 Legende

Eine Legende muss sich, als Teil des Kartenproduktes, gut vom Inhalt des Produktes abgrenzen und lesbar sein. Jedoch soll sie nicht im Vordergrund des Globenbildes stehen. Bei einem unruhigen Globenbild wird bei Legenden unter anderem mit halbtransparenten Rahmen gearbeitet, um das Globenbild durch den Rahmen zu erkennen. (vgl. [OPT-16: 5])



Abbildung 8: Legende am Hyperglobus ([KIE-18])

2.2.2.5 Generalisierung

Ein weiterer Aspekt der kartographischen Gestaltung am sphärischen Display ist die Generalisierung. Das Ziel dahinter ist es, die Lesbarkeit des Produktes zu optimieren. Je nach Fragestellung gilt es, wichtige Aussagen in den Vordergrund und unwichtige Aussagen in den Hintergrund zu stellen. Um bei großen Datenmengen die Lesbarkeit und Trennung einzelner Objekte zu gewährleisten, können dafür für unterschiedliche Maßstabsstufen und graphische Dichten eingesetzt werden. (siehe Abb. 9) Um die Lesbarkeit am Medium zu garantieren, müssen jeweilige Mindestgrößen jedoch berücksichtigt werden. (vgl. [LIE-02: 29])

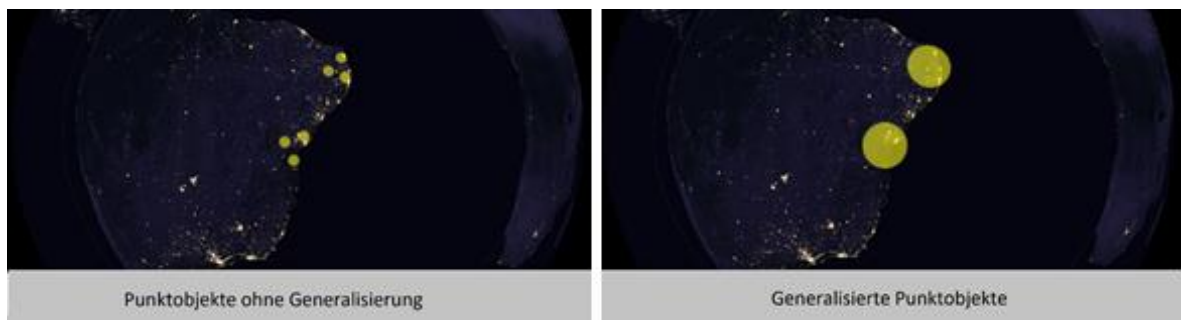


Abbildung 9: Generalisierung von Punktoobjekten am Hyperglobus (Datengrundlage: [EAR-17])

2.2.3 Verzerrungsproblematik am sphärischen Display

Die beiden wesentlichen Bestandteile für eine Visualisierung am sphärischen Display sind ein globaler Basislayer (Hintergrundkarte) und globale Vektordaten. Jene Vektordatensätze

werden als thematische Layer verwendet und stellen, je nach Geometrietyp, verschiedene Objektklassen dar. Beispielhafte Punktdatensätze sind zum Beispiel Städte oder Erdbeben. Liniengeometrien werden unter anderem zur Darstellung von Flugrouten oder Tiefseekabel herangezogen. Flächen hingegen stellen Flächen oder Habitate dar. (vgl. [LIE-12: 1])

Um diese Vektorobjekte am Globus verorten zu können, wird ein globales Koordinatensystem angewendet. (vgl. [COZ-11: 13]) Koordinaten werden in Grad angegeben, wobei die Position von West (-180°) nach Ost (180°) und Süd (-90°) nach Nord (90°) verlaufend angegeben wird. (siehe Abb. 10) Der Nullpunkt liegt bei den Breitenkreisen am Äquator und bei den Längengraden am Nullmeridian (Greenwich). (vgl. [SCH-17: 34])

Sollen Punktdatensätze am sphärischen Display visualisiert werden, so werden sie nach dem momentan gängigsten Ansatz (siehe Kapitel 4.2) als Platkarte in einem Geoinformationssystem oder Bildbearbeitungsprogramm aufbereitet und im Rasterformat (2D) abgespeichert. Da es sich beim sphärischen Display um ein dreidimensionales Produkt handelt, ist eine zweidimensionale Platkarte nicht für eine Wiedergabe auf dem Hyperglobus geeignet. (siehe Kapitel 1.1) Durch die zunehmende Verzerrung, je höher der geographischen Breite, kann diese nicht für alle Anwendungsansätze verwendet werden.

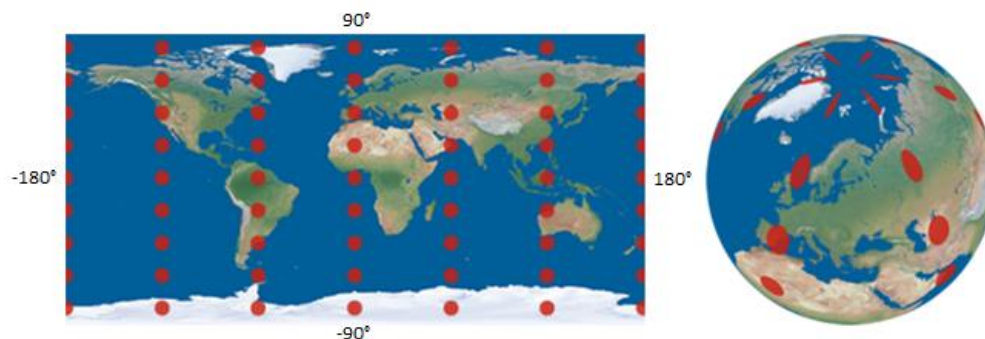


Abbildung 10: Visualisierung am sphärischen Display ohne Berücksichtigung der Verzerrung (2D → 3D) ([LIE-12: 2])

In Abbildung 10 ist die zunehmende Breitenverzerrung am dreidimensionalen Globus deutlich zu erkennen. Daher müssen Geoobjekte am dreidimensionalen, sphärischen Display, abhängig von ihrer geographischen Breite, entzerrt werden.

Um die Verzerrungsproblematik von Kartenprojektionen zu visualisieren, wurde vom Mathematiker Tissot die Methode der Tissot'schen Indikatrix entwickelt. Diese bildet die zunehmende Verzerrung nach der geographischen Breite ab. Anhand der Verzerrungsellipsen werden die Treueigenschaften von kartographischen Abbildungen überprüft.

$$\text{Verzerrung} = \frac{1}{\cos(\text{Breite in Grad})}$$

Formel 1: Formel zur Berechnung der Breite eines Objektes ([PUN-17: 1])

Daher müssen Punktoobjekte für eine Visualisierung im dreidimensionalen Raum, abhängig von der jeweiligen geographischen Breite, auf der Platkarte adaptiert werden. Verzerrungen zwischen dem nördlichen und südlichen 25° Breitengrad sind zu vernachlässigen, da sie am Globus nicht zu erkennen sind. Hingegen ist auf Höhe beider 60° Breitengrade die doppelte Breite Voraussetzung für die korrekte Darstellung der Punktsignatur im dreidimensionalen Raum. (vgl. [LIE-12: 2], [SMU-17: 50])

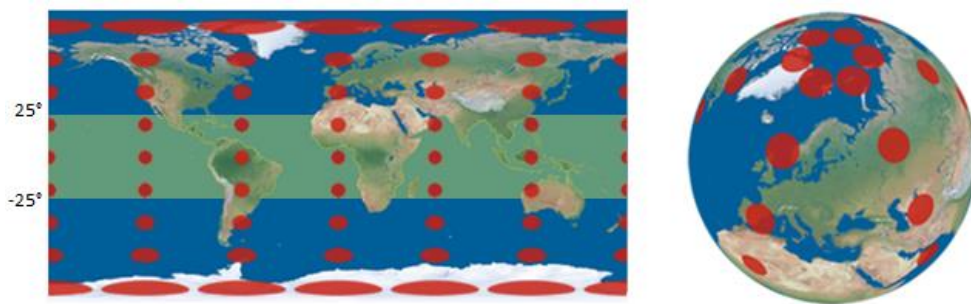


Abbildung 11: Visualisierung am sphärischen Display unter Berücksichtigung der Verzerrung (2D -> 3D) ([LIE-12: 2])

2.2.4 Globenwürdigkeit

Da es sich beim Hyperglobus um eine taktile Darstellung des Planeten handelt, werden Datensätze benötigt, welche in einem globalen Maßstab vorliegen. Bei der Datenauswahl selbst spielt die globale Relevanz eine wesentliche Rolle, da die Visualisierung von großmaßstäbigen Prozessen am sphärischen Display keinen Mehrwert an Information bringt. Von Riedl, A. wurden hierzu Kriterien zur Beurteilung von Datensätzen, bezogen auf deren Globenwürdigkeit, definiert.

„Die Globenwürdigkeit eines Themas bzw. einer Aufgabenstellung – und damit der Vorzug einer Präsentation am Globus gegenüber einer Darstellung als Weltkarte – ist gegeben, wenn:

- *eine enge Bindung an die Gestalt des Himmelskörpers besteht,*
- *die Daten des weltumspannenden Phänomens global verfügbar sind,*
- *der Sachverhalt auch unter dem Gesichtspunkt der Kleinmaßstäbigkeit die Aussagekraft bewahrt,*

- oder in Kombination mit anderen Sachverhalten eine sinnvolle Ergänzung darstellt“ ([RIE-00: 80])

2.2.5 Thematiken

„Im traditionellen Globenbau musste ein Globenthema sowohl in seiner Aussage als auch in der Darstellungsmethode über einen möglichst langen Zeitraum gültig sein, um den Kostenfaktor bei der Herstellung auf ein erträgliches Maß zu reduzieren[...]Mit dem Einzug der Neuen Medien in der Kartographie ergibt sich die Unabhängigkeit des Datenspeichers vom Sichtgerät.“ [RIE-00: 87]

Dadurch, dass es sich beim sphärischen Display um ein digitales Produkt handelt, sind der potentiellen Themenvielfalt keine Grenzen gesetzt. Diesbezüglich hat Riedl A. einen passenden Vergleich aufgestellt: „Die Themenvielfalt ist durchaus mit der eines Atlas vergleichbar“. ([RIE-00: 87]) Digitale Globen sind als Anschauungsobjekt gut geeignet, um geographische Inhalte durch dynamische Präsentationsformen und Interaktivität wiederzugeben. Dadurch wird eine effiziente Übermittlung von Raumvorstellungen wie auch globalen Prozessen und Strukturen ermöglicht. Zusätzlich rückt die Stellung des Globus dadurch in eine zentrale Position: vom Dekorationsgegenstand hin zum fundierten Anschauungsinstrument. (vgl. [RIE-00: 84-87])

Folgende, in Tabelle 4 genannte Themen sind ein Auszug an Inhalten, welche für eine Visualisierung am sphärischen Display geeignet sind.

AUSZUG AUS THEMATIKEN FÜR DAS SPHÄRISCHE DISPLAY					
Mathematische Geographie	Die Erde als Planet	Jahreszeitenverlauf	Humangeographie	Geschichte, Bevölkerung	Entdeckungsgeschichte
		Wendekreise-/Polarkreise			Geopolitik
		Zeitzone			politische Tendenzen
	Erdgestalt	Geoidgestalt			Bevölkerungsverteilung
		Reliefglobus			Migration
		Gradnetz			Hungersnöte
Physische Geographie	Postitionsbestimmung	Gradnetzentwürfe		Verkehr	Tiefseekabel
		Plattentektonik			Schiffsverkehr
		Vulkanismus			Flugstraßen
	Geologie, Tektonik	Kontinentaldrift			Satellitenlaufbahnen
		Innerer Aufbau			Raumfahrt
		Atmosphäre			Krankheitsausbreitungen
		Klimazonen		Wirtschaft	Fischfanggebiete
	Klima, Ozeanographie	Niederschlag			Ernteerträge
		Luftdruck			Energieressourcen
		Meeresströmungen			Rohstoffressourcen
		Böden			Wirtschaftliche Stärke
	Vegetations-, Tiergeographie	Vegetationszonen		Staat, Macht	Militärische Bündnisse
		Jährliche Vegetation			Rüstungsindustrie
		Wanderwegen			Politische Gliederung
	Umweltschutz, Ökologie	Umweltverschmutzung			Kriege
		Ozonkonzentration			
		Desertifikation			
		Artenvielfalt			

Tabelle 4: Auszug aus Themenbereichen für den Hyperglobus ([RIE-00: 88-89])

2.2.6 Animation

„Die Multimediatechnologie bietet mit Hilfe der Animation die echte dynamische Präsentation von globalen Sachverhalten (Bewegungen, Prozesse, Entwicklungen, Simulationen). Veränderungen in Raum und Zeit lassen sich in ihrer Dynamik wiedergeben. Animation ist das geeignete Vehikel zum Transport dynamischer Themen.“
[RIE-00: 57]

In der traditionellen Kartographie wurden Animationen via Zeitreihen dargestellt. In der digitalen Kartographie werden für animierte Darstellungen Bilder erzeugt, um einen belebten Eindruck zu erzeugen. Idealerweise werden pro Sekunde 24 Bilder abgebildet. Dadurch wird der Eindruck einer Bewegung erzeugt. (vgl. [RIE-00: 57])

Verglichen mit der Filmindustrie, wird der Animation in der Kartographie erst seit 30 Jahren eine Rolle zu Teil. Dabei werden graphische Variablen, wie Füllung, Form, Größe und Farbe anhand der dynamischen Variablen Reihenfolge, Dauer und Veränderungsrate animiert. Während Benutzer zur Interpretation dynamischer Phänomene anhand statischer Darstellungen ein großes Vorstellungsvermögen benötigen, erleichtern Animationen diesen Prozess deutlich. So wird der Lernprozess durch eine effizientere und effektivere Informationsvermittlung unterstützt. Das Ergebnis ist sowohl eine schnellere Interpretation des Inhaltes und eine Verankerung der transportierten Information. (vgl. [RIE-00: 57-58])

Unter anderem sind folgende Animationsarten sind für Visualisierungen am Hyperglobus interessant:

- **Zeitliche Animation** von Geoobjekten (Zeitverlauf)
- **Räumliche Animation** (Bewegung) von Geoobjekten (Punkt A -> Punkt B)

2.2.7 Interaktion

„Information perception and knowledge building are individual processes, formed through individual interest, motivation, experience and user knowledge. From this point of view, interactive, dynamic and 3D visualizations may be beneficial for transmitting spatially-related content for providing preferred modes of information access. Several theories have a direct influence on the conception of cartographic applications.“ ([JOB-07: 218])

Durch die Option der Interaktivität, welche in Kapitel 5, basierend auf umgesetzten Global Stories analysiert wird, können NutzerInnen ihr vorhandenes Wissen verändern, prägen und erneuern. Dies hängt von Interaktionen und Erfahrungen ab. Der Mehrwert einer Information nimmt zu, wenn die Informationsaufnahme durch mehrere Perspektiven stattfindet. Interaktive Anwendungen ermöglichen es somit, dass bereits neu aufgenommene Information an erlangtes Wissen angeknüpft wird, durch so genannte persönliche Wissens-Inseln. (vgl. [JOB-07: 218-220])

Auch bei der Umsetzung von Global Stories ist es das Ziel, NutzerInnen aktiv teilnehmen zu lassen. Hierdurch wird die Informationsaufnahme und Wissenserweiterung, personenabhängig, verstärkt. Ein Beispiel hierfür ist das globale Flugnetz. NutzerInnen können durch die direkte, synchrone Vektordatenintegration eine interaktive Auswahl nach Flugrouten oder Fluganbieter vornehmen.

Dies ist der Grund, warum die Interaktivität als eines der wesentlichen Features für Umsetzungen am sphärischen Display gesehen wird. Umso mehr rückt durch dieses Feature die hohe Bedeutung einer internen, synchronen Visualisierung (Kapitel 4.3) von Vektordaten in den Fokus. Denn durch den direkten Objektbezug- nicht nur auf Punkte, sondern auch Linien und Polygone – ergeben sich erweiterte Möglichkeiten der Interaktivität in Global Stories.

Folgende Interaktionen von Geoobjekten sind am Hyperglobus interessant:

- **Attributbasierte Interaktion**
 - zum Beispiel nach Ort, Name, Größe, etc. von Geoobjekten
 - Global Story Quiz (bei korrekter Antwort Rotation des HG zum korrekten Geoobjekt)
- **Orientierung und Navigation** zu Geoobjekten am sphärischen Display

3 Dateiformate

„In a computer, a file format is the layout of a file in terms of how the data within the file is organized. A program that uses the data in a file must be able to recognize and possibly access data within the file. For example, the program that we call a Web browser is able to process and display a file in the HTML file format so that it appears as a Web page, but it cannot display a file in a format designed for Microsoft's Excel program. A particular file format is often indicated as part of a file's name by a file name extension (suffix). Conventionally, the extension is separated by a period from the name and contains three or four letters that identify the format. A program that uses or recognizes a particular file format may or may not care whether the file has the appropriate extension name since it can actually examine the bits in the file to see whether the format (layout) is one it recognizes.“ ([FIL-05])

Wie bereits in Kapitel 1 erwähnt, werden verschiedene Dateiformate angewandt, um räumliche Information am taktilen sphärischen Display, dem Hyperglobus, zu visualisieren.

In einem Dateiformat werden Struktur und Inhalt einer Datei definiert. Die jeweilige Formatart wird durch die Extension (*.xxx) einer Datei angegeben. Zu Faktoren hinsichtlich der Umsetzung der Informationsvisualisierung und Wahl des Dateiformates können unter anderem Fragestellung wie auch die Dateigröße der Daten gezählt werden. Dieses Kapitel gibt eine Einführung in die Dateiformate Raster und Vektor. Zusätzlich wird betrachtet, wie und in welcher Weise jene am sphärischen Display eingesetzt werden und welche Vorteile- und Nachteile sie besitzen. Ziel ist es, eine Grundlage für die in Kapitel 4 vorgestellten Visualisierungsmethoden am Hyperglobus und die Bedeutsamkeit von Vektorformaten auf dem Globenmedium zu schaffen.

3.1 Das Rastermodell

„Geometrische Sachverhalte sind in Rastermodellen um vieles einfacher darzustellen als in Vektormodellen. Es gibt nur einen Entitätstyp, ein geometrisches Element im Rastermodell, die Rasterzelle oder Rastermasche, in Anlehnung an die Anwendungen in der graphischen Datenverarbeitung und der Bildverarbeitung als Pixel bezeichnet. (Die dreidimensionale Verallgemeinerung ist das Voxel.)“ ([BAR-95: 95])

Ein Rasterbild besteht aus regelmäßig angeordneten, gleich großen Rasterzellen, welche einer regelmäßigen Anordnung unterliegen. Dabei handelt es sich um eine Matrix aus Pixeln (Zellen). Jede einzelne Zelle besitzt eine Information bzw. einen Wert, der Informationen wie Höhe, Größe oder Spektralwert wiedergibt. Rasterformate finden in Geoinformation in vielen Bereichen Einsatz, unter anderem in der Fernerkundung, über photogrammetrische Aufnahmen von Objekten bis hin zu gescannten Kartenprodukten. Dabei hängt es von der Zellengröße ab, wie detailliert Features im Bild dargestellt werden und wie sich die Bildqualität bei fortschreitender Skalierung verhält. Parallel dazu nimmt die Speichergröße der Rasterdatei zu, wenn die Größe der Rasterzellen abnimmt. (vgl. [WAS-16])

Koordinaten eines Rasterbildes werden in einem kartesischen Koordinatensystem abgebildet (vgl. Abb. 12), in welchem sowohl Zeilen und Spalten der Matrix parallel zu x- und y-Achse verlaufen und mit dem Wert 0 beginnen. (vgl. [WAS-16])

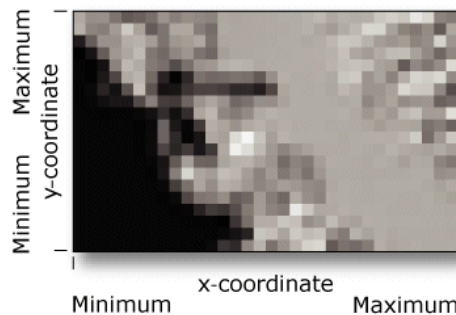


Abbildung 12: Rasterformat ([WAS-16])

3.1.1 Rasterformate am sphärischen Display

Der wesentliche Vorteil einer Visualisierung von Rasterbildern am sphärischen Display ist jener der flächenhaften, globalen Betrachtung. Aus genanntem Grund enthält eine Visualisierung am taktilen Hyperglobus zumindest immer eine Rasterdatei als Basislayer (Hintergrundkarte). Eine einfache Datenstruktur in Form einer Bildmatrix und kurze Erfassungszeiten sind als positive Attribute zu erwähnen. Als Nachteile ergeben sich bei einer Visualisierung von Rasterdaten am sphärischen Display hingegen ein quasi unmöglicher Objektbezug, große Datenmengen und ein daraus resultierender großer Speicherbedarf. Dies resultiert daraus, dass die Bildberechnung von Rasterdaten am Hyperglobus in Echtzeit stattfindet. Es werden pro Sekunde bis zu 30 Bilder mit hoher Auflösung abgebildet (4096*2048 px), dies teils in mehreren Ebenen. Auch der Abnahme der Auflösung eines Bildes bei zunehmender Skalierung ist in diesem Zusammenhang zu berücksichtigen. (vgl. [RIE-12], [HAR-09], [LAN-13: 274])

Entscheidende Faktoren hinsichtlich einer Visualisierung eines Rasterformattyps am sphärischen Display sind Farbtiefe (Farbabstufungen), Transparenz und Komprimierung (Speicherbedarf). Die Anzahl der darstellbaren Farbabstufungen wird, abhängig von gespeicherten Bits pro Pixel (bpp), durch die Farbtiefe definiert. Am taktilen sphärischen Display kommt der Farbraum RGB (Rot-Grün-Blau) zum Einsatz. Rasterbilder am Hyperglobus selbst weisen meist einen 8-bit pro Bildkanal auf, dies entspricht einem True-Color Bild (24-bit) mit einem Spektrum über 16,7 ($(2^8)^3$) Millionen darstellbaren Farben. Soll Transparenz im Bild mitgespeichert werden, muss ein weiterer Kanal abgespeichert werden, der sogenannte Alpha- oder Transparenzkanal. Wird dieser mit 8 bpp mitgespeichert besitzt das Rasterbild eine Farbtiefe von 32 bit. (vgl. [BIL-17: 1])

Die Kompression von (Bild)Daten wird durchgeführt, um den Speicherbedarf (Größe) einer Datei zu verkleinern und eine Optimierung in der Performance der Bildverarbeitung zu erreichen. Überflüssige und redundante Information werden hierbei durch Algorithmen aus der Datei entfernt. Bei der verlustfreien Kompression findet ohne Informationsverlust eine Reduktion der Bildgröße statt. In diesem Fall kann die ursprüngliche Datei ohne Qualitätsverlust wiederhergestellt werden. Verlustbehaftete Kompressionsalgorithmen sind hingegen mit Qualitätsreduktion verbunden. Dafür werden geringere Dateigrößen als bei der verlustfreien Kompression erzielt. Die Ursprungsqualität der Ausgangsdatei kann nicht wiedererlangt werden. (vgl. [BIL-17: 1-2])

3.1.1.1 JPEG

“JPEG works on either full-color or gray-scale images; it does not handle bilevel (black and white) images, at least not well. It doesn't handle color mapped images either; you have to pre-expand those into an unmapped full-color representation. JPEG works best on "continuous tone" images. Images with many sudden jumps in color values will not compress well.”
([INT-03])

Bei JPEG handelt es sich um ein Rasterformat, welches zur Komprimierung von Bilddateien verwendet wird. Dies resultiert in kleiner werdenden Dateigrößen und in einem Verlust der Bildqualität. Das Format wird hauptsächlich von Fotografen und Grafikern verwendet und wird zur Darstellung von natürlich strukturierten Graustufen- und Farbbildern verwendet (Halbtonbilder). Enthalten Bilddateien klar abgegrenzte Objekte, wie z.B. Punkte, Polygone oder Linien, oder homogene Farbbereiche, so ist das JPEG-Format nicht zur Darstellung dieser dafür geeignet. Es ist zur Visualisierung von Farbverläufen- und Übergängen vorgesehen.

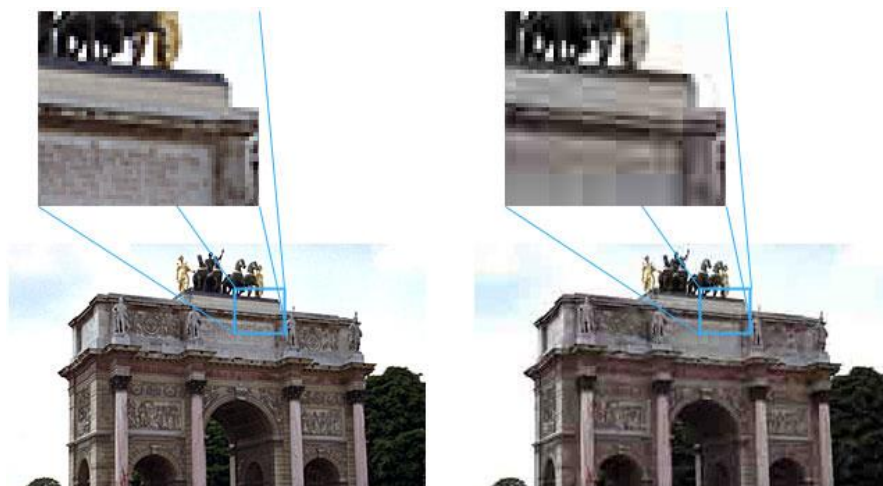


Abbildung 13: Qualitätsunterschiede durch Komprimierung beim *.jpeg Format ([WIC-02])

3.1.1.2 PNG

Bei PNG (Portable Network Graphics) handelt es sich um ein Bildformat, welches mit einer verlustfreien Komprimierung arbeitet. Daher ist es, verglichen mit JPEG-Format, sehr gut zur Visualisierung von Grafiken und klar abgegrenzten Bildobjekten geeignet. Bei Reduzierung der Speichergröße findet in der Bilddatei kein Informationsverlust statt. Zusätzlich kann im Bildformat Transparenz (siehe Abb. 14) mit abgespeichert werden. (vgl. [SMU-17: 38])

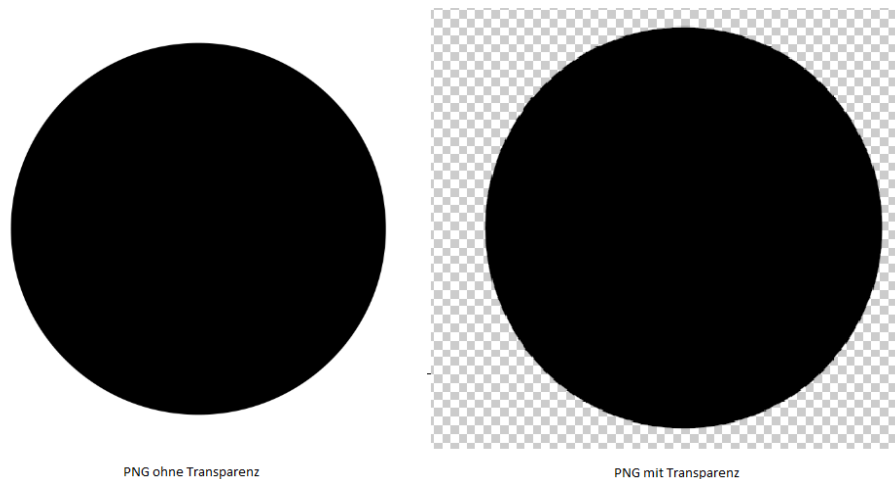


Abbildung 14: PNG ohne/mit Transparenz (eigene Darstellung)

3.1.1.3 DDS

Beim DDS Format handelt es sich um ein eigens von Microsoft entwickeltes Format. Als Bestandteil von DirectX kann das Format in real-time Rendering Applikationen verwendet werden. Die Kompression wird beim DDS Format direkt in der Grafikkarte durchgeführt. Zusätzlich wird weniger Speicherbedarf benötigt, da im Gegensatz zu JPEG / PNG keine Dekodierung benötigt wird und komprimierte Daten direkt in den Grafikkartenspeicher geladen werden. Bei den zur Kompression angewandten Algorithmen handelt es sich um S3 Texture Compression Algorithmen, diese sind verlustbehaftet. Insgesamt wurde der S3TC Algorithmus in fünf Varianten implementiert, diese wurden nach in DirectX zugewiesenen Codes (FourCC) benannt: DXT1 – DXT5, wobei DXT2 und DXT4 kaum Einsatz finden. (vgl. [BRO-13])

Texturen werden bei Anwendung der DXT-Kompression in 4x4 Pixel-Blöcke zerlegt. Anschließend werden diese komprimiert. Basierend auf jenen 16 Pixeln wird eine Farbtabelle von 4 Farben bestimmt. Aus jener Tabelle wird jedem Pixel einer der Farbwerte zugewiesen. Jedoch werden in der Farbtabelle selbst nur die zwei Extremwerte abgespeichert. Die beiden anderen Farbwerte werden extern interpoliert. Die Farbwerte selbst werden als 16-bit abgespeichert, wobei diese folgend aufgeteilt sind: **Rot** 5-bit/**Grün** 6-bit/**Blau** 5-bit. (vgl. [BRO-13])

- **DXT1**

Zwei 16-bit RGB Farbwerte und eine 4x4 2-bit Lookup-Tabelle resultieren in 64-bit pro 4x4 Pixel-Block. Daraus resultiert eine 1:8 Kompression, verglichen mit 512-bit ohne Kompression. In DXT1 Alpha kann zusätzlich ein 1-bit Alpha Kanal mitgespeichert werden. Dafür wird hingegen aus der Farbtabelle eine der vier Farben als transparent definiert. Dies limitiert den Farbumfang zusätzlich. (vgl. [BRO-13])

- **DXT3 und DXT5**

Hier werden Farbwerte wie in DXT1 abgespeichert. Zusätzlich wird das Speichern eines Transparenzkanals ermöglicht. (64 bit) Des resultiert in einer Kompressionsrate von 1:4. (vgl. [BRO-13])

- **DXT3**

Transparenzinformationen werden von DXT3 explizit mit 4-bit abgespeichert. Daraus resultieren 16 Stufen der Transparenz. (vgl. [BRO-13])

- **DXT5**

Ähnlich wie Farbinformation werden bei DXT5 Transparenzinformationen in einer „Farbtabelle“ plus einer 4x4 3-bit Lookup-Tabelle abgespeichert. Es werden zwei Extremwerte (8-bit Alpha) gespeichert und 6 weitere Alphawerte interpoliert. Dies resultiert in gleichmäßigeren Transparenzverläufen. (vgl. [BRO-13])

Abbildung 15 gibt einen Überblick über in der Globensoftware genutzte Bildformate.

	JPEG	PNG	DXT1	DXT1A	DDS DXT3	DXT5	RGB
Farbtiefe (R-G-B bit)	8-8-8	8-8-8	5-6-5	5-6-5	5-6-5	5-6-5	8-8-8
Alphakanal (bit)	-	8	-	1	4 explizit	4 interpoliert	-
Verlustfreie Komprimierung	Nein	Ja	Nein	Nein	Nein	Nein	Ja

Abbildung 15: In der Globensoftware verwendete Bildformate ([BIL-17: 5])

Die Bildqualität ist bei Qualitätsverlust durch die Datenkompression vom Inhalt der Bilddatei abhängig. Die Kompressionsrate von Bildern ist fix, dies bedeutet, dass bei gleicher Bildauflösung die Größe jeder Bilddatei ident ist. Wie bereits am Beginn von Kapitel 3.1.1.3 erwähnt werden im DDS-Format Bilder mit 16-bit abgespeichert. Verglichen mit einem Originalbild (z.B. JPEG, True-Color-Bild mit 24-bit) findet ein Verlust von feinen Farbnuancen statt. (vgl. [BIL-17: 4])

Der Prozess der Komprimierung im Format DXT1 wurde bereits in diesem Kapitel erklärt. Durch die lineare Interpolation zwischen den Farbwerten bedeutet dies, dass, je unterschiedlicher Farbwerte sind, Farbinformation verloren geht. Dies liegt daran, dass diese auf keiner Linie im Farbraum liegen. (siehe Abb.16) Daher ist der Qualitätsverlust von komprimierten Bildern größer, je mehr verschiedene Farben auf kleinem Raum vorkommen. (vgl. [BIL-17: 4])

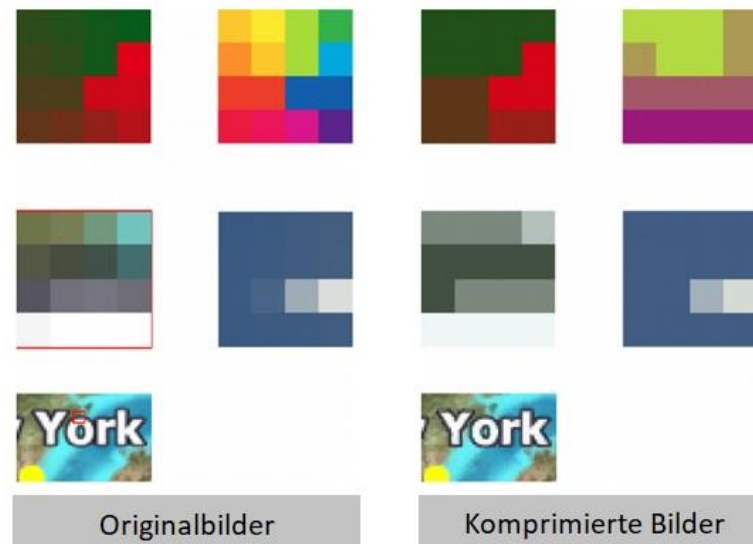


Abbildung 16: Originalbild vs. komprimiertes Bild ([BIL-17: 4])

3.2 Das Vektormodell

„Vektordaten repräsentieren diskrete Phänomene, in der OGC-Nomenklatur als Features bezeichnet. Ein Feature ist ein identifizierbares Objekt mit raum-zeitlichen und beliebigen weiteren Attributen. Die räumlichen Charakteristika werden über geometrische Primitiva – also Punkte, Kurven, Oberflächen und Körper – modelliert. Zwischen Features können topologische Beziehungen – etwa enthalten-in, überlappt-mit – bestehen“ ([BAU-08: 435])

Unter Vektoren werden die geometrischen Grundprimitiva Punkt, Linie oder Fläche verstanden. Die Lage eines Vektors selbst wird immer durch Koordinaten festgelegt, wobei die Lageinformation der Primitiva folgend definiert ist:

- **Punkt:** Koordinatenpaar X/Y – Träger der Geometrie
- **Linie:** Startpunkt X/Y + Richtungswinkel und Länge der Linie

Zusätzlich ist es auch möglich die dritte Dimension durch eine Z-Koordinate im Rahmen von Vektoren abzubilden. Jedoch werden Höheninformationen in Vektordaten als Attributinformation geführt und werden dadurch, im Gegensatz zu Lagekoordinaten, nicht zwingend benötigt.

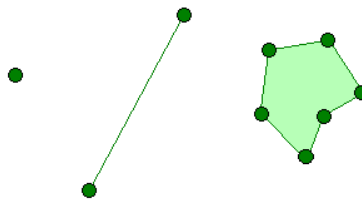


Abbildung 17: Geometrische Grundprimitiva (eigene Darstellung)

Als wesentliches Merkmal von Vektorformaten gilt die verlustfreie Skalierung und ein geringerer Speicherbedarf gegenüber anderen Dateiformaten. Als weiterer, wichtiger Aspekt sind der mögliche Objektbezug und daraus resultierende Animations- und Interaktionsmöglichkeiten zu nennen. (siehe Abb. 18)



Abbildung 18: Verlustfreie Skalierung Vektorformat (Datengrundlage: [NAT-18])

3.2.1 Vektormodelle

Digitale Datenmodelle werden in der Geoinformation und Kartographie herangezogen, um reale Sachverhalte abzubilden. Im Zuge jener werden räumliche und nicht-räumliche Attribute, nicht-räumliche Relationen und raumbezogene Relationen geographischer Objekte in digitalen Objektmodellen (DOM), basierend auf Vektoren, verwaltet. Nach Hake G. gilt es hierbei verschiedene Vektormodelle zu unterscheiden. (vgl. [HAK-02: 156-157])

Von einem Spagetti-Datenmodell spricht man bei linienhaften Digitalisierungen, unter anderem von Landkarten. Diverse Nachbarschaftsbeziehungen sind zwischen digitalisierten Objekten nicht modelliert, lassen sich aber durch räumliche Operationen berechnen. Aus diesem Grund eignet sich das Spagetti-Modell weniger für räumliche Analyse, jedoch zur Erstellung von Kartenprodukten. (vgl. [HAK-02: 158])

In einem topologischen Datenmodell findet eine explizite Modellierung von Nachbarschaftsbeziehungen statt. Kanten werden Identifikatoren von Objekten links und von rechts zugeordnet. Basierend darauf lassen sich Flächen bilden. Da dieses Modell eine redundanzfreie Speicherung ermöglicht, stellt es eine Verbesserung gegenüber dem Spagetti-Datenmodell dar. Dem gegenüber stehen zeitkonsumierende sequentielle

Suchprozesse bei großräumigen Auswertungen (z.B. Menge aller Maschen, welche zu einer Kante gehören.) Das hierarchische Vektordatenmodell ist eine Weiterentwicklung des topologischen Datenmodells. Als Ergänzung wurde in diesem als logische Ergänzung die Kette (chain) eingeführt. (vgl. [HAK-02: 158-159])

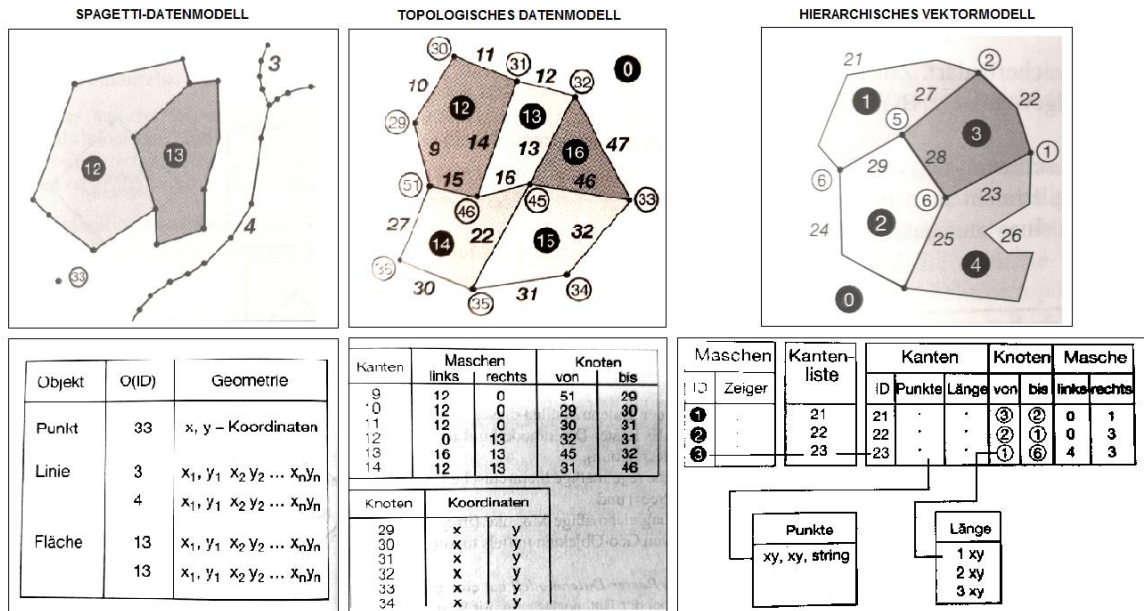


Abbildung 19: Vektormodelle ([HAK-02:158-159])

Das topologische Daten- und hierarchische Vektormodell werden hauptsächlich für räumliche Analysen herangezogen. Da es sich beim sphärischen Display um ein digitales, dreidimensionales kartographisches Produkt handelt, stehen topologische Aspekte und räumliche Analysen nicht im Zentrum. Unstrukturierte Geometriedaten und deren Sachattribute sind bestens geeignet, um Information am Hyperglobus zu visualisieren. Aus diesem Grund ist das Spagetti-Datenmodell ein geeignetes Vektormodell im Zuge dieser Forschungsarbeit.

3.2.2 Skalenspektrum & Variabilität

Geoobjekte im Vektorformat sind, abhängig von ihrer Thematik, im Rahmen ihrer Attribute von Variablen und Merkmalen gekennzeichnet. Die Attribute selbst decken hierbei ein Skalenspektrum ab (vgl. [LAN-13: 165]), welches sich nach und Lange N. in die in Tabelle 5 genannten statistischen Skalenniveaus gliedert. Die Niveaus selbst zeichnen sich durch besondere mathematische Eigenschaften aus.

Skalenniveau	Eigenschaften	Beschreibung	Beispiele
<i>Nominal</i>	gleich/ungleich	Vorhanden oder nicht vorhanden und gleich oder ungleich	PLZ, Namen, Codierungen
<i>Ordinal</i>	größer/kleiner	auf Nominalskala aufbauend, Vergleichsoperationen sind möglich	Bewertungsstufen, Ränge, Schulnoten
<i>Intervall</i>	Abstand (+/-)	auf Nominalskala aufbauend, Addition und Subtraktion sind möglich	Metrische Daten mit nicht eindeutigem Nullpunkt: Grad Celsius
<i>Rational</i>	Verhältnis(./:)	auf Intervallskala aufbauend, Multiplikation und Division, Verhältnisse können berechnet werden	Betrachtung von Verhältnissen, Multiplikation und Division

Tabelle 5: Skalenniveaus ([LAN-13: 165])

Als weitere, wichtige Eigenschaft von Vektorobjekten ist deren Dynamik anzuführen. So ändern sich Geoobjekte über einen Zeitverlauf gesehen oft in ihrer Thematik. Im Rahmen dieser Veränderung treten zwei Variabilitäten auf:

- **Zeitliche Variabilität**

Unter der zeitlichen Variabilität ist eine Veränderung der Merkmale oder Attribute zwischen Zeitpunkten zu verstehen. Dabei kann es sich um eine Erweiterung durch neue Merkmalsausprägungen oder eine Datenaktualisierung handeln. Es ändert sich jedoch nichts an der Lage oder Geometrie der Objekte, sondern nur an der thematischen Information. (vgl. [LAN-13: 166])

- **Räumliche Variabilität**

Die räumliche Variabilität hingegen beschreibt die räumliche Veränderung über einen Zeitverlauf. Dies impliziert eine Änderung von Ausdehnung oder Lage der Geometrie der Daten. (vgl. [LAN-13: 166])

Für eine Visualisierung am Hyperglobus spielt der thematische Aspekt der Vektordaten die entscheidende Rolle, da jener die Hauptinformationsquelle von Global Stories ist. In Kapitel 1 wurden bereits Forschungsfragen zu den Aspekten der Interaktion und Animation definiert. Daher wird der Datensatz, welcher für die praktische Evaluierung der Dateiformate in Kapitel 8 herangezogen wird, bezüglich der Anwendung jener Features untersucht. Dadurch wird gesichert, dass die Daten auch für eine Untersuchung, bezüglich der in den Forschungsfragen definierten Anforderungen, geeignet sind.

3.3 Dateiformate im Vergleich

Generell ist kein Vergleich zwischen Raster- und Vektordaten möglich, da es sich hierbei um von Grund auf verschiedene Dateitypen handelt. Grundsätzlich ist es möglich, Fragestellungen mit beiden Dateiformaten zu behandeln und umzusetzen. Hierbei gibt es jedoch je nach Fragestellung Vor- und Nachteile der Formate, sei es die Auflösung, die Dateigröße oder Analysetechniken. (vgl. [LAN-13: 335]) Oft werden die Formate auch in Kombination verwendet, wie unter anderem am taktilen sphärischen Display.

Zusammenfassend sind Rasterformate für kleinmaßstäbige Anwendungen und großräumige Überblicke, wie es vor allem in der Fernerkundung, der digitalen Bildverarbeitung und bei Modellierungen der Fall ist, zu bevorzugen. Dem gegenüber stehen ein großer Speicherbedarf, damit verbundene Laufzeitprobleme und ersichtliche Rasterzellen bei Skalierung. (vgl. [LAN-13: 336-335]) Vektorgrafiken hingegen eignen sich für einen anderen Einsatzbereich, dieser umfasst kleinmaßstäbige Anwendungsbereiche. Dies liegt an einer höheren Eindeutigkeit und Genauigkeit bei Zoomprozessen. (vgl. Tab. 6).

	Vektor	Raster
Vorteile	hohe geometrische Genauigkeit	einfache Datenstrukturen
	eindeutige Objektbeschreibung	geringer Aufwand bei Erfassung der Geometrie und Topologie
	geringe Datenmengen	kompatibel mit Fernerkundungs- und Scannerdaten
	größere Ähnlichkeit der graphischen Präsentation mit traditionellen Karten	einfaches Überlagern und Verschneiden von Geoobjekten
		einfache logische und algebraische Operationen
Nachteile	komplexe Datenstrukturen	keine Form- und Lagetreue der Geoobjekte
	aufwändige Erfassung der Geometrie und Topologie	höherer Speicheraufwand
	aufwendige und rechenintensive logische und algebraische Operationen	kleine Pixelgröße mit explodierenden Datenmengen für höhere Genauigkeitsanforderungen
	parallele geometrische und topologische Beschreibung der Geoobjekte	Skalierungsproblematik (graphische Präsentation)
		aufwendige Koordinatentransformationen

Tabelle 6: Rasterformat vs. Vektorformat ([LAN-13: 336])

3.3.1 Dateiformate am sphärischen Display

Für eine Visualisierung am sphärischen Display sind sowohl Raster- als auch Vektordaten essentiell. Das Abbild der Erde, oft handelt es sich hierbei um ein Satellitenbild, wird durch Hintergrundkarten (Plattkarten) im Rasterformat dargestellt. Dieses ist notwendig, um den

globalen Kontext und den Bezug zum Planeten darzustellen. Wenn es jedoch um die Thematik geht, sind Vektordaten meist unverzichtbar, denn nur mit einer Basiskarte können in vielen Fällen keine Global Stories aufgebaut werden.

Insgesamt werden die drei Bildformate JPEG, PNG und DDS am sphärischen Display eingesetzt. Handelt es sich bei einer Visualisierung am Hyperglobus um Bilddaten, welche Farbverläufe enthalten, so ist das JPEG-Format als Dateiformat geeignet. Farbverläufe sind unter anderem in Satellitenbildern oder z.B. einer globalen Geländekarte (Schummerung) enthalten. Geht es um die Visualisierung von klar abgegrenzten Objekten, wie Punkt, Linien und Polygone, so ist das PNG-Format zu wählen, da Geoobjekte auch bei stattgefundener Komprimierung nicht an Bildqualität verlieren. Zusätzlich wird das PNG-Format bevorzugt, da es Transparenz mitspeichern kann. Dadurch wird ermöglicht, dass mehrere Layer kombiniert werden können und die Hintergrundkarte am sphärischen Display nicht vom thematischen Layer überdeckt wird. Die Komprimierung der Formate JPEG und PNG ins DDS-Format wird aus Performancegründen der Grafikkarte am HG durchgeführt.

Trotzdem sind Vektordatensätze als Ausgangsformat als zentrale Datenquelle des sphärischen Displays anzusehen. Ohne sie ist meist kein thematischer Mehrwert im Zuge der Geovisualisierung am taktilen Globenmedium zu erreichen.

Zusammenfassend ist die Wahl von Dateiformaten abhängig von der jeweiligen Fragestellung und Thematik der Global Story. Wichtige, ausschlaggebende Punkte hinsichtlich der Formatauswahl sind in Tabelle 7 genannt. Durch den Objektbezug können Geoobjekte (Punkt/Linie/Fläche) direkt angesprochen werden. Diese Option entfällt bei Rasterformaten. Animationen können für Visualisierungen am Hyperglobus sowohl mit Raster wie auch Vektordaten umgesetzt werden. Im Fall der indirekten Rastervisualisierung (Kapitel 4.2) bedeutet dies jedoch, dass die Animation bereits in der Bildbearbeitungssoftware durchgeführt wird und diese in Bildsequenzen gerendert wird. Hingegen kann die Animation von Vektorobjekten objektbasiert umgesetzt werden und benötigt keine vorherige, externe Bearbeitung.

	Vektorformat	Rasterformat
Interaktivität	+	-
Animation	++	+

Tabelle 7: Vor- und Nachteile der Dateiformate (eigene Darstellung)

4 Methoden der Geovisualisierung am sphärischen Display

Das taktile sphärische Display ist ein dreidimensionales Objekt. Werden Vektordaten in einem GIS aufbereitet und für eine Visualisierung am Hyperglobus ins Rasterformat konvertiert, so wird die Verzerrung der thematischen Daten mit ansteigender Breite immer höher. Dies ist bei der quadratischen Platkarte auf die zunehmende Verzerrung, je höher die geographische Breite, zurückzuführen. (siehe Kapitel 2.2.2) Aus diesem Grund findet die Entzerrung von Vektordaten bereits vor der Konvertierung in das Inputraster statt. Dieser Schritt ermöglicht, dass die räumliche Information als Endresultat korrekt entzerrt am dreidimensionalen Globenkörper wiedergegeben wird.

Liem J. ([LIE-12: 45]) hat hinsichtlich dieser Problemstellung folgende in Abb. 20 genannte Kriterien für die Methodenklassifizierung im Rahmen der Datenintegration von Vektor- und Rasterdaten am sphärischen Display definiert.

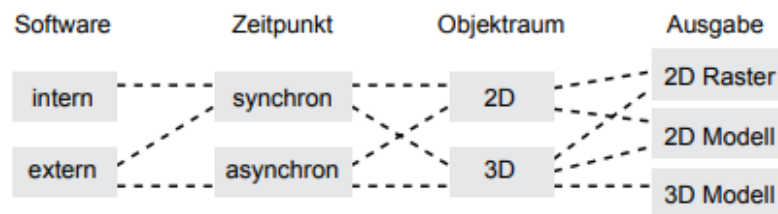


Abbildung 20: Methoden zur Datenintegration am Hyperglobus ([LIE-12: 45])

Grundlegend wird immer unterschieden, ob die Implementierung in der Software direkt und zur Laufzeit oder extern und unabhängig durchgeführt wird. Im Hinblick auf den Zeitpunkt der Aufbereitung der Vektorgeometrien gilt es zwischen synchronen und asynchronen Prozessen zu differenzieren. Werden Prozesse in der Globensoftware selbst durchgeführt, sind diese immer als synchron anzusehen. Dies gilt auch, wenn Vektorgeometrien parallel zur Laufzeit der Globensoftware aufbereitet, visualisiert und in Echtzeit an jene übergeben werden. Von einem asynchronen Visualisierungszeitpunkt hingegen spricht man, wenn die Vektorgeometrieaufbereitung vor Laufzeit der Globensoftware durchgeführt wurde. Einen weiteren wesentlichen Aspekt stellt der Objektraum dar, bei dem zwischen zwei- und dreidimensionalen Räumen unterschieden wird. Meist werden Vektordatensätze, basierend auf der kartographischen Projektion der quadratischen Platkarte, aufbereitet und entzerrt. In diesem Arbeitsschema findet der Import in die Globensoftware (und den 3D-Raum) erst nach der Konvertierung in ein Rasterformat statt. Doch der Import von Vektorgeometrien kann auch direkt (siehe Abb. 21) auf dem dreidimensionalen sphärischen Display erfolgen. Dies impliziert keine gesonderte Entzerrung der vorhandenen Vektorgeometrien. (vgl. [LIE-12: 46-47])

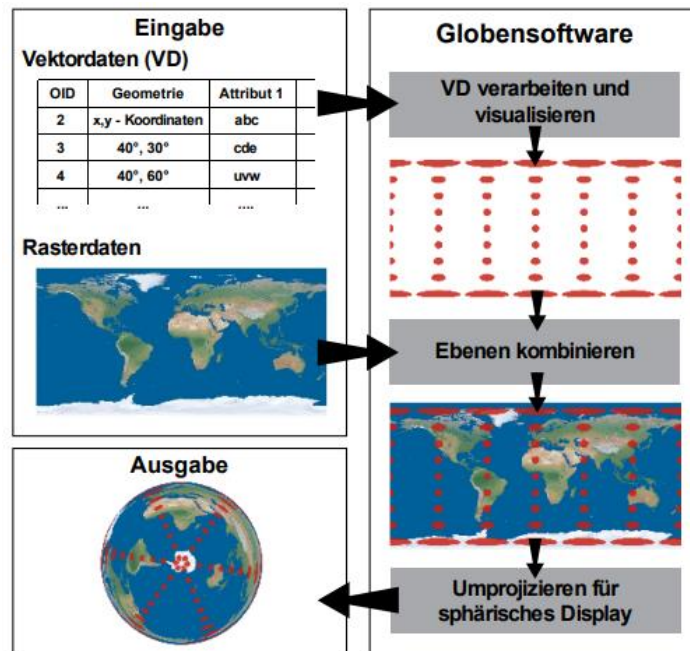


Abbildung 21: Interne, synchrone Datenintegration am sphärischen Display ([LIE-12: 47])

Die Arbeitsschritte und Abbildungen der folgenden Unterkapitel beruhen auf einer Visualisierung durch die Globensoftware OmniSuite. Im Rahmen seiner Diplomarbeit hat sich Kristen J. ([KRI-12]) mit der *Entwicklung von interaktiven kartographischen 3D-Applikationen am Beispiel eines taktilen Hyperglobus* auseinandergesetzt. Die Software OmniSuite, welche im Rahmen jener Arbeit programmtechnisch umgesetzt wurde und von Riedl A. ([RIE-00]) entwickelt wurde, wird unter anderem von der HRG zur Erstellung von Global Stories am sphärischen Display angewandt.

„Grob lassen sich in der drei Bereiche unterscheiden: ein Autorenmodul zur Erstellung von Inhalten (sog. Global Stories), ein Präsentations- und Interaktionssystem zur Steuerung der Präsentation und ein Programm, das in Echtzeit die Inhalte rendert und projiziert. Dieses Programm nennt sich *Render Engine* und ist der Kern der Globensoftware. Die errechnete Abbildung nennt Kristen die „*Omiglobe-Abbildung*“. Das Programm wird in C++ entwickelt und verwendet die freie 3D Grafik-Engine *Ogre3D*.“ ([LIE-12: 17])

Hinsichtlich technischer und programmspezifischer Detailinformationen wird auf jene Arbeit verwiesen.

4.1 Direkte Rastervisualisierung

Im Prozess der direkten Rastervisualisierung am taktilen Hyperglobus werden quadratische Plattkarten im Rasterformat, welche in einer Ausdehnung von 2^n Pixel vorliegen, in die Globensoftware OmniSuite importiert. Diese Rasterdateien werden im Schritt des Imports

von der Software in das benötigte *.dds (Direct Draw Surface) Rasterformat konvertiert. Letztendlich werden die Rasterbilder am dreidimensionalen sphärischen Display als Hintergrundkarte visualisiert. (siehe Abb. 22)

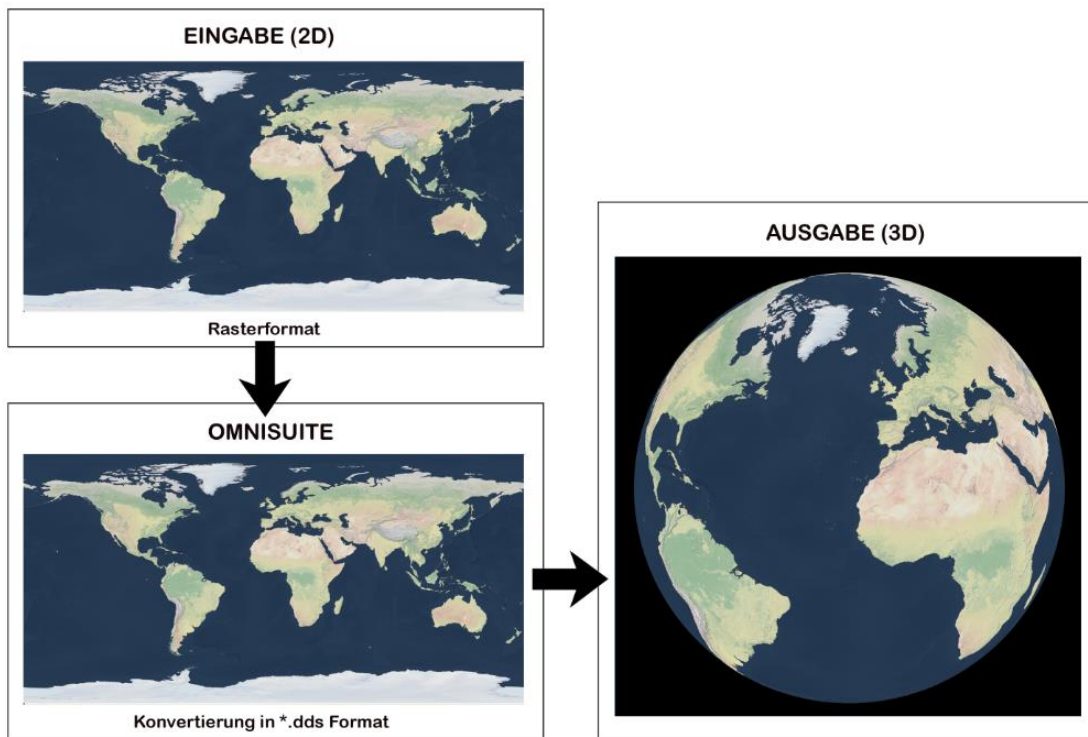


Abbildung 22: Direkte Rastervisualisierung OmniSuite (Datengrundlage: [NAT-18])

4.2 Indirekte Rastervisualisierung

Die indirekte Rastervisualisierung beschreibt den Ansatz der momentan meist genutzten Visualisierungsmethode von Vektordatensätzen am Hyperglobus. Wie bereits von Kristen J. (vgl. [KRI-12: 47-48]) beschrieben, basiert die Abbildung von Vektordaten am sphärischen Display auf mehreren verschiedenen Lösungsansätzen. In der zweidimensionalen Visualisierungsvariante für Vektordaten werden thematische Layer über der Hintergrundkarte ergänzt und gerastert. Jedoch unterliegen die Vektordatensätze der Problematik der Verzerrung im dreidimensionalen Raum. Deswegen ist eine auf der Tissot'schen Indikatrix basierende Entzerrung vor dem Schritt der Rasterkonvertierung unumgänglich. Der Zwischenschritt der Geoobjektentzerrung, abhängig von der geographischen Breite, wird meist in einer Bildbearbeitungssoftware durchgeführt. (vgl. [SMU-17: 67]) Erst danach kann ohne Geometrieverfälschung ein Import in die Globensoftware stattfinden. Nach den von Liem J. definierten Visualisierungskriterien ist die indirekte Rastervisualisierung somit als asynchrone, externe Visualisierungsmethode anzusehen. (siehe Abb. 23)

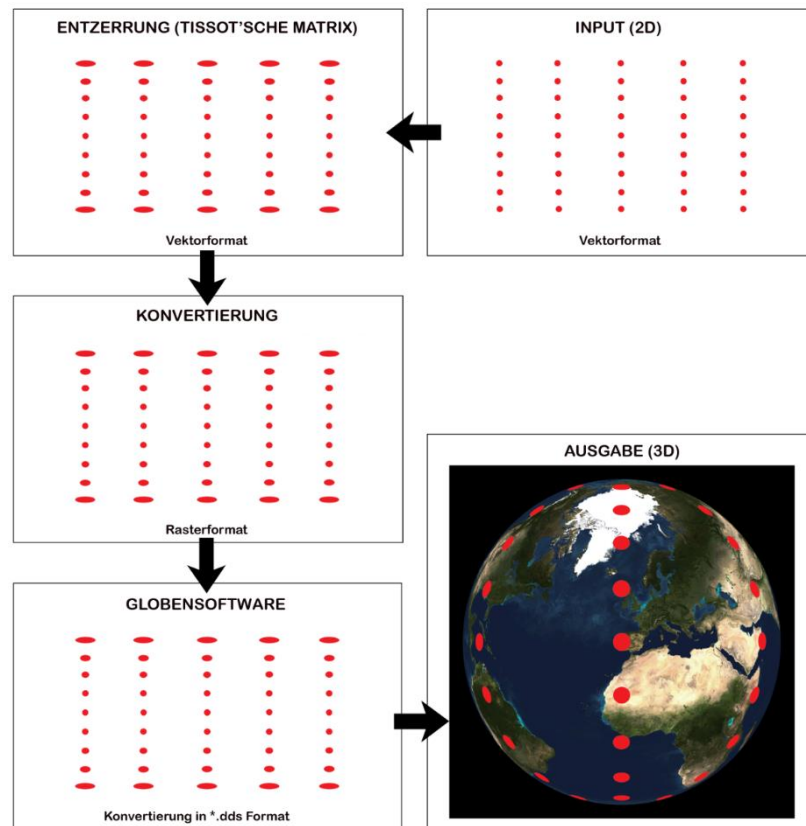


Abbildung 23: Indirekte Rastervisualisierung OmniSuite (eigene Darstellung)

Im Zuge seiner Masterarbeit hat sich Smutny T. mit der Automatisierung des externen Visualisierungsprozesses am sphärischen Display beschäftigt. Basierend auf einem Python Skript wurden die Dateneingabe und Vektordatenaufbereitung automatisiert. Anhand eines weiteren Skripts wurden die aufbereiteten Daten über ein ArcGIS Layerfile als Raster im Plattkartenformat 4096*2048 exportiert. (vgl. [SMU-17: 68])

4.3 Direkte Vektorvisualisierung

Das Ziel der direkten Vektorintegration am sphärischen Display ist es, Vektordaten direkt und entzerrt am Hyperglobus zu visualisieren. Wie bereits von Kristen J. erwähnt, ist der Prozess der direkten Entzerrung in der Laufzeit, bezogen auf die Rechenleistung - und Implementierung, komplex. (vgl. [KRI-12: 48]) Jedoch ist ein Vorteil, dass im Rahmen der direkten Integration ein Objektbezug möglich ist, wodurch sich Interaktions- und Animationsmöglichkeiten ergeben.

Die direkte, interne Visualisierung war auch der zweite Forschungsteil von Smutny T. Es wurde untersucht, wie die Vektordatenintegration am Hyperglobus automatisiert und optimiert werden kann. Basierend auf Python-Skripten, wurden Vektordaten direkt in die Globensoftware geladen und am Hyperglobus visualisiert. Dieser Ansatz wurde von ihm für

die Thematik von „Real-Time Flugdaten“ umgesetzt, wobei die Vektordaten im Rahmen der Skriptlaufzeit aus einem Textfile generiert wurden. (vgl. [SMU-17: 79-81])

Auch Liem J. hat sich im Zuge seiner Masterarbeit mit Vektorformaten und Visualisierungsmethoden am sphärischen Display beschäftigt. Folgende Varianten wurden von ihm im Rahmen der direkten Vektorvisualisierung definiert und umgesetzt.

- **Intern – Synchron – 2D – 2D Modell**

Variante 1 beschreibt einen internen, synchronen Ansatz. Hierbei wird in einem zweidimensionalen Raum ein 2D Modell aus Vektordaten produziert. Im Rahmen der Umsetzung werden sowohl OpenGL ES Shader als auch Methoden der sphärischen Geometrie angewandt. (vgl. [LIE-12: 51-52])

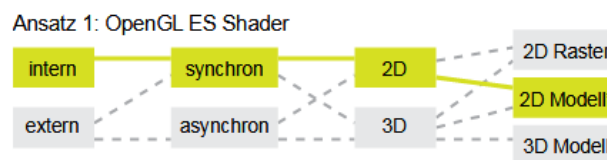


Abbildung 24: Vektorintegration am sphärischen Display - Variante 1 ([LIE-12: 50])

- **Intern – Synchron – 2D – 2D Modell / Intern – Synchron – 3D – 3D Modell**

Sowohl in Ansatz 2a als auch 2b wird die Umsetzung der Vektordatenvisualisierung dieser Ansätze als 3D Modell im dreidimensionalen Raum auf einer Kugeloberfläche visualisiert. Bei einer Methode wird das Netz intern im Speicher abgelegt und verwaltet, bei der zweiten extern als *.mesh Datei abgespeichert. Eine weitere Unterscheidung findet durch die externe vs. interne Implementierung und den Zeitpunkt (asynchron/synchron) der Visualisierung statt. (vgl. [LIE-12: 52])

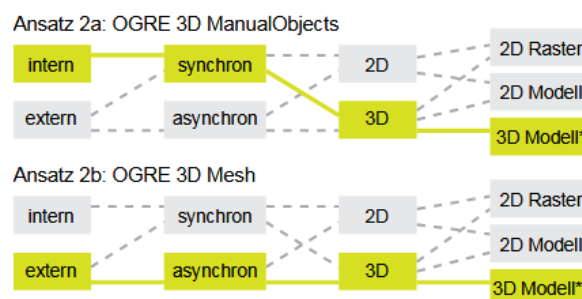


Abbildung 25: Vektorintegration am sphärischen Display -Variante 2 ([LIE-12: 50])

• Extern – asynchron – 3D – 2D Modell

Ansatz 3a beschreibt einen asynchronen, externen Prozessablauf. Auch bei diesem Prozess findet die Generation der Vektordaten, wie bereits bei 2a und 2b, im 3D Raum aus CSG (Constructive Solid Geometry) statt. Zusätzlich ist auch der Zeichenträger dieses Prozesses eine Kugeloberfläche. Der Unterschied besteht darin, dass das Endprodukt ein zweidimensionales Basisraster ist. (vgl. [LIE-12: 52])

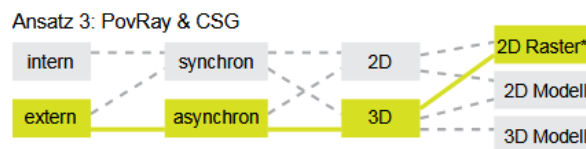


Abbildung 26: Vektorintegration am sphärischen Display -Variante 3 ([LIE-12: 50])

Die Anzahl und Varietät der genannten Methoden soll aufzeigen, welches Potential in Ansätzen der direkten Visualisierung von Vektordaten am sphärischen Display liegt. Die Eigenschaften und das Format der jeweiligen Vektordateien sind wesentliche, ausschlaggebende Faktoren, welche hinsichtlich einer Visualisierung am taktilen Hyperglobus hinterfragt und diskutiert werden müssen. An jenem Schwerpunkt möchte diese Masterarbeit nun im Rahmen ihrer Forschungsfragen anknüpfen.

4.4 Vergleich der Visualisierungsmethoden

Die folgende Tabelle liefert eine Gegenüberstellung der in Kapitel 4.1 – 4.3 genannten Visualisierungsmethoden und Dateiformate für den taktilen sphärischen Hyperglobus.

	Vorteile	Nachteile
Raster direkt	• schnelle Erfassung in der Globensoftware • keine Programmierkenntnisse	• Enormer Speicherbedarf • Fehlender Objektbezug • keine Animations- und Interaktionsmöglichkeiten • Verzerrungsproblematik
Raster indirekt	• schnelle Erfassung in der Globensoftware • keine Programmierkenntnisse • Manipulation/Erweiterung von Vektordaten	• Enormer Speicherbedarf • Fehlender Objektbezug • keine Animations- und Interaktionsmöglichkeiten • Verzerrungsproblematik oder manuelle Entzerrung
Vektor indirekt	• Geringer Speicherplatz • Automatisierte Entzerrung • Objektbezug • Animation • Interaktivität	• Implementierung der Vektordatenfunktionalität • Programmierkenntnisse • hohe Rechenleistung bei direkter Entzerrung

Tabelle 8: Vergleich der Visualisierungsmethoden am Hyperglobus (eigene Darstellung)

5 Analyse umgesetzter Global Stories

Seitdem die HRG Forschungsgruppe existiert, werden im Zuge von Kundenaufträgen, oder auch in der Lehrveranstaltung Multimediatechnologie und Geokommunikation, Global Stories am sphärischen Display umgesetzt. Um für den Hyperglobus essentielle Elemente zu definieren, werden nun umgesetzte Global Stories analysiert. Die genannten Visualisierungsmethoden selbst wurden bereits in Kapitel 4.1 – 4.3 näher beleuchtet.

Die folgenden Unterkapitel untersuchen gewählte Global Stories nun anhand der in Kapitel 2.2 genannten zentralen Elemente für eine Visualisierung am sphärischen Display.

- **Graphische Gestaltung**
- **Verzerrung**
- **Globenwürdigkeit**
- **Thematik**
- **Animation**
- **Interaktion**

Die Auswahl der Stories wurde dahingehend getroffen, dass alle in Kapitel 3.2 genannten geometrischen Primitiva in diesen eingesetzt wurden. Dabei handelt es sich um punkthafte-, linienhafte- und flächenhafte Geodaten. Zusätzlich sollen die genannten Global Stories alle in Kapitel 4 genannten Methoden der Geovisualisierung am sphärischen Display abdecken.

5.1 Earthquakes 1914 – 2014

In dieser Global Story wurden globale Erdbebenereignisse über einen Zeitraum visualisiert. Hierzu wurden Erdbeben, mit einer Magnitude größer 6, am sphärischen Display über einen Zeitraum von 100 Jahren abgebildet. Endnutzer sehen die Erdbeben über den Zeitverlauf als Punktobjekt aufscheinen. Die Größe jener Punktobjekte variiert je nach Erdbebenstärke. Über eine Legende wurde der Zeitverlauf eingebunden. (siehe Abb. 27)

- **Visualisierungsmethode:** OmniSuite Punktformat (*intern/synchron*) + Hintergrundkarte (*extern/asynchron*) + Legende im Rasterformat (*extern/asynchron*)
- **Graphische Gestaltung:** Die Größe von Erdbebenereignissen wird anhand der Magnitude automatisch kategorisiert und es wird eine Legende für den Zeitverlauf eingeblendet.
- **Verzerrung:** Die Punktobjekte werden automatisch in OmniSuite entzerrt.
- **Globenwürdigkeit:** Raumbezug und globale Datenverfügbarkeit der Thematik.
- **Thematik:** Physische Geographie
- **Animation:** Automatisierte Animation der Punktobjekte zu einem definierten Zeitpunkt mit FadeIn- und FadeOut-Effekt.
- **Interaktion:** Jedes Punktobjekt kann durch die direkte Vektorintegration direkt angesprochen werden. (z.B. Erdbebenzeitpunkt oder Ort des Erdbebens)

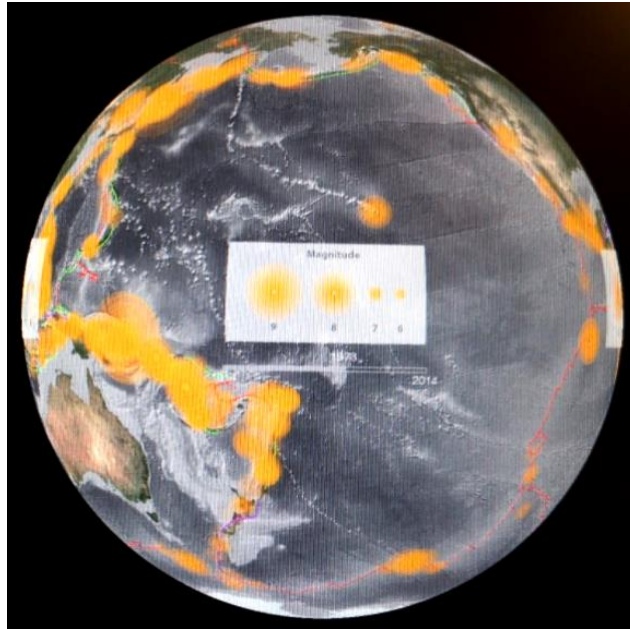


Abbildung 27: Earthquakes 1914 – 2014 ([KRI-14])

5.2 Global Sea Level Rise

Auch der globale Meeresspiegelanstieg ist eine geeignete Thematik für das sphärische Display und wurde auf diesem umgesetzt. Um den globalen Anstieg des Meeresspiegels in Küstenregionen zu visualisieren, wurden Polygone verwendet. Da ein Anstieg von wenigen Metern schwierig auf einem globalen Maßstab zu erkennen ist, wurde mit der Zoom-Funktion gearbeitet. Diese ermöglicht eine bis zu 6-fache Vergrößerung. Eine Legende gibt zusätzlich Information über die Höhe des Anstieges. (siehe Abb. 28)

- **Visualisierungsmethode:** Polygone durch indirekte Rastervisualisierung (*extern/asynchron*) + Hintergrundkarte (*extern/asynchron*) + Legende im Rasterformat (*extern/asynchron*)
- **Graphische Gestaltung:** Anstieg des Meeresspiegels wird in blau (Wasser) visualisiert. Zusätzlich wird über eine Legende die Höhe des Meeresspiegelanstiegs eingeblendet.
- **Verzerrung:** Flächenobjekte für eine Visualisierung am HG nicht entzerrt werden
- **Globenwürdigkeit:** Raumbezug und globale Datenverfügbarkeit der Thematik.
- **Thematik:** Physische Geographie
- **Animation:** Die Sichtbarkeit der Polygone wurde, abhängig von der Höhe des Anstiegs, animiert.
- **Interaktion:** Durch die indirekte Rastervisualisierung ist eine interaktive Auswahl der Höhe des globalen Meeresspiegelanstiegs nicht möglich.



Abbildung 28: Global Story Sea Level Rise (IKRI-18)

5.3 Tiefseekabel

Auch das Thema der globalen Tiefseekabel war für eine Umsetzung am sphärischen Display geeignet. Die Tiefseekabel lagen im Geometrietyp Linie vor. Durch eine Analyse im Vorfeld wurden besondere Unterwasserseekabel ausgewählt und deren Verlauf am sphärischen Display eingeblendet. Per Tiefseekabel wurden in einer Legende wichtige Detailinformationen eingeblendet. Zusätzlich wurden für ein Kapitel alle globalen Tiefseekabel manuell animiert. (Abb. 30)

- **Visualisierungsmethode:** Linien (Tiefseekabel) durch indirekte Rastervisualisierung (*extern/asynchron*) + Hintergrundkarte (*extern/asynchron*) + Legende im Rasterformat (*extern/asynchron*)
- **Graphische Gestaltung:** Anstieg des Meeresspiegels wird in blau (Wasser) visualisiert. Zusätzlich wird über eine Legende die Höhe des Meeresspiegelanstiegs eingeblendet.
- **Verzerrung:** Linienobjekte wurden zur Visualisierung am HG nicht entzerrt.
- **Globenwürdigkeit:** Raumbezug und globale Datenverfügbarkeit der Thematik.
- **Thematik:** Humangeographie (Verkehr)
- **Animation:** Der Verlauf von Tiefseekabeln wurde manuell in einer Bildbearbeitungssoftware animiert.
- **Interaktion:** Durch die indirekte Rastervisualisierung ist eine interaktive Auswahl nach Betreiber/Alter/Länge/Übertragungsgeschwindigkeit der Tiefseekabel nicht möglich.

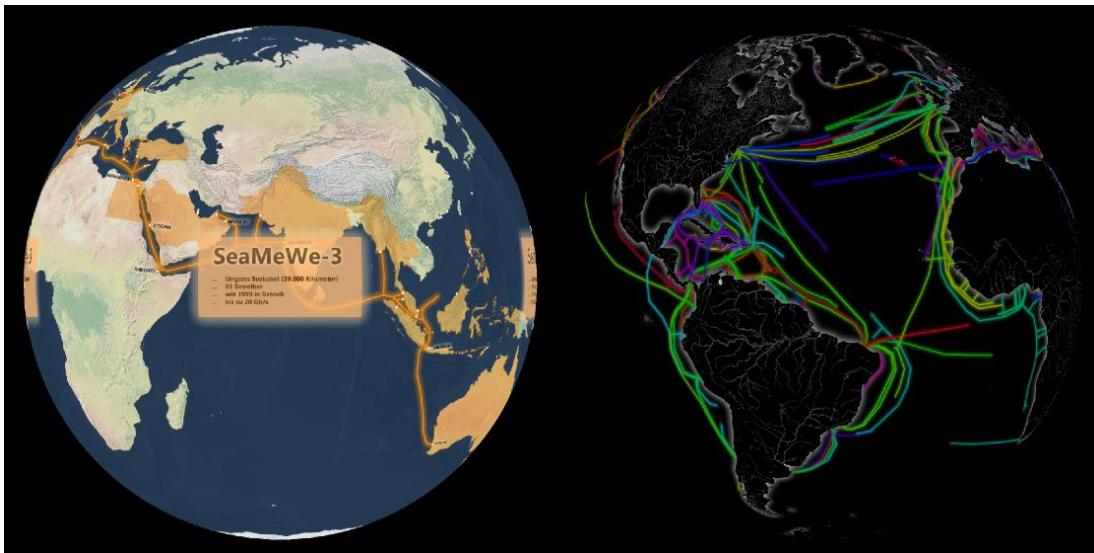


Abbildung 29: Global Story Tiefseekabel ([KIE-18])

5.4 Diskussion der analysierten Global Stories

Durch die drei analysierten Global Stories wird verdeutlicht, welche Einschränkung durch die asynchrone, externe Visualisierung von Vektorobjekten am Hyperglobus besteht. Die Elemente der graphischen Gestaltung, Globenwürdigkeit und Thematik für den Hyperglobus wurden allesamt von allen Global Stories erfüllt.

Jedoch gab es in den Global Stories Differenzen bezüglich der Einbindung von Animation und Interaktion. Der Unterschied liegt in der automatischen bzw. manuellen Animation der Geoobjekte, da die manuelle Variante mit einem größeren Arbeits- und Zeitaufwand verbunden ist. Interaktivität kann (zum momentanen Zeitpunkt) nur in Datensätzen, visualisiert durch die interne, synchrone Methode, eingesetzt werden. Der Grund ist, dass durch diese Methode der Objektbezug von Vektorformaten nicht durch eine Konvertierung ins Rasterformat verloren geht.

6 Vektorformate in der Geoinformation

Der folgende Abschnitt der Masterarbeit beschäftigt sich nun mit der Vielfalt an Vektorformaten, welche in der Geoinformation Einsatz finden. In Kapitel 2 bis 4 wurde bereits die Bedeutsamkeit von Vektorformate am sphärischen Display diskutiert. Nun gilt es, bestehende Vektorformattypen im GIS-Bereich zu differenzieren und für ihre Eignung für eine zukünftige Visualisierung am taktilen Hyperglobus zu beurteilen. Ziel ist es, die Vor- und Nachteile der untersuchten Formate gegenüberzustellen und darauf basierend eine Formatauswahl für die folgende theoretische Analyse in Kapitel 7 zu bilden.

Die Auswahl der in den folgenden Unterkapitel beschriebenen Vektorformattypen basiert auf dem „OpenGIS® Implementation Standard for Geographic Information - Simple feature access - Part 1: Common architecture“ (ISO 19125) ([OGC-11]) und der Verbreitung genannter Formate in der Geoinformation. Es wird auch untersucht, welche der spezifizierten Formate im Bereich von Open Data Plattformen angeboten werden, da es sich bei Geodaten im GIS-Bereich idealerweise um frei verfügbare Daten handelt. Die Verbreitung von Vektorformaten auf Open-Source Plattformen wird hierfür in Kapitel 6.2 untersucht.

6.1 Simple Feature Geometrien

„A feature is an "abstraction of real world phenomena" (ISO 19101); it is a geographic feature if it is associated with a location relative to the Earth. So a digital representation of the real world may be thought of as a set of features.“([OGC-12: 6])

Im Rahmen der Simple Feature Geometrien wird unter ISO 19107 eine allgemeine Architektur von Geometrien einfacher, räumlicher Objekte beschrieben. In der Spezifikation werden durch den Standard die nun folgenden Geometrietypen definiert. (Vorweg wird angemerkt, dass *MultiFeatures* zu *SingleFeatures* getrennt werden können.)

6.1.1 Punkt / Multipunkt [Point/MultiPoint]

Ein Punkt nach ISO 19107 ist ein 0-dimensionales Objekt, welches zu den geometrischen Primitiva zählt. Er besitzt X/Y- Koordinaten, welche seine Lage im Koordinatensystem beschreiben. Weiters kann ein Punkt auch attributiv mit Z-(3D) und M-Koordinaten (Werte im Punkt gemessen) ausgestattet sein. Als Multipunkt-Geometrie [MultiPoint] zählt eine Geometrie-Sammlung von Punkten, wobei diese frei von jeglicher semantischen Logik ist. Die Grenze eines Punktes oder Multipunktes wird von der leeren Menge gebildet. (vgl. [OGC-11:21], [LIE-12: 21])

6.1.2 Linie [Curve]

Die Linie, welche genauso als geometrische Primitiva gilt, wird in der Objektklasse Curve visualisiert. Diese ist ein eindimensionales Objekt, welche in Form einer Punktsequenz mit dem Subtyp Curve gespeichert wird. Dies bezieht sich auf die Interpolationsart der Linie zwischen Punkten. LineString ist die einzige Unterklasse vom Subtyp Curve, welche im OGC Standard ISO 19107 (Open Geospatial Consortium) definiert ist. Diese beschreibt die lineare Interpolation zwischen zwei Punkten. Die folgenden Geometrietypen in Tabelle 9 wurden in den Unterklassen LineString/MultiLineString festgelegt. Wie in der Objektklasse der MultiPoints handelt es sich bei MultiLineString um eine Sammlung von LineStrings (vgl. [OGC-11: 21])

Typ der Unterklasse LineString nach ISO 19107	Beispiel	Typ der Unterklasse MultiLineString nach ISO 19107	Beispiel
<i>simple LineString</i> Startpunkt s und Endpunkt e nicht ident und keine geometrische Überlappung		<i>simple MultiLineString</i> Startpunkt e1 und Endpunkt s2 sind ident und keine geometrische Überlappung	
<i>non-simple LineString</i> Startpunkt s und Endpunkt e nicht ident und geometrische Überlappung		<i>non-simple MultiLineString</i> Element1: Startpunkt s und Endpunkt e nicht ident und geometrische Überlappung + Element2: Start- und Enpunkt ident und keine überlappende Geometrie	
<i>simple, closed LineString</i> LinearRing Start- und Enpunkt ident und keine überlappende Geometrie			
<i>non-simple, closed LineString</i> Start- und Enpunkt ident mit überlappenden Geometrie		<i>non-simple closed MultiLineString</i> Startpunt s1/e1 und Endpunkt s2/e2 sind ident	

Tabelle 9: LineString/MultiLineString nach ISO 90107 ([OGC-11: 23-25])

6.1.3 Flächen [Surface]

Flächen werden im Zuge der Simple Feature Geometrien in der Objektklasse Surface abgebildet. Diese lässt sich in zwei wesentliche Unterklassen gliedern, Polygon und PolyhedralSurface. Bei letzterem handelt es sich um eine dreidimensionale Verbindung von einzelnen Flächen entlang ihrer gemeinsamen Oberfläche. Für die Forschung im Rahmen dieser Masterarbeit steht die Unterklasse Polygon im Vordergrund. Ein Polygon selbst ist

ein geometrisches Objekt im 2D-Raum, welches zumindest aus einer äußeren Grenze plus zusätzlich aus inneren Ringen besteht. Dabei werden die äußeren und inneren Polygongrenzen von einem in Tabelle 10 abgebildeten LinearRing gebildet. Ein MultiPolygon ist wiederum ein MultiSurface bestehend aus mehreren Polygonen. Dabei gilt, dass Innenbereiche der Polygone sich nicht überschneiden dürfen, sich Polygongrenzen nicht schneiden und nur an endlich vielen Punkten berühren. (vgl. [OGC-11: 26-32])

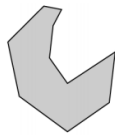
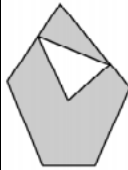
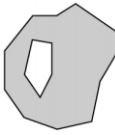
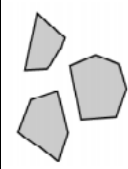
Typ der Unterklasse Polygon nach ISO 19107	Beispiel	Typ der Unterklasse MultiPolygon nach ISO 19107	Beispiel
<i>1 ring Polygon</i> Polygon ohne innere Ringe/Löcher + bestehend aus einem regulär geschlossenen, verbundenen Punktset.		<i>2 Polygon element</i>	
<i>2 ring Polygon</i> Polygon mit innerem Ring/Loch + bestehend aus einem regulär geschlossenen, verbundenen Punktset.		<i>2 Polygon element</i>	
<i>3+ ring Polygon</i> Polygon mit mehreren inneren Ringen/Löchern + bestehend aus einem regulär geschlossenen, verbundenen Punktset.		<i>3 Polygon element</i>	

Tabelle 10: Polygon/MultiPolygon nach ISO 90107 ([OGC-11: 27-32])

6.1.4 Methoden der Simple Feature Specification

In einem weiteren Schritt wurden in ISO 19107 diverse Standards festgelegt um die räumliche Beziehung zwischen Geoobjekten festzulegen und ihre Geometrie wie auch Topologie zu überprüfen (Tabelle 11). (vgl. [OGC-11: 36-40])

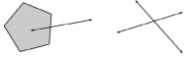
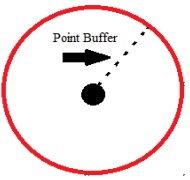
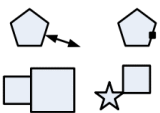
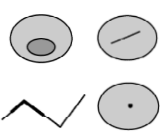
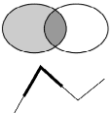
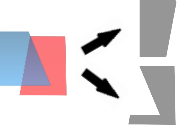
räumliche Beziehung	Beispiel	räumliche Analyse	Beispiel
Crosses P/L, P/A, L/L, L/A		Buffer Gibt ein geometrisches Objekt zurück, welches alle in der Buffer-Distanz enthalten geometrischen Objekte enthält	
Touches A/A, L/L, L/A, P/A, P/L		Intersection Die Methode der Intersection berechnet den geometrischen Schnittpunkt aus überlappenden Geoobjekten. (Überschneidung)	
Within A/A, A/L, L/L, P/P		Union Der Output ist eine Vereinigung von verschiedenen Geoobjekten (Input)	
Overlaps A/A, L/L, P/P		Difference Teile von Geoobjekten (Features), die sich NICHT überschneiden, werden in ein neues Feature geschrieben (Output)	

Tabelle 11: Auswahl räumlicher Beziehungen/Analysen nach ISO 90107 ([OGC-11: 36-40])

6.1.5 Vektorformate im Rahmen der Simple Feature Specification

Nachdem die Theorie hinter Vektorformaten und die Methoden der Simple Feature Spezifikation nun erläutert wurden, stellt sich die Frage, welche Vektorformate sich, aufgrund der genannten Voraussetzungen, für eine praktische Analyse eignen.

Hierzu wurden von der OGR (OGR Simple Features Library), unter dem Ansatz der Simple Features Geometry, kompatible Vektorformate gewählt. Diese Auswahl basiert und orientiert sich an der OpenGIS SimpleFeature Spezifikation des OGC. (vgl. [GDA-17])

Tabelle 12 liefert hierzu einen Überblick über geeignete Vektorformate im Zuge der OGR. Insgesamt werden in dieser über 90 Vektorformate genannt. *Blau hinterlegte* Vektorformate entsprechen der Simple Feature Spezifikation. Die Analyse aller Formate überschreitet den Umfang dieser Arbeit jedoch deutlich. Daher wird in Kapitel 6.2 zusätzlich die Verbreitung von Vektorformaten auf einem globalen und österreichischen Webportal überprüft. Basierend darauf wird eine erste Auswahl an Formaten getroffen.

Format Name	Code	Format Name	Code
Aeronav FAA files	<i>AeronavFAA</i>	Mapinfo File	<i>MapInfo File</i>
AmigoCloud API	<i>AmigoCloud</i>	Microstation DGN v7	<i>DGN</i>
ESRI ArcObjects	<i>ArcObjects</i>	Microstation DGN v8	<i>DGNv8</i>
Arc/Info Binary Coverage	<i>AVCBin</i>	Access MDB (PGeo and Geomedia capable)	<i>MDB</i>
Arc/Info .E00 (ASCII) Coverage	<i>AVCE00</i>	Memory	<i>Memory</i>
Arc/Info Generate	<i>ARCGEN</i>	MongoDB	<i>MongoDB</i>
Atlas BNA	<i>BNA</i>	Mapbox Vector Tiles	<i>MVT</i>
AutoCAD DWG	<i>CAD</i>	MySQL	<i>MySQL</i>
AutoCAD DWG	<i>DWG</i>	NAS - ALKIS	<i>NAS</i>
AutoCAD DXF	DXF	NetCDF	<i>netCDF</i>
Carto	<i>Carto</i>	Oracle Spatial	<i>OCI</i>
Cloudant / CouchDB	<i>Cloudant</i>	ODBC	<i>ODBC</i>
CouchDB / GeoCouch	<i>CouchDB</i>	MS SQL Spatial	<i>MSSQLSpatial</i>
Comma Separated Value (.csv)	<i>CSV</i>	Open Document Spreadsheet	<i>ODS</i>
OGC CSW (Catalog Service for the Web)	<i>CSW</i>	OGDI Vectors (VPF, VMAP, DCW)	<i>OGDI</i>
Czech Cadastral Exchange Data Format	<i>VFK</i>	OpenAir	<i>OpenAir</i>
DB2 Spatial	<i>DB2ODBC</i>	ESRI FileGDB	<i>OpenFileGDB</i>
DODS/OPeNDAP	<i>DODS</i>	OpenStreetMap XML and PBF	<i>OSM</i>
Google Earth Engine Data API	<i>EEDA</i>	PCI Geomatics Database File	<i>PCIDSK</i>
EDIGEO	<i>EDIGEO</i>	Geospatial PDF	<i>PDF</i>
ElasticSearch	<i>ElasticSearch</i>	PDS	<i>PDS</i>
ESRI FileGDB	<i>FileGDB</i>	Planet Labs Scenes API	<i>PLScenes</i>
ESRI Personal GeoDatabase	<i>PGeo</i>	PostgreSQL SQL dump	<i>PGDump</i>
ESRI ArcSDE	<i>SDE</i>	PostgreSQL/PostGIS	<i>PostgreSQL/PostGIS</i>
ESRIJSON	<i>ESRIJSON</i>	EPIInfo .REC	<i>REC</i>
ESRI Shapefile / DBF	ESRI Shapefile	S-57 (ENC)	<i>S57</i>
FMEObjects Gateway	<i>FMEObjects Gateway</i>	SDTS	<i>SDTS</i>
GeoJSON	GeoJSON	SEG-P1 / UKOOA P1/90	<i>SEGUKOOA</i>
Géoconcept Export	<i>Geoconcept</i>	SEG-Y	<i>SEG-Y</i>
Geomedia .mdb	<i>Geomedia</i>	Selafin/Seraphin format	<i>Selafin</i>
GeoPackage	<i>GPKG</i>	Norwegian SOSI Standard	<i>SOSI</i>
GeoRSS	<i>GeoRSS</i>	SQLite/SpatiaLite	<i>SQLite</i>
Google Fusion Tables	<i>GFT</i>	SUA	<i>SUA</i>
GML	GML	SVG	SVG
GMLAS	<i>GMLAS</i>	Storage and eXchange Format	<i>SXF</i>
GMT	<i>GMT</i>	U.S. Census TIGER/Line	<i>TIGER</i>
GPSTBabel	<i>GPSTBabel</i>	TopoJSON	<i>TopoJSON</i>
GPX	<i>GPX</i>	UK .NTF	<i>UK. NTF</i>
GRASS Vector Format	<i>GRASS</i>	VDV-451/VDV-452/IDF	<i>VDV</i>
GPSTrackMaker (.gtm, .gtz)	<i>GPSTrackMaker</i>	VRT - Virtual Datasource	<i>VRT</i>
Hydrographic Transfer Format	<i>HTF</i>	Walk	<i>Walk</i>
Idrisi Vector (.VCT)	<i>Idrisi</i>	WAsP .map format	<i>WAsP</i>
Informix DataBlade	<i>IDB</i>	OGC WFS (Web Feature Service)	<i>WFS</i>
INTERLIS	<i>"Interlis 1" and "Interlis 2"</i>	OGC WFS 3.0 (Web Feature Service) (experimental)	<i>WFS3</i>
INGRES	<i>INGRES</i>	MS Excel format	<i>XLS</i>
JML	<i>OpenJUMP .jml</i>	MS Office Open XML spreadsheet	<i>XLSX</i>
KML	KML	X-Plane/Flightgear aeronautical data	<i>XPLANE</i>
LIBKML	<i>LIBKML</i>		

Tabelle 12: OGR Vektorformate konform der Simple Feature Spezifikation (blau hinterlegt) (OGR-o.J.)

6.2 Verbreitung von Vektorformaten auf Open-Data Plattformen

Um eine für diese Forschungsarbeit überschaubare Menge an geeigneten Vektorformattypen zu wählen, wird nun in einem weiteren Schritt die Verfügbarkeit der in Kapitel 6.1.5 genannten Formate auf Open-Source Webportalen überprüft.

Allgemein handelt es sich bei Geodaten, welche im Bereich der Geoinformation eingesetzt werden, idealerweise um frei (kostenlose) verfügbare Quellen. Daher werden für die Analyse des Formataufkommens sowohl eine lokale, österreichische (data.gv.at) und eine globale Plattform (datacatalog.worldbank.org) untersucht. Auf diesen Datenportalen sind frei verfügbare Datensätze im lokalen als auch globalen Kontext gesammelt und ohne Lizenz oder Kosten zum Download bereitgestellt.

Analysiert man nun die Verfügbarkeit von Vektorformaten nach ihrer Anzahl auf der lokalen Plattform (Tabelle 13), so wird das Shapefile (*.shp) am Webportal am öftesten zum Geodaten austausch angeboten, gefolgt von GeoJSON (*.geojson) und KML(*kml).

Vektorformat	Anzahl
Shapefile (*.shp)	667
(Geo)JSON (*.geojson)	564
KML (*.kml)	389
KMZ (*.kmz) **	234
GML (*.gml)	179
SVG (*.svg)	154
DXF (*.dxf)	3

Tabelle 13: Vektorformate auf der Plattform data.gv.at (IDAT-o.J.)

*** einige der Dateitypen im KML Format werden parallel auch im KMZ Format zur Verfügung gestellt, weswegen die Listung des KMZ Formates zu vernachlässigen ist*

Blickt man auf die Website der Worldbank, so werden globale Datensätze im Portal unter anderem in den folgenden Formattypen angeboten (Tabelle 14). (vgl. [DAT-18])

Globaler Datensatz	Format	Link
Wind Masts Inventory	GeoJSON	https://datacatalog.worldbank.org/dataset/world-wind-masts-inventory-2017
Solar Stations Inventory	GeoJSON	https://datacatalog.worldbank.org/dataset/world-solar-stations-inventory-2017
Night Light Annual Composite	KML	https://datacatalog.worldbank.org/dataset/world-night-light-annual-composite-2015
World Perennial Water Courses	SHP	https://datacatalog.worldbank.org/dataset/world-perennial-water-courses
World Heritage List	KML	https://datacatalog.worldbank.org/dataset/world-heritage-sites
Agricultural Practices Phosphorous	KML, GeoJSON, GML, Shapefile	https://datacatalog.worldbank.org/dataset/world-external-geospatial-platforms-food-agriculture
Mean green water savings though agricultural/industrial commodities	KML, GeoJSON, GML, Shapefile	https://datacatalog.worldbank.org/dataset/world-external-geospatial-platforms-food-agriculture
Soil Classifications	SHP	https://datacatalog.worldbank.org/dataset/soil-classifications
Major River Basins of the World	SHP	https://datacatalog.worldbank.org/dataset/major-river-basins-world

Tabelle 14: Vektorformate auf der Plattform datacatalog.worldbank.org (IDAT-18)]

Die Analyse von Vektorformaten auf lokalen und globalen Portalen zeigt, dass alle zur Verfügung gestellten Formate im Rahmen der OGR Vektorformate (Tabelle 12) genannt werden. Damit wird die theoretische und praktische Formatevaluation in Kapitel 7 und 8 mit Vektorformaten durchgeführt, welche auf der Simple Feature Geometrie Spezifikation aufbauen.

Vorweg ist ein weiterer Vektordateityp, welcher in Kapitel 5.4.2 beschrieben wird das WKT-Format (Well-Known Text). Obwohl es nicht von OGR (vgl. [OGR-o.J]) gelistet wird, handelt es sich hierbei um jenes Format, welches in den OGC Spezifikationen unter der internationalen Norm *ISO 19125: 2014* „*Geographic information – Simple feature access – Part 1: Common architecture*“ beschrieben und somit als der OGC Standard gehandelt wird. (vgl. [OGC-15a])

Im Gegenzug wird das DXF (Drawing Interchange Format) nicht näher analysiert. Es ist ein kommerzielles, von Autodesk entwickeltes Vektorformat. Es dient der Dateninteroperabilität zwischen CAD (Computer-Aided-Design) und anderen Programmen. (vgl. [WHA-18]) Es wird in technischen Branchen der Vermessung, Bauingenieurwissenschaft, Architektur und Weiteren eingesetzt, um technische Zeichnungen (2D) und Modellierungen (3D) durchzuführen. Bezogen auf die Thematik dieser Masterarbeit kann bereits an diesem Punkt gesagt werden, dass es als Format nicht für eine Visualisierung von globalen Vektordaten am Hyperglobus geeignet ist.

7 Theoretische Formatanalyse

Bevor die theoretische Analyse der Vektorformate durchgeführt wird, müssen die Beurteilungskriterien definiert werden. Folgend genannte Punkte werden, basierend auf den Forschungsfragen und den in Kapitel 2 genannten zentralen Elementen, zur Beurteilung der gewählten Vektorformate herangezogen:

- **Geometrie** – Welche Geometrietypen kann das Format abbilden?
- **Raumbezug** – Besteht ein geographischer Raumbezug der Objekte?
- **Graphisches Erscheinungsbild** – Erweiterung der Geometrie durch graphische Gestaltung?
- **Animation** – Ist eine Animation von Objekten möglich?
- **Interaktion** – Können Geoobjekte objektbasiert interaktiv angesprochen werden?

Basierend auf jenen Kriterien wird in Kapitel 7.5 ein Vergleich und eine Beurteilung der Vektorformate aufgestellt. Daraus resultierend wird eine Auswahl für die praktische Formatanalyse in Kapitel 8 getroffen.

7.1 Formatgrundlagen

Bevor ab Kapitel 7.3 näher auf von der Autorin gewählte kommerzielle und offene Vektorformate eingegangen wird, werden grundlegende Formatstrukturen und Möglichkeiten der Stildefinitionen erläutert.

7.1.1 XML

„XML ist eine Methode zur Speicherung strukturierter Daten in Textform[...]XML eignet sich zur Speicherung und Austausch von Daten, die sich sinnvoll als Text dargestellt werden können. Zur Speicherung großer Bitfolgen, wie sie zum Beispiel bei Multimediatechniken wie Bildern, Ton oder auch Videos vorkommen, ist XML ungeeignet.“([WEB-03])

XML, auch Extensible Markup Language genannt, ist eine Auszeichnungssprache. Sie wird zur Darstellung strukturierter Datensätze im Textformat herangezogen. (vgl. [WEB-03: 1]) Diese Dateistruktur wird von mehreren, in diesem Kapitel genannten Vektorformaten, verwendet.

Allgemein können Daten und ihre Struktur im XML-Format abgespeichert werden. Dabei werden Informationen in Tags (Elementen) strukturiert. (vgl. [WEB-03: 1-3])

In jeder XML-Datei findet man eine syntaktische (Form) und semantische Ebene (Bedeutung), wobei im semantischen Teil sowohl Elemente als auch Attribute zu finden sind. Wie in Abbildung 30 zu sehen, werden Elemente immer als sogenannte „Tags“

verwendet. Dabei bauen Elemente einer XML-Datei ineinander auf, Konstruktionen wie in Beispiel 2 aus Abbildung 30 sind nicht zulässig. Zusätzlich spielt in XML auch die Groß- und Kleinschreibung eine Rolle, die Sprache ist somit CaseSensitiv. (vgl. [KML-o-J.]

1	<xml version="1.0" encoding="UTF-8" standalone="yes">	1	<xml version="1.0" encoding="UTF-8" standalone="yes">
2	<Straße Typ="Ortsstraße">	2	<Straße Typ="Ortsstraße">
3	<LineColor>	3	<LineColor>
4	<Farbe>schwarz</Farbe>	4	<Farbe>schwarz
5	</LineColor>	5	</LineColor>
6	<LineWidth>2</LineWidth>	6	</Farbe>
7	<LineType>MultiLineString</LineType>	7	<LineWidth>2
8	</Straße>	8	<LineType>MultiLineString
		9	</LineWidth>
		10	</LineType>
		11	</Straße>

KORREKTER AUFBAU

INKORREKTER AUFBAU

Abbildung 30: Korrekter vs. inkorrektcr Tagaufbau in XML (eigene Darstellung)

Hauptsächlich wird XML - im Gegensatz zu HTML, der Standardsprache von Websites - zur Repräsentation von Daten angewendet. Die folgenden, in Tabelle 15 genannten Punkte sollen wesentliche Vor- und Nachteile aufzeigen. Gegenüber den bereits genannten Vorteilen steht der die tagbasierte Struktur des XML-Formates. (vgl. [EVJ-07: 5])

	XML
Vorteile	leicht lesbare Struktur
	unterstützt von vielen Plattformen
	oft im Rahmen von OpenStandards eingesetzt
	viele XML Tools aufgrund der breiten Reichweite
Nachteile	Tagbasiert (Speicherbedarf)

Tabelle 15: Vor- und Nachteile von XML ([EVJ-07: 5])

7.1.2 JSON

JSON ist ein populäres, offenes Datenaustauschformat zwischen Server und Client und wurde von JavaScript abgeleitet. Aus diesem Grund sind seine Objekte in der Struktur ähnlich jenen in JavaScript. Diese sind jedoch in keiner Art von JavaScript abhängig. Der große Vorteil des JSON-Formates ist seine Unabhängigkeit. So wird das JSON-Format von nahezu allen gängigen Programmiersprachen unterstützt. Dazu zählen Ruby, Python, C++, C#, Java und viele mehr. Dieser Aspekt ist gegenüber anderen Dateitypen als großer Pluspunkt zu zählen. (vgl. [SRI-13: 15])

Wurde noch vor kurzer Zeit die Auszeichnungssprache XML (auf welcher unter anderem das KML-, GML- und SVG-Format basiert) als Standard für den Datentransfer in Webbrowsern herangezogen, so steht nun das JSON-Format an dieser Stelle. Dies liegt daran, dass keine spezifische Implementierung notwendig ist, da das JSON-Parsing im Web-

Browser von der JavaScript Engine übernommen wird. Auch der Speicherbedarf von JSON ist gegenüber XML aufgrund der Dateistruktur geringer. (vgl. [SRI-13: 29])



Abbildung 31: JSON vs. XML Datenstruktur (eigene Darstellung)

Wie in Abbildung 31 ersichtlich, basiert das JSON Format im Gegensatz zu XML auf einer besonderen Dateistruktur, auch das „Schlüssel-Wert“ Prinzip genannt. Dies bedeutet, dass es sich bei einer JSON-Datei um eine Sammlung aus Schlüsseln handelt, denen jeweils ein Wert zugewiesen wird. (vgl. [INT-o.J.])

7.2 Graphische Aufbereitung von Vektorformaten

“A cartographic style is the visual representation of geographic data. It is the culmination of a set of rules, applied to data that is rendered as an image. A cartographic stylesheet is the file that stores these rules. They can take many forms depending on the data, the software and the needs or preferences of the user.” ([CAR-o.J.])

Vektorformate, welche die Strukturen von Geodaten beschreiben, speichern grundsätzlich keine Stilinformation mit. Diese Stilinformation wird jedoch zwingend benötigt, um die visualisierten Geodaten auch korrekt im Zuge der kartographischen Gestaltungslehre graphisch aufzubereiten. Speziell für die thematische Visualisierung von Global Stories ist dies ein Muss. Um Vektoren graphisch zu gestalten, werden diese intern durch einen style-Tag oder extern durch Stylesheets erweitert. Zur graphischen Aufbereitung von Geodaten bieten sich, je nach Fragestellung und Umgebung, verschiedenen Möglichkeiten an.

Die Formate SVG, GML, KML, SVG, GeoJSON, WKT und das OmniSuite Punktformat werden über unter anderem über Stylesheets graphisch aufbereitet. Daher folgt im folgenden Kapitel eine Einführung in die Stylesheet-Sprache CSS.

7.2.1 CSS

Bei Cascadian Style Sheet (CSS) handelt es sich um eine Stylesheet-Sprache. Durch sie wird definiert, wie ein Dokument - oder im Fall von Geodaten - ein Objekt, visualisiert dargestellt wird. Wichtig ist, dass es sich beim CSS Dokument eines Vektordatensatzes nur um die **Stildefinition** der Objekte handelt. Die Idee dahinter ist es Objekte, basierend auf ihrer Spezifikation mit definierten Styles auszuweisen. Dabei besteht ein CSS Dokument immer aus einem Selektor (Bezeichnung des Objektes, auf welche sich die Regel bezieht) und aus Eigenschaften, welche dem jeweiligen Element zugewiesen werden. CSS ist nicht CaseSensitiv. (vgl. [HOW-17a])

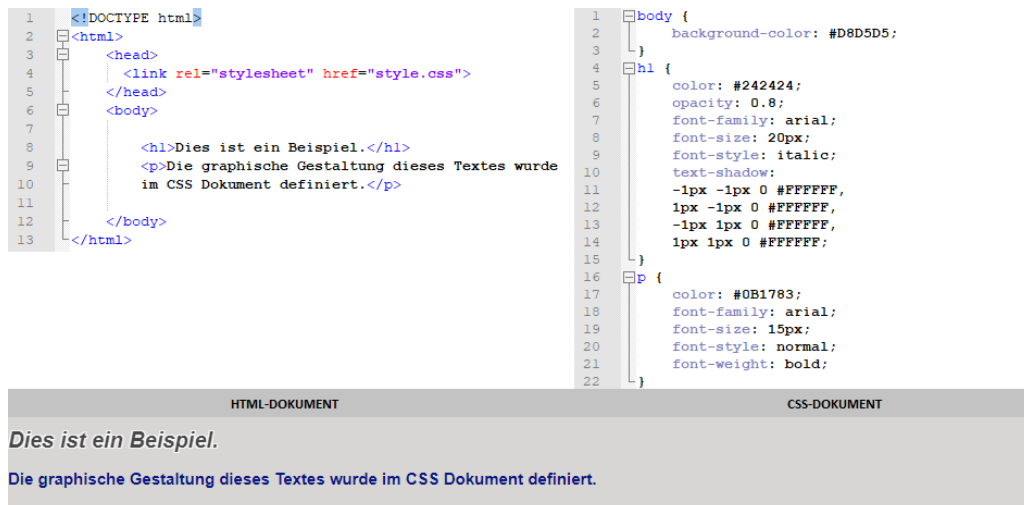


Abbildung 32: CSS Beispiel (eigene Darstellung)

7.3 Kommerzielle¹ Vektorformate

7.3.1 SHP

“A shapefile is a simple, nontopological format for storing the geometric location and attribute information of geographic features.” ([WHA-16])

¹ Der Begriff kommerziell bezieht im Zusammenhang mit dem Shapefile darauf, dass das Format von ESRI entwickelt wurde. Die Softwareprodukte dieses Unternehmens sind kostenpflichtig und nur mit gültiger Lizenz nutzbar. Der Standard des Shapefiles selbst ist öffentlich einsehbar und kostenfrei.

Das Shapefile (*.shp) ist ein Format für Geodaten und wurde von der Firma ESRI entwickelt. Es wurde in den frühen 90er Jahren mit ArcView GIS (Version 2) eingeführt. (vgl. [WHA-15])

Im Gegensatz zu anderen Vektordateiformaten handelt es sich beim Shapefile nicht nur um eine Datei, sondern um mindestens drei Elemente. Diese sind eine Haupt-, Index- und Attributdatei. In erster Linie werden die Geometrien eines jeden Shapefiles abgespeichert. Die Indexdatei hingegen unterstützt die interne Datenverarbeitung. Attribute eines Shapefiles werden in einer Tabelle vom Typ dBase gespeichert, wobei jede Geometrie (Linie, Punkt, Fläche) je einen Tabelleneintrag besitzt. Ein oft mitgespeichertes, zusätzliches Element des Shapefiles ist die Projektionsdatei (*.prj) In dieser ist das Koordinatensystem des Shapefiles definiert. (vgl. [ESR-98: 6])

Geometrie

Das Shapefile unterstützt Geoobjekte des Geometrietyps Punkt, Linie und Fläche. Verglichen mit anderen Vektorformaten kann pro Shapefile immer nur ein Geometrietyp abgespeichert werden. Werden zur Umsetzung einer Thematik nun verschiedene Geometrietypen benötigt (zum Beispiel in der Raumplanung: Haltestelle, Siedlung, Straße), kann die räumliche Information nicht gesammelt in einem Datensatz gespeichert werden. Im Gegenteil, es muss einen Punktdatensatz für Bushaltestellen, ein Liniendatensatz für die Straße und ein Polygondatensatz für Siedlungen (siehe Abb. 33) erstellt werden. (vgl. [LIE-12: 31]) Dabei darf eine Shapefile nie die Größe von 2 GB überschreiten. (vgl. [GEO-18a]) Diese Problematik lässt sich in anderen Vektorformaten sehr wohl umsetzen

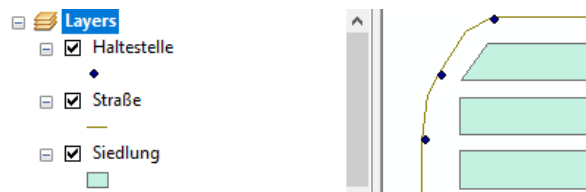


Abbildung 33: Geometrietypen eines Shapefiles (eigene Darstellung)

Wirft man einen Blick auf den Quellcode eines Shapefiles (siehe Abb. 34), so ist dieser verschlüsselt und kann von Texteditoren nicht geöffnet bzw. gelesen und adaptiert werden. (Diese Problematik gilt auch für die *.shx Datei)

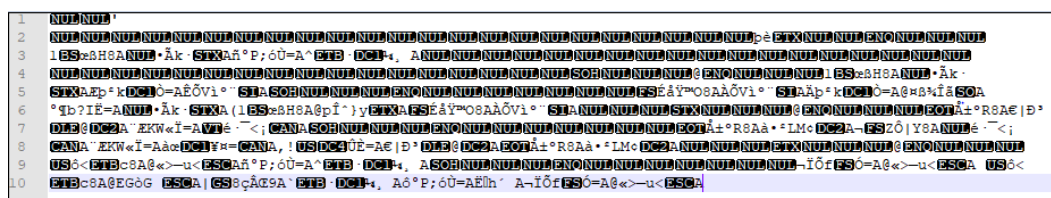


Abbildung 34: Shapefile im Texteditor (eigene Darstellung)

Wenn man hingegen mit QGIS arbeitet und in jener Software eine Layerdefinitionsdatei abspeichert, lässt sich der Dateicode öffnen und gegebenenfalls auch adaptieren. Der Code der erstellten Stildefinition selbst basiert auf der XML-Struktur. (siehe Abb. 37)

```

58 <symbols>
59 <symbol alpha="1" clip_to_extent="1" type="fill" name="0">
60 <layer pass="0" class="SimpleFill" locked="0">
61 <prop k="border_width_map_unit_scale" v="0,0,0,0,0,0"/>
62 <prop k="color" v="4,158,40,255"/>
63 <prop k="joinstyle" v="bevel"/>
64 <prop k="offset" v="0,0"/>
65 <prop k="offset_map_unit_scale" v="0,0,0,0,0,0"/>
66 <prop k="offset_unit" v="MM"/>
67 <prop k="outline_color" v="0,0,0,255"/>
68 <prop k="outline_style" v="solid"/>
69 <prop k="outline_width" v="0.26"/>
70 <prop k="outline_width_unit" v="MM"/>
71 <prop k="style" v="solid"/>
72 </layer>
73 </symbol>
74 </symbols>
75 <rotation/>
76 <size scale_scalemethod="diameter"/>
77 </renderer-v2>
78 <labeling type="simple"/>
79 <customproperties/>
80 <blendMode>0</blendMode>
81 <featureBlendMode>0</featureBlendMode>
82 <layerTransparency>0</layerTransparency>
83 <displayfield>OBJECTID</displayfield>
84 <label>0</label>

```

Abbildung 37: Layerdefinitionsdatei QGIS (*.qml) (eigene Darstellung)

Animation

Shapefiles können in ArcGIS Pro über einen Zeitraum animiert werden. Hierfür wird eine Attributspalte mit Datum oder Zeit benötigt. Der Inhalt der Spalte kann zum Beispiel im UTC-Zeitformat (YYYY-MM-DD hh:mm:ss) vorliegen. (vgl. [HOW-17]) Abbildung 38 stellt Animationseinstellungen in ArcMap dar.

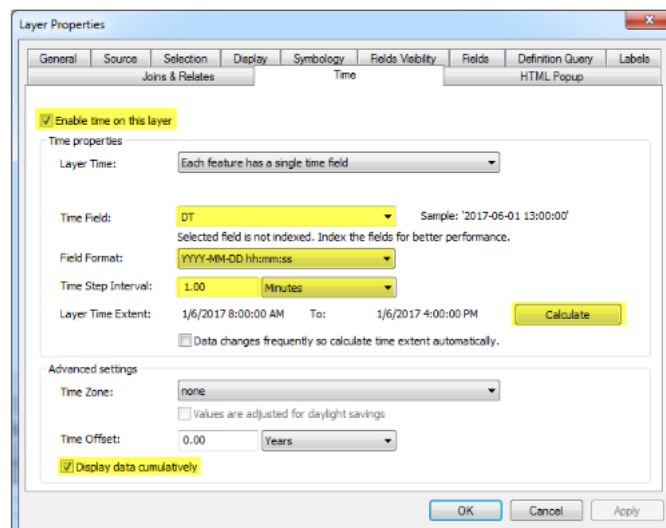


Abbildung 38: Animationseinstellungen in ArcGIS ([HOW-17])

In den *Layer Properties* → *Time* können das Datumsformat, die Animationszeit und weitere Angaben zur Animation des Shapefiles definiert werden. Die Animation selbst kann nur in

der Software ArcGIS Pro (oder ArcMap) definiert werden. Da das Shapefile verschlüsselt ist, lassen sich Animationen nicht im Quellcode definieren. (*Parallel dazu können Shapefiles auch in der Software Open-Source Software QGIS animiert werden.*)

Interaktivität

Shapefiles lassen sich in ArcGIS Pro interaktiv ansprechen. Durch das Werkzeug *Auswahl* → *nach Attributen auswählen* oder *Auswahl* → *Lagebezogen auswählen* können Features interaktiv ausgewählt und die gewählte Auswahl angezeigt werden. Zusätzlich können durch SQL-Abfragen interaktiv Objekte eines Shapefiles abgefragt werden. (vgl. [INT-16])

7.4 Offene Vektorformate

7.4.1 GML

„GML has three main roles with respect to geographic information. First as an encoding for the transport of geographic information from one system to another; second as a modeling language for describing geo-graphic information types; and third as storage format for geographic information. “ ([LAK-05: 1081])

Die GML-Auszeichnungssprache (Geography Markup Language) ist eine vom OGC entwickelte XML Anwendung, um geographische Features zu definieren. Dabei steht nicht nur die Geometrie im Fokus, sondern auch Objektattribute jener. Als OGC Formatstandard dient es zur Modellierung geographischer Systeme und als offenes, freies Format zum Austausch räumlicher Information, vor allem im Internet. Wie bei XML-Dokumenten üblich, ist eine GML-Datei in zwei Blöcke geteilt, einerseits die Dokumentbeschreibung – Schema – und zweitens die tatsächlichen, räumlichen Daten. (vgl. [GEO-18])

Die in GML angewandten XML-Kodierungen basieren auf Klassen der folgenden internationalen Standards und OpenGIS Abstract Specifications und sind außerdem konform mit ISO 19118 (Geographic information -- Encoding): (vgl. [OGC-12: 6])

- **“ ISO 19103** — *Geographic information – Conceptual schema language*
- **ISO 19107** — *Geographic information – Spatial schema*
- **ISO 19108** — *Geographic information – Spatial schema*
- **ISO 19109** — *Geographic information – Rules for application schemas*
- **ISO 19111** — *Geographic information – Spatial referencing by coordinates*

- **ISO 19123** — *Geographic information – Schema for coverage geometry and functions* “[OGC-12: 6)]

Ziel ist, im Rahmen dieser internationalen Standards eine standardisierte Kodierung von geographischen Informationskonzepten festzulegen, um den Datenaustausch zwischen Plattformen zu optimieren.

Wie bereits erwähnt, werden Features in GML nicht nur durch ihre Geometrie, sondern auch durch andere Attribute beschrieben. Abbildung 39 zeigt hierzu eine beispielhafte Objektstruktur.

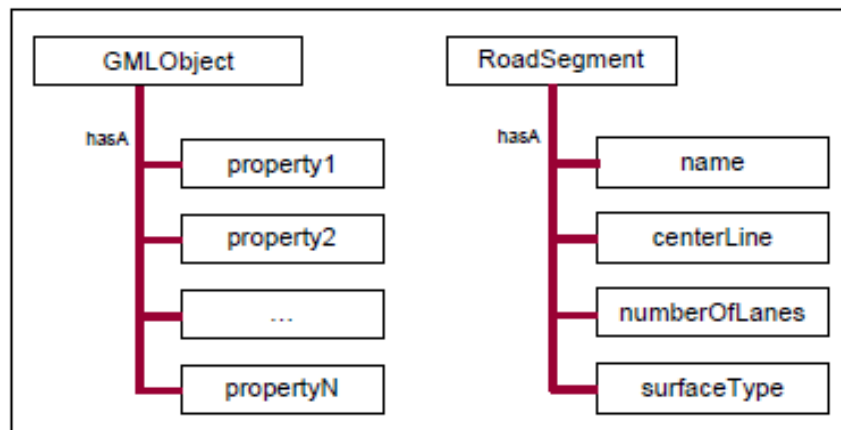


Abbildung 39: GML Objektstruktur ([BUR-06: 180])

Basierend auf der Objektstruktur ist ersichtlich, dass die Geometrie auf derselben Ebene wie andere Attribute geführt wird. Nach Burggraf David S. lassen sich durch jene Attribute verschiedene Beziehungen zwischen GML Objekten festlegen, wobei jedem GML Objekt Werte durch Objektattribute zugewiesen sind. (vgl. [BUR-06: 180])

Dieses Schema wird nun in Abbildung 40 beispielhaft für das Objekt RoadSegment umgesetzt.

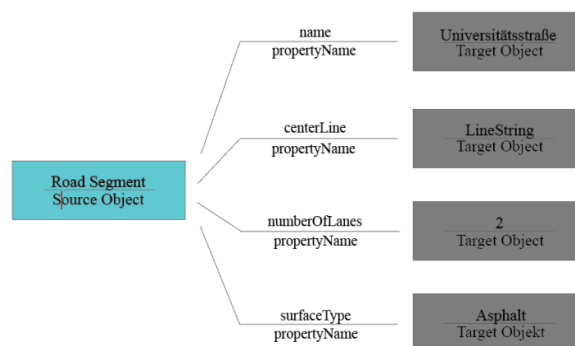


Abbildung 40: GML Objekt-Attributstruktur ([BUR-06: 180])

Nachdem der grundlegende Objektaufbau in GML diskutiert wurde, folgt nun eine kurze Analyse der Dateistruktur. Hier ist festzuhalten, dass ein GML Objekt kein anderes Objekt im Rahmen seiner Attribute enthalten kann. Dieses Objekt muss gesondert in einem Attribut von Objekt 1 definiert werden.

Im XML-Code wird in GML jedes Objekt eindeutig mit einer einmaligen „gml:id“ ausgewiesen. Attribute von GML-Objekten können im Codeblock oder außerhalb definiert werden. Im Codeblock selbst wird der propertyName (vgl. Abb. 41) durch ein „xlink:href“ Attribut ersetzt. Dieser bezieht sich folgend auf ein im Dokument vorkommendes GML Objekt oder Attribut, welches durch eine „id“ in der Hauptdatei oder einem anderen, GML-Dokument, in der gleichen Pfadebene, definiert ist. Zusätzlich können sich die <xlink:href> Attribute auch auf GML-Objekte im Internet beziehen.

“In general, remote property values can reference a GML object located anywhere on the Internet.”([BUR-06:181])

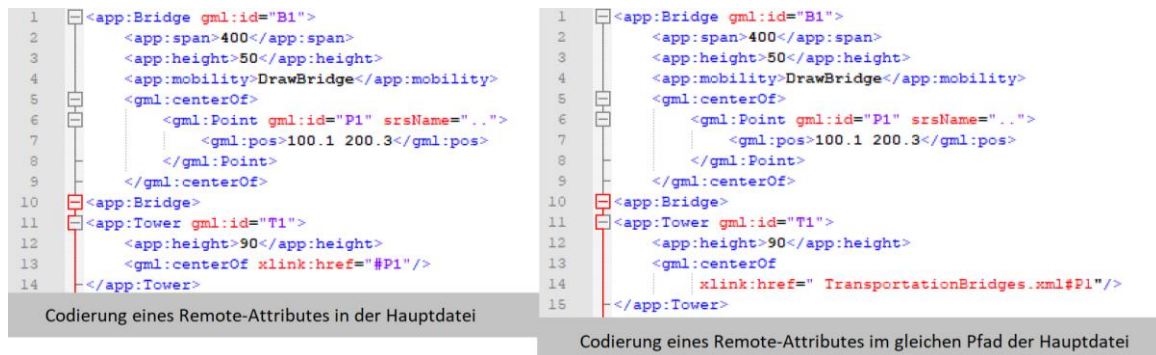


Abbildung 41: Codebeispiel von „Remote“ GML-Objekten ([LAK-04: 85])

Geometrie

Geometrietypen, welche vom Format GML unterstützt werden, sind die geometrischen Primitiva Point/MultiPoint, Line/MultiLine und Polygon/MultiPolygon. Wie in Kapitel 7.4.1 bereits erwähnt, wird die Geometrie eines GML-Objektes auf derselben Ebene wie Attribute geführt. (vgl. [GEO-18])

Raumbezug

GML-Objekte besitzen einen geographischen Raumbezug. Hierfür wird das Koordinatensystem in den geometryProperties für jedes Objekt mitgespeichert. Durch die Geometriekoordinaten im jeweilig definierten Koordinatensystem lassen sich die Objekte einer GML-Datei lagerichtig verorten. (siehe Abb. 42)

```

14 <gml:featureMember>
15   <ogr:test fid="test.0">
16     <ogr:geometryProperty><gml:MultiPolygon srsName="EPSG:4030"><gml:polygonMember><gml:Polygon><gml:outerBounds
17       <ogr:ID>4470</ogr:ID>
18       <ogr:GlideNumbe xsi:nil="true"/>
19       <ogr:Country>Canada</ogr:Country>
20       <ogr:OtherCount xsi:nil="true"/>
21       <ogr:long>-74.405782000000002</ogr:long>
22       <ogr:lat>45.574773999999998</ogr:lat>
23       <ogr:MainCause>Snowmelt and Heavy Rain</ogr:MainCause>
24       <ogr:Severity>1.5000000000000000</ogr:Severity>
25     </ogr:test>
26   </gml:featureMember>

```

Abbildung 42: Definition des Koordinatensystems eines GML-Objekts (eigene Darstellung)

Graphisches Erscheinungsbild

Das GML-Format wird primär zur Modellierung von geographischen Objekten und deren Transport im Web verwendet. Aspekte der graphischen Visualisierung von Geoobjekten werden nicht berücksichtigt. (vgl. [LAK-05: 1081])

Animation

Zum Zeitpunkt des Verfassens dieser Masterarbeit liegt keinerlei Dokumentation über den Einsatz einer Animationsfunktion im GML-Format vor.

Interaktivität

Objekte einer GML-Datei können interaktiv über die eindeutige „fid“ (= ID) jedes Objektes oder anhand diverser Objektattribute angesprochen werden.

7.4.2 KML/KMZ

KML ist eine von Google entwickelte Auszeichnungssprache, welche zur zwei- und dreidimensionalen Visualisierung von räumlicher Information entwickelt wurde. Beim genannten KMZ-Format (*.kmz) handelt es sich die gezippte Version des KML Formates. Im Jahr 2008 wurde KML vom Open Geospatial Consortium als OGC Standard übernommen und wird auch als Google Earths Beschreibungssprache geführt. Das Ziel von KML liegt in der Visualisierung der Geodaten und der User-Navigation. Wie auch GML basiert KML auf dem XML Standard. Die Dateistruktur basiert auf Tags und besteht aus Elementen und Attributen für spezifische Displaywiedergaben. (vgl. [OGC-07: iii - xi])

Vor allem die folgenden zwei Ziele hinter der Übernahme von KML als OGC Standard sind im Kontext der Thematik dieser Masterarbeit wichtig:

- Internationaler Standard, um geographische Visualisierungen und Annotationen auf 2D Online- und Mobilkarten und 3D Erdbrowsern wiederzugeben
- Möglichkeit der Interoperabilität zwischen verschiedenen Implementierungen von Erdbrowsern (vgl. [OGC-08])

“The KML community is wide and varied, ... Casual users create KML Placemarks to identify their homes, describe journeys, and plan cross-country hikes and cycling ventures. Scientists use KML to provide detailed mappings of resources, models, and trends such as volcanic eruptions, weather patterns, earthquake activity, and mineral deposits. Real estate professionals, architects, and city development agencies use KML to propose construction and visualize plans. Students and teachers use KML to explore people, places, and events, both historic and current. Organizations such as National Geographic, UNESCO, and the Smithsonian have all used KML to display their rich sets of global data.” [KML-17]

Im Gegensatz zu GML ist der Fokus bei KML darauf gerichtet, geographische Information für eine Darstellung in 2D Webbrowsern oder virtuellen 3D Globen ansprechend zu repräsentieren und visualisieren. Dabei basiert der Aufbau eines KML-Dokumentes jenem auf der XML-Dateistruktur. (vgl. [KML-19])

Das KML Format wird dabei unter anderem für die folgenden Aufgaben herangezogen:

- Beschriftung von Orten
- Beschriftungen und Symbole zur Orientierung und Identifizierung von Objekten an der Planetenoberfläche
- Verschiedene Ansichten -> Kameraposition
- Bilder und Hyperlinks, eingebettet in der HTML Beschreibung des KML Objektes
- Stildefinition von (Geo)Features
- Feature-Hierarchie (vgl. [OGC-15: 1])

In den nun genannten Aufgaben des KML Formates wird der Unterschied zwischen GML und KML deutlich. Während es bei GML um die Modellierung und Beschreibung von Geometrien und ihren Attributen geht, handelt es sich bei KML um ein Format, bei dem die Visualisierung und Repräsentation von (Geo)Features im Vordergrund steht.

KML basiert wie auch GML auf XML, somit ist es mit der Dateistruktur von Abbildung 30 zu vergleichen. Als Beispiel wird wieder ein „RoadSegment“ herangezogen. Hierfür wurde in QGIS ein imaginäres Straßenstück vektorisiert und als KML-Datei exportiert. (siehe Abb. 43)

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <kml xmlns="http://www.opengis.net/kml/2.2">
3  <Document id="root_doc">
4  <Schema name="RoadSegment" id="RoadSegment">
5    <SimpleField name="id" type="float"/></SimpleField>
6  </Schema>
7  <Folder><name>RoadSegment</name>
8  <Placemark>
9    <Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>0</fill></PolyStyle></Style>
10   <MultiGeometry><LineString><coordinates>
11     -139.765920467186,
12     -52.401883612923 -120.214196329255,
13     -23.5734612046021 -130.804713570634,
14     21.6340175048673 -150.356437708565,
15     -2.27186361135497 -146.283161846496,
16     -44.885424616852</coordinates></LineString></MultiGeometry>
17   </Placemark>
18 </Folder>
19 </Document></kml>

```

Abbildung 43: KML Code "RoadSegment" (eigene Darstellung)

Abgesehen von der XML-Dateistruktur fällt in Abbildung 43 auf, dass das Feature „RoadSegment“ in einem „Placemark“ definiert ist. Ein Placemark Tag beinhaltet ein Feature und seine Geometrie. Es wird unter dem Tag Placemark gespeichert, da es sich bei KML um ein für Google Earth konzipiertes Format handelt und jeder Placemark unter „Meinen“ Orten als Listenelement angeführt wird. (vgl. [KML-o.J.]) Jedoch bedeutet dies auch, dass jedes Attribut eines Objektes als separater Placemark gespeichert wird. Dies impliziert, dass je mehr Objekt-Tags enthalten sind auch die Dateigröße schnell ansteigt.

Geometrie

Das KML-Format unterstützt die Geometrietypen Point/MultiPoint, Line/MultiLine, Polygon/MultiPolygon und MultiPolygon. (vgl. [KML-o.J.a])

Raumbezug

Ein definiertes Koordinatensystem findet man im Code einer KML-Datei nicht, da KML-Dateien (und Google Earth) als Standard-Spezifikation immer mit dem WGS84 (EPSG: 4326) arbeiten und Koordinaten in diesem System abgebildet werden. (vgl. [KML-17])

Graphische Erscheinungsform

Es stehen mehrere Optionen zur Wahl, um KML-Objekte graphisch aufzubereiten.

- Intern **pro Geoobjekt**

In einer KML-Datei wird die Stilinformation jedes Geoobjektes mitgespeichert. Die Farbinformation wird in einem 8-Digit Hexcode angegeben. In dieser ist der Style eines jeden Features definiert. (vgl. [KML-o.J.])

```
<Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>e9e9e9</fill></PolyStyle></Style>
```

- Shared styles - Intern über eine **<styleId>**

Gibt es in einer KML-Datei mehrere Objekte mit demselben Style, so kann dieser bereits am Anfang der Datei durch eine Style-ID definiert werden. Diese Methode wird auch *shared styles* genannt und wird für den Testdatensatz umgesetzt. Durch die ID-Definition am Anfang der Datei greift jedes Objekt über eine **<styleUrl>** auf die Stildefinition zu. (vgl. [KML-o.J.])

- Extern über **Shared styles**

Die graphische Aufbereitung von KML-Objekten lässt sich auch extern definieren. Hierfür muss auf der gleichen Ordner Ebene der KML-Hauptdatei eine Referenzdatei liegen. Im Tag der **<styleUrl>** greift die KML-Datei auf die extern definierte Style-ID zu. (vgl. [KML-o.J.])



Abbildung 44: Externe graphische Stilisierung von KML-Objekten (eigene Darstellung)

- **FadeIn/FadeOut**

Auch kann in KML ein FadeIn und FadeOut von Objekten definiert werden. Seit der Version 5.0 werden Erweiterungen zur Verfügung gestellt, um neue Features in Google Earth zu integrieren. Die Erweiterungen verwenden „gx“ als Prefix. Das Ein- und Ausblenden von Objekten kann im Rahmen einer Google Earth Tour (**<gx: Tour>**) durch eine angegebene Zeit **<gx: duration>** festgelegt werden. (vgl. [KML-o.J.])

```
<gx:AnimatedUpdate id="ID">
  <gx:duration>0.0</gx:duration>    <!-- double, specifies time in seconds -->
  <Update>
    <targetHref>...</targetHref>    <!-- required; can contain a URL or be left blank -->
    <!-- (to target elements within the same file) -->
    <Change>...</Change>
    <Create>...</Create>
    <Delete>...</Delete>
  </Update>
  <gx:delayedStart>0</gx:delayedStart> <!-- double, specifies time in seconds -->
</gx:AnimatedUpdate>
```

Abbildung 45: Syntax der KML-Prefix **<gx:AnimatedUpdate>** ([KML-o.J.])

Animation

“When Google Earth opens a KML file that contains a Feature with a TimePrimitive element, it displays a time slider. (Google Earth automatically selects the beginning and ending units for the time slider based on the earliest and latest times found in the KML

Features in a particular file.) Using the slider and play button, the user can "play" the entire sequence or can select individual time periods for display.” ([TIM-18])

KML-Objekte lassen sich durch die Angabe einer Zeitinformation animieren. Diese Zeitinformation beeinflusst wann und wie lange die jeweiligen Objekte sichtbar sind. Dies wird von der Zeitleistenfunktion in Google Earth kontrolliert. KML-Animationen arbeiten mit zwei Zeit-Elementen: (siehe Abb. 47)

- **TimeStamp**: spezifizierter sichtbarer Moment pro Objekt
- **TimeSpan**: pro Objekt werden ein Start- (<begin>) und Endzeitpunkt (<end>) der Sichtbarkeit definiert (vgl. [TIM-18])

Je nach Fragestellung können diese Zeit-Elemente verwendet werden, um Geoobjekte in Google Earth zu animieren.

```
<TimeSpan id="ID">
  <begin>...</begin>      <!-- kml:dateTime -->
  <end>...</end>          <!-- kml:dateTime -->
</TimeSpan>

<TimeStamp id="ID">
  <when>...</when>      <!-- kml:dateTime -->
</TimeStamp>
```

Abbildung 46: TimeSpan und TimeStamp Syntax in KML ([KML-o.J.])

Interaktivität

Durch die eindeutige Placemark-Definition von KML-Objekten, können diese interaktiv anhand ihrer Placemark ID angesprochen werden. Dies gilt auch für im KML-Dokument definierte Folder, welche Gruppen von Geoobjekten enthalten. Zusätzlich lassen sich Geoobjekte einer KML-Datei auch basierend auf identen Attributen gruppieren bzw. kategorisieren.

7.4.3 GeoJSON

„GeoJSON is a format for encoding a variety of geographic data structures. GeoJSON supports the following geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon. Geometric objects with additional properties are Feature objects. Sets of features are contained by FeatureCollection objects.“ ([GEO-o.J.]

Das Format GeoJSON (*.geojson) basiert auf JSON (siehe Kapitel 7.1.2). Diese Erweiterung des Formates arbeitet mit der Codierung von Geometrien und verwendet genauso das Schlüssel-Wertepaar Prinzip.

```

1  /* Das ist ein GeoJSON Beispiel */
2  {
3    "type": "FeatureCollection",
4    "features": [
5      {
6        "type": "Feature",
7        "geometry": {
8          "type": "LineString",
9          "coordinates": [
10           [
11             7.11227700501511,
12             50.724511372039
13           ],
14           [
15             7.11320378477021,
16             50.7232906914878
17           ]
18         ]
19       },
20       "properties": {
21         "geoTyp": "Linie"
22       }
23     ]
24   },
25   {
26     "type": "Feature",
27     "geometry": {
28       "type": "Polygon",
29       "coordinates": [
30         [
31           [
32             -0.289431323256268,
33             0.00451046354601
34           ],
35           [
36             -0.272488103953148,
37             0.004461017569546
38           ],
39           [
40             -0.271961932650073,
41             -0.023700585651537
42           ]
43         ]
44       ]
45     }
46   }
47 ]

```

Abbildung 47: GeoJSON Datenstruktur (eigene Darstellung)

Der Unterschied zu Geokodierungsbeispielen der bisher genannten Vektorformate ist die besondere Dateistruktur und der dadurch gering gehaltene Speicherbedarf. (vgl. [CRI-14: 41-42]) Abbildung 47 bildet ein Codebeispiel einer Geocodierung in *.geojson ab.

Geometrie

Das GeoJSON Format kann nicht nur die Grundprimitiva Punkt, Linie und Fläche, sondern auch Multipunkt, -Linie und -Fläche abbilden. (vgl. [GEO-o.J.])

Raumbezug

In einem GeoJSON-Dokument lässt sich der geographische Raumbezug der Datei definieren. Im Schlüssel „CRS“: wird der EPSG Code des gewählten Koordinatensystems definiert.

```
"crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::4030" } }
```

Graphisches Erscheinungsbild

“simplestyle is a set of agreed-upon 'special values' in the pre-existing GeoJSON data standard that define styles. As such, files implementing simplestyle are by definition valid GeoJSON files and valid JSON files.. The entirety of the simplestyle spec is optional, so any combination of specified properties are valid. When properties are not given, the defaults below are assumed by implementations.” ([SIM-17])

Da es sich bei GeoJSON um ein Format zum Austausch von Geodaten handelt, ist eine graphische Stilisierung von Geoobjekten nicht standardmäßig im Format implementiert. Die

simplestyle Spezifikation definiert spezielle Werte GeoJSON-Standard. Der Stil der Geoobjekte wird in den Properties einer Datei definiert. (vgl. [SIM-17])

```

1  {
2    "type": "FeatureCollection",
3    "features": [{ "type": "Feature",
4      "geometry": {
5        "type": "Point",
6        "coordinates": [0, 0]
7      },
8      "properties": {
9        "title": "A title",
10       "description": "A description",
11       "marker-size": "medium",
12       "marker-symbol": "bus",
13       "marker-color": "#fff",
14       "stroke": "#555555",
15       "stroke-opacity": 1.0,
16       "stroke-width": 2,
17       "fill": "#555555",
18       "fill-opacity": 0.5
19     }
20   }]
21 }
```

Abbildung 48: GeoJSON Stildefinition nach der *simplestyle* Spezifikation ([SIM-17])

GeoJSON Objekte können auch kategorisiert graphisch aufbereitet werden. Hierfür müssen die Geoobjekte einer Kategorie in einer FeatureCollection liegen. Wird in dieser FeatureCollection eine Klasse gebildet, so können jene Geoobjekte über eine externe CSS-Stilisierung graphisch aufbereitet werden (siehe Abb. 49 und 50). (vgl. [GEO-14])

```

1  {
2    "type": "FeatureCollection",
3    "className": {
4      "baseVal": "Linie"
5    },
6    "features": [
7      { "type": "Feature",
8        "geometry": {
9          "type": "Line",
10         "coordinates": [0, 1]
11       },
12       "properties": {
13         "title": "A title",
14         "description": "A description",
15       },
16     },
17     { "type": "Feature",
18       "geometry": {
19         "type": "Line",
20         "coordinates": [1, 0]
21       },
22       "properties": {
23         "title": "Another title",
24         "description": "Another description",
25       }
26     }
27   ]
28 }
```

Abbildung 49: GeoJSON kategorisierte graphische Aufbereitung (eigene Darstellung)

```

1  <style type="text/css">
2
3  .Linie{
4    stroke:#DDDDDD;
5    stroke-width: 1;
6    fill:#DDDDDD;
7    opacity: 0.6;
8  }
```

Abbildung 50: CSS Datei zur Stilisierung der GeoJSON Objekte (eigene Darstellung)

Eine weitere Möglichkeit zur graphischen Gestaltung von GeoJSON Objekten bietet Mapbox.² (vgl. [GEO-o.J.a])

Animation

Zum momentanen Stand gibt es keinerlei Aufzeichnung oder Literatur bezüglich des Bestehens einer Animationsfunktion in GeoJSON.

Interaktivität

Geoobjekte in GeoJSON können unter anderem interaktiv anhand ihrer ObjectID angesprochen werden. Dies gilt auch für Attribute der jeweiligen Geoobjekte.

7.4.4 SVG

„SVG (Abkürzung für "Scalable Vector Graphics") ist ein freies, quelloffenes, standardisiertes Dateiformat für Vektorgrafiken. Es wird vom [W3C](#) (World Wide Web Consortium), der wichtigsten internationalen Standardisierungsinstanz für das Internet, entwickelt und gepflegt.“ ([DAS-o.J.])

Das SVG-Format ist ein Standard des World-Wide-Web Consortium. Das Format kann unter anderem Punkte, Linien, Polygone, Rechtecke, Ringe, Ellipsen, Bilder und Text visualisieren. Zusätzlich können in einer SVG-Datei Animation und Interaktion von Vektorobjekten umgesetzt werden. (vgl. [SCA-18])

Die Dateistruktur einer SVG-Datei basiert, wie auch schon bei anderen Vektorformaten, auf XML. Weitergehend kann das Format direkt in einer HTML-Webseite integriert werden. Dies lässt sich davon ableiten, dass das SVG Format die W3C Spezifikation für die Darstellung von Graphiken im Internet ist. (vgl. [HTM-18]) Der wesentliche Unterschied zwischen dem SVG-Format und anderen, im Rahmen dieser Arbeit genannten Vektorformaten, liegt am Dateityp selbst. Denn es geht nicht spezifisch um die Repräsentation von Geodaten, sondern um die Visualisierung von jeglichen Vektorgrafiken für eine Darstellung im Internet.

² Mapbox ist eine Open Source Plattform für benutzerdefinierte Kartendienste. Diese bauen Großteils auf OpenStreet Map auf. Kartenprodukte werden als Webmaps und mobile Applikationen zur Verfügung gestellt. (vgl. [MAP-18]) Die Anwendungen der Plattform basieren auf Mapbox GL JS. Dabei handelt es sich um eine JavaScript-Bibliothek, die WebGL zum Rendern interaktiver Karten aus Vektorkacheln und Map-Stilen verwendet. Es ist Teil des Mapbox GL-Ökosystems, zu dem Mapbox Mobile gehört. Dies ist ein kompatibler Renderer, der in C++ mit Bindungen für Desktop- und mobile Plattformen geschrieben wurde. ([MAP-18b])

Geometrie

Das SVG-Format kann Objekte vom Typ Kreis, Line/PolyLine, Polygon, Rechteck und Ellipse abbilden. (siehe Kapitel 7.4.4, S. 64)

Raumbezug

Geoobjekte in einer SVG-Datei werden nicht anhand ihrer geographischen Koordinaten abgebildet, sondern auf Basis eines SVG internen Koordinatensystems. Um Objekte exakt zu platzieren, liegt der Nullpunkt in der oberen, linken Ecke des Dokuments. (vgl. [SCA-18])

Graphisches Erscheinungsbild

Das folgende Codebeispiel (Abb. 51) bildet ein *.svg Beispiel in Form einer Ellipse ab. Die Stildefinition des Vektorobjekts ist in diesem Beispiel direkt in das SVG-Dokument eingebettet. Es kann jedoch genauso separat ein eigenes CSS-Dokument (Cascadian Style Sheet) erstellt werden. Dies erleichtert in einem langen Codeblock den Überblick. (Der Dateicode in Abbildung 51 baut auf der bekannten XML-Struktur auf. Zusätzlich ist eine weitere Option, dieses Dokument direkt in eine HTML-Webseite einzubauen. Dazu würden zusätzlich die HTML-Datei und ein Körper (<html><body> </body></html>) im Dokument definiert werden.)

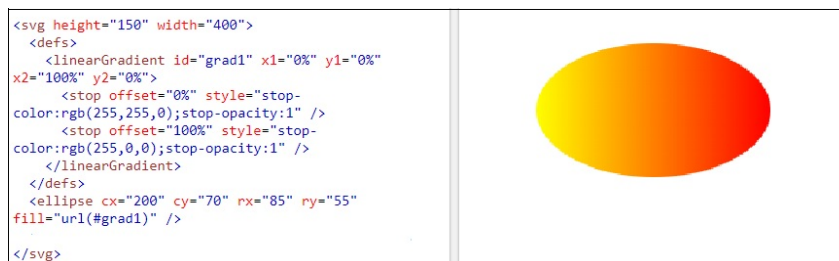


Abbildung 51: SVG Code Ellipse ([SVG-18])

Animation

Objekte einer SVG-Datei lassen durch Animationselemente animieren. Diese Elemente wurden in der SMIL (Synchronized Multimedia Integration Language) Animation Spezifikation definiert. Folgende Tags finden Einsatz in SVG-Animationen: (vgl. [AGU-14])

- **<animate>** - Animation von skalaren Attributen und Eigenschaften über einen bestimmten Zeitraum
- **<set>** - Animation der Sichtbarkeitseigenschaft von SVG-Objekten

Unter der WKBGeometry wird die binäre Darstellungsform des WKT Formates verstanden. In jenem Format wird das geometrische Objekt, basierend auf numerischen Typen (Unsigned Integer, Double), serialisiert. (vgl. [OGC-11: 62])

Geometrie

Das WKT-Format wurde in den Simple Feature Geometrien spezifiziert. Da diese bereits genauer in Kapitel 6.1 betrachtet wurden, ist an diesem Punkt bereits die Frage geklärt, welche Geometrien durch das WKT-Format abgebildet werden können. Doch wie sieht die Dateistruktur hinter diesem Format aus? Dies wird nun in Tabelle 16 für verschiedene Geometrietypen dargestellt. (vgl. [OGC-11: 61])

Geometrie	WKT Code
Point (2D)	POINT (10 10)
Point (3D)	POINT (10 10 5)
MultiPoint	POINT (10 10,20 20)
LineString	LINSTRING (10 10,20 20,30 40)
MultiLineString	MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))
Polygon	POLYGON (10 10,10 20,20 20,20 15,10 10)
Polygon (Loch innen)	POLYGON ((0.5 0.5,0 5,5 5,0.5 0.5),(10 10,10 20,20 20,20 15,10 10)
MultiPolygon	MULTIPOLYGON (((10 10, 10 20, 20 20, 20 15, 10 10)),((60 60, 70 70, 80 60, 60 60)))
GeometryCollection	GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINSTRING (15 15,20 20))
TIN	TIN (((0 0 0, 0 0 1, 0 1 0, 0 0 0)),((0 0 0, 0 1 0, 1 0 0, 0 0 0)),((0 0 0, 1 0 0, 0 0 1, 0 0 0)),((1 0 0, 0 1 0, 0 0 1, 1 0 0)))

Tabelle 16: WKT Struktur verschiedener Features ([OGC-11: 61])

Tabelle 16 zeigt, dass das WKT Format, basierend auf seiner Grundstruktur, einfach aufgebaut ist. Jedes Feature wird durch seinen FeatureType plus der Geometrie definiert. Die Koordinaten selbst werden durch Leerzeichen getrennt. Dabei ist die Auszeichnungssprache im Gegensatz zu bisher genannten Formaten nicht CaseSensitiv. Wichtig ist, dass die runde Klammer “()” im WKT Code immer eine Sequenz definiert. Somit wird der Inhalt einer jeden Klammer als ein Geometriepunkt aufgenommen. Insgesamt kann das Format Features vom 0- bis in den 3-dimensionalen Bereich abbilden. (vgl. [OGC-11: 62]) Letztere Dimension (3D) spielt jedoch im Kontext der Forschungsarbeit dieser Masterarbeit keine Rolle. Verglichen mit dem KML-Format in Kapitel 7.4.2 ist es im WKT Format möglich, eine Geometriesammlung zu definieren. Dies ist ein positiver Aspekt in Hinblick auf die Dateigröße.

Raumbezug

Die Koordinaten eines WKT Codes werden vom jeweiligen Koordinatensystem des Geoobjekts übernommen. Jedoch ist das Koordinatensystem nicht im WKT Code selbst definiert.

Graphisches Erscheinungsbild

Da es sich bei WKT nur um eine Geometriedefinition handelt, ist eine individuelle Visualisierung/Gestaltung der Features nicht vorgesehen.

Animation

Es gibt keine Dokumentation zur Animation von WKT Objekten.

Interaktivität

Da im WKT String immer nur der Geometrietyp und die Koordinaten des jeweiligen Geoobjektes abgebildet werden, kann mit WKT Objekten nicht interaktiv gearbeitet werden. Da es sich, verglichen mit den bisher analysierten Vektorformaten, um keine „Datei“ handelt, sind WKT Objekte nicht über eine ID oder Attribute interaktiv anzusprechen.

7.4.6 OmniSuite Punkt-Format

Im Zuge der Software OmniSuite bietet sich, abgesehen von der asynchronen Integration diverser genannter Vektorformate, die Möglichkeit der Implementierung von Punktdatensätzen. Dabei handelt es sich um eine interne und synchrone Methode (siehe Kapitel 4.3). Vorteile sind eine schnellere Inhaltserstellung und Möglichkeiten der Interaktivität. Ein Nachteil ist, dass es bei der Performance von großen Punktdatensätzen zu Verzögerungen kommen kann. (vgl. [GLO-18a: 79]) Bei den Daten selbst handelt es sich in diesem Sinn nicht um „Vektoren“, sondern um ein Textfile mit je einer Spalte für Längen- und Breitengrad pro Punkt.

Um das benötigte Textfile zu generieren, wird eine Tabelle in einem Kalkulationsprogramm, wie zum Beispiel Microsoft Excel oder Libre Office Calculator, aufbereitet. Breiten- (lat) und Längengrade (long) müssen in dieser Tabelle in den ersten beiden Spalten festgehalten sein (siehe Tabelle 17). Zusätzliche Attribute, wie unter anderem Namen oder Kategorien, können in den darauffolgenden Spalten definiert werden. (vgl. [GLO-18a: 81])

	A	B	C	D	E
1	lat	long	name	category	value
2	5.32	-4.03	Abidjan	2	4.60
3	31.20	29.92	Alexandria	2	4.60
4	39.93	32.86	Ankara	1	4.22
5	33.76	-84.39	Atlanta	2	4.71
6	-36.84	174.74	Auckland	2	1.31
7	33.33	44.42	Baghdad	1	6.37
8	-6.91	107.61	Bandung	2	4.76
9	13.75	100.47	Bangkok	1	14.54
10	39.91	116.39	Beijing	1	18.24

Tabelle 17: OmniSuite Punktdatensatz Auszug ([GLO-18a: 89])

Bevor Punkte in OmniSuite abgebildet werden können, muss die Tabelle noch im *.csv Format (Comma Separated Value) abgespeichert werden. Wichtig ist es, dass Trennzeichen zwischen Spalten nicht als Beistrich oder Strichpunkt, sondern als Leerzeichen gespeichert sind. Anschließend kann die Dateieindung *.csv in *.txt einer Textdatei geändert werden. Abbildung 53 bildet den Inhalt einer solchen *.txt Datei ab. (vgl. [GLO-18a: 82])

```
1 lat long value start
2 55.39 -134.65 7.50 0.12
3 -62.57 -161.43 6.10 0.40
4 4.97 95.86 6.10 0.57
5 42.61 79.71 6.10 0.76
6 -28.08 -70.62 6.80 0.82
7 -10.64 166.37 6.10 0.82
8 -10.63 166.38 6.10 0.83
9 -11.10 165.53 6.00 0.86
10 -10.90 165.38 6.30 0.87
11 -11.12 165.38 6.40 0.87
12 42.76 143.11 6.90 0.89
13 -10.94 165.26 6.00 0.90
```

Abbildung 53: Beispiel eines Textfiles eines Punktdatensatz ([GLO-18a: 93])



Abbildung 54: OmniSuite Punktformat am sphärischen Display ([GLO-18a: 88])

Geometrie

Das OmniSuite Punktformat kann, wie es der Name bereits verrät, nur Punktgeometrien am sphärischen Display visualisieren.

Raumbezug

Wie bereits in Kapitel 2.2.2 diskutiert, liegen die Koordinaten des Hyperglobus in einem geographischen Koordinatensystem vor. Somit liegen die Punktkoordinaten der Breitengrade zwischen -90° bis 90° und jene der Längengrade zwischen -180° bis 180° . Der Zentralmeridian befindet sich in Greenwich. (vgl. [GLO-18a: 80])

Graphisches Erscheinungsbild

Die kartographische Gestaltung von OmniSuite Punktoobjekten wird gesondert in einem CSS-Dokument definiert. Durch dieses können Objekte nach der Thematik und Fragestellung graphisch aufbereitet und modifiziert werden. (vgl. [GLO-18a: 83])

```

1  .category1 {
2    marker-symbol: ellipsoid;
3    marker-width: 25 * [value] * 2.1;
4    marker-height: 25 * [value] * 2.1;
5    marker-color: rgba(219, 241, 252, 0);
6    marker-outline-style: solid;
7    marker-outline-width: 4px;
8    marker-outline-color: rgba(219, 241, 252, 255);
9    text-name: [name];
10   text-dy: 5;
11   text-placement: S;
12 }
13
14 .category2 {
15   marker-symbol: ellipsoid;
16   marker-width: 25 * [value] * 2.1;
17   marker-height: 25 * [value] * 2.1;
18   marker-color: rgba(2, 37, 237, 0);
19   marker-outline-style: solid;
20   marker-outline-width: 4px;
21   marker-outline-color: rgba(2, 37, 237, 255);
22   text-name: [name];
23   text-dy: 5;
24   text-placement: S;
25 }

```

Abbildung 55: CSS-Datei eines Punktdatensatzes (eigene Darstellung)

Animation

Über eine implementierte Animationsfunktion können Punktoobjekte des Formates zeitlich animiert werden. Dies beinhaltet sowohl das objektbasierte Ein- als auch Ausblenden zu einem angegebenen Zeitpunkt in Sekunden. (vgl. [GLO-18a: 91])

Interaktivität

Die Visualisierung von Punktoobjekten am sphärischen Display ermöglicht die Interaktion und Animation von Punktdatensätzen. So können als Beispiel interaktiv Flughäfen oder Fluglinien aus weltweiten Flugdaten eines Landes ausgewählt werden.

7.5 Beurteilung der Vektorformate

Nachdem potentielle Vektorformate analysiert wurden, gilt es diese auf Basis der definierten Kriterien zu vergleichen und eine Auswahl für die praktische Evaluation zu treffen.

Grundlegend ist es zuerst wichtig, wie die verschiedenen Vektorformate vorliegen, beziehungsweise konvertiert werden können. Da es sich hierbei um Geodaten handelt, sollte ein Vektorformat für NutzerInnen in einem Geoinformationssystem geöffnet werden können. Da an diesem Punkt der vorliegenden Arbeit noch keine praktische Analyse

stattgefunden hat, wird die Popularität der jeweiligen Formate in den GIS-Programmen ArcGIS (Pro) und QGIS überprüft. Ausschlaggebend ist an diesem Punkt nicht nur ob ein Format geöffnet werden kann, sondern auch, ob man die Geodaten bei Bedarf in andere, genannte Vektorformate konvertieren kann. (siehe Tabelle 18) Auch dieses Ergebnis wird in die Formatauswahl miteinfließen.

	ArcGIS (Pro)						
	SHP	GML	KML	GeoJSON	SVG (*)	WKT (**)	OmniSuite Format
SHP	+	+	+	+	+	±	
GML	+	+	+	+	+	±	-
KML	+	+	+	+	+	±	-
GeoJSON	+	+	+	+	+	±	-
SVG	-	-	-	-	-	-	-
WKT	+	-	-	-	-	±	-
OmniSuite Format (***)	±	±	±	±	±	±	-

	QGIS						
	SHP	GML	KML	GeoJSON	SVG (*)	WKT (**)	OmniSuite Format
SHP	+	+	+	+	+	±	-
GML	+	+	+	+	+	±	-
KML	+	+	+	+	+	±	-
GeoJSON	+	+	+	+	+	±	-
SVG	-	-	-	-	-	-	-
WKT	+	-	-	-	-	±	-
OmniSuite Format (***)	±	±	±	±	±	±	-

* immer nur pro Objekt

** Gesamtes Kartenlayout

*** nur als *.csv Datei mit Beistrich anstatt dem Leerzeichen

Tabelle 18: Vektordatenkonvertierung in ArcGIS (Pro) und QGIS (eigene Darstellung)

Die Optionen der Vektordatenintegration wie auch Konvertierung sind in ArcGIS (Pro) und der Open-Source Plattform QGIS ident. Die Formate *.shp, *.gml, *.kml und *.geojson lassen sich in beiden Softwareprodukten öffnen und auch ins jeweilige andere Format konvertieren. (Die Formatauswahl ist hierbei größer als nur die in Kapitel 7 genannten Formate, jedoch wurden andere nicht im Rahmen dieser Arbeit untersucht.) Der Unterschied ist, dass im Gegensatz zu QGIS bei ArcGIS (Pro) zum Import immer eine Toolbox benötigt wird, wenn es sich nicht um das standartmäßige *.shp File handelt. Eine *.svg Datei kann in keinem der beiden Systeme importiert werden, da es sich nicht um räumliche verortete Daten handelt (siehe Kapitel 7.4.4). Jedoch ist es möglich, die aufbereiteten Vektorformate (oder auch ein gesamtes Kartenprojekt) im Layoutmodus im SVG-Format zu exportieren. In diesem Fall sollten die Daten bereits im benötigten Rasterformat (2ⁿ Pixel) exportiert werden. Auch das WKT-Format kann in beiden GIS-Programmen importiert werden (in QGIS durch das experimentelle Plugin MMQGIS). Beim Export hingegen kann immer nur ein WKT-String pro Objekt aus einem Datensatz erstellt

werden. Das OmniSuite Punktformat könnte, nur wenn die Leerzeichen in der *.txt Datei durch Beistriche ersetzt werden, als CSV-Format in das GIS geladen und in ein *.shp, *.gml, *.kml und *.geojson konvertiert werden.

Der erste zentrale Punkt der Formatanalyse waren die potentiellen Geometrieobjekte jedes Vektorformats. Bis auf das OmniSuite Punktformat kann jeder der analysierten Dateitypen die geometrischen Primitiva Punkt, Line und Polygon abbilden. Der Quellcode der Geometrieobjekte ist bis auf das Shapefile bei allen Vektorformaten unverschlüsselt. Da bereits ein Vektorformat für Punkt in der Globensoftware OmniSuite implementiert ist und der Fokus daher auf Linien- und Polygondaten gerichtet ist, wird dieses nicht weiter berücksichtigt.

Auch der Raumbezug spielt hinsichtlich einer korrekten Verortung der Geoobjekte am sphärischen Display eine wichtige Rolle. Bis auf das SVG-Format ist in jedem Vektorformat das Koordinatensystem angegeben. Auch wenn das SVG-Format nur mit einem internen, lokalen Koordinatensystem arbeitet, kann es Geodaten lagekorrekt abbilden. Hierfür müssen die Geoobjekte in einem Geoinformationssystem im Plattkartenformat mit einer Ausdehnung 2ⁿ Pixel exportiert werden.

Wie bereits in Kapitel 2.2.1 und 7.2 diskutiert wurde, ist die graphische Aufbereitung von Vektorobjekten essentiell für eine Visualisierung am sphärischen Display. Ist ein Vektorformat nur zur Definition und zum Austausch von Geodaten konzipiert, so ist es nicht für eine potentielle Anwendung am Hyperglobus geeignet. Aus diesem Grund werden die Formate GML und WKT nicht weiter untersucht.

Der Einsatz von Animationen am sphärischen Display spielt einen wesentlichen Faktor in der Erstellung von Global Stories. Um einen belebten Eindruck am Globus zu erzeugen, soll ein zukünftiges Format für das sphärische Display diese Anforderung erfüllen. Wenn diese Funktion aktuell noch nicht für ein Format entwickelt wurde, so soll die Option der Implementierung dieser Funktion zumindest nicht ausgeschlossen sein. Alle verbliebenen Formate, ausgenommen vom GeoJSON Format, besitzen eine Animationsfunktion, wenn auch mit unterschiedlichen Umsetzungen. Wie in Kapitel 7.4.3 erwähnt, sind die Formaterweiterungen des GeoJSON Formats anwendungsabhängig und wurden daher schon für verschiedene Einsatzbereiche implementiert. (→ *Graphisches Erscheinungsbild*) Aus diesem Grund wird die momentan noch nicht vorhandene Animationsfunktion des GeoJSON Formats nicht als Ausschlusskriterium für dieses gezählt. Für das Shapefile können Animationen (wie auch Interaktionen) nur intern in der GIS-Software umgesetzt werden. Somit wird auch das Shapefile nicht weiter in der praktischen Analyse berücksichtigt.

Um Geoobjekte für NutzerInnen am sphärischen Display interaktiv zu nutzen, ist die objektbasierte Interaktion mit Vektordaten ein wesentlicher Aspekt. Dadurch kann die transportierte Information geprägt und gefestigt werden. Bei allen Vektorformaten,

ausgenommen vom WKT-Format, lassen sich Geoobjekte, wenn auch auf unterschiedene Arten, interaktiv ansprechen.

Die gewählten Formate für die praktische Vektorformatanalyse in Kapitel 8 sind somit KML, SVG und GeoJSON. Es wird sich zeigen, welche Vor- und Nachteile diese drei Dateitypen für eine Visualisierung am sphärischen Display besitzen. Tabelle 19 bildet hierzu einen Überblick über die analysierten Vektorformattypen.

	Geometrie		geographischer Raumbezug		Graphisches Erscheinungsbild		Animation		Interaktion	
SHP	- Point/MultiPoint - Line - Polygon	±	Koordinatensystem in *.prj Datei definiert	+	*.lyr File - ArcGIS Pro (verschlüsselt) *.qlr File - QGIS	±	→ <i>nur softwareintern</i> ArcGIS Pro: Layer Properties → Time	±	→ <i>nur softwareintern</i> - nach Attributen - lagebezogen - SQL-Abfrage	±
GML	- Point/MultiPoint - Line/MultiLine - Polygon/MultiPolygon	±	- Koordinatensystem definiert <geometryProperties>	+	- keine graphische Visualisierung - primär zum Modellierung von Geoobjekten	-	nicht umgesetzt	-	- über eindeutige <i>fid</i> - nach Attributen	+
KML	- Point/MultiPoint - Line/MultiLine - Polygon/MultiPolygon	±	- intern in KML-Datei - extern in KML-Referenzdatei	+	- intern pro Geoobjekt - Shared styles (intern) - extern in KML-Referenzdatei - FadeIn/FadeOut von Objekten	++	- TimeSpan - TimeStamp	+	- über eindeutig definierten Placemark - nach Folder - nach Attributen	+
GeoJSON	- Point/MultiPoint - Line/MultiLine - Polygon/MultiPolygon	+	Koordinatensystem im Schlüssel CRS definiert	+	- nicht in GeoJSON Datei selbst → <i>Anwendungsabhängig</i> - CSS - Mapbox	+	nicht umgesetzt	-	- über eindeutige <i>FeatureID</i> - nach Attributen	
SVG	- Circle - Line/PolyLine - Polygon - Rectangle - Ellipse	+	SVG-internes Koordinatensystem	-	- in *.svg-Datei selbst - CSS	+	<animate> <set> <animateMotion> <animateColor>	+	→ <i>nur teilweise</i> - über eindeutige <i>path id</i> - pro Objekt eigene Layerebene	±
WKT	- Point/MultiPoint - Line/MultiLine - Polygon/MultiPolygon - GeometryCollection - TIN -> und viele mehr...	++	nicht im WKT-Code selbst definiert - geographische Koordinaten werden aus Ursprungsobjekt übernommen	±	nicht umgesetzt	-	nicht umgesetzt	-	nicht umgesetzt	-
OmniSuite Punktformat	Point	-	geographisches Koordinatensystem	+	CSS (extern)	+	Animation von Punktobjekten über Zeitangabe	+	nach Objekattributen	+

Tabelle 19: Vergleich der analysierten Vektorformate (eigene Darstellung)

8 Praktische Formatanalyse

Nachdem im letzten Kapitel eine theoretische Analyse potentieller Vektorformate stattgefunden hat, geht es nun um die praktische Analyse der gewählten Formate. Ziel dieser ist es, eine Empfehlung für einen Vektorformattyp auszugeben, welcher in der Zukunft am Hyperglobus visualisiert werden soll. Die Problematik, welche hinter der Forschung dieser praktischen Analyse steht ist die folgende:

Durch das OmniSuite Punktformat ist es Kristen J. bereits gelungen, eine direkte Vektorintegration am sphärischen Display zu implementieren. Durch dieses Format ist es möglich, Punktdaten objektbezogen anzusprechen, sie zu animieren auch interaktiv mit diesen zu arbeiten. Diese direkte, synchrone Integration ist für Linien- und Polygondatensätze jedoch noch nicht umgesetzt. Wenn es gilt diese am Hyperglobus zu visualisieren, muss auf den Ansatz der indirekten Rastervisualisierung zurückgegriffen werden (siehe Kapitel 4.2). Je nach Fragestellung und Datensatz, ist dies natürlich mit einem sehr hohen Arbeits- und Zeitaufwand verbunden. Zusätzlich ist die Option der Interaktivität nicht gegeben, da die ursprünglichen Vektordaten als Rasterbildfolge importiert wurden.

Es wird nun im ersten Schritt ein Datensatz vorgestellt, welcher die Kriterien für eine praktische Analyse erfüllt und es wird für ihn eine Global Story mit den momentan verfügbaren Methoden (Direkt-Raster/Indirekt-Raster und Direkt-Vektor) umgesetzt. Kriterien der Auswahl des Datensatzes waren, dass es sich beim Geometrietyp der Daten um Polygone oder Linien handelt. Zusätzlich wurde in einer Kundenfrage an die HRG nach einer Visualisierung der Testdaten im Zeitverlauf angefragt. Diese Kriterien passen ideal in das Forschungsgebiet dieser Masterarbeit. Da es sich jedoch um sehr umfangreichen Datensatz handelt, wird ein Ausschnitt von diesem verwendet. Basierend darauf findet im zweiten Teil dieses Kapitels eine praktische Analyse der Vektorformate, in Bezug auf die mögliche, zukünftige Implementierung in der Globensoftware, statt.

8.1 Umsetzung einer Global Story nach dem aktuellen Ansatz

8.1.1 Der Datensatz

Beim Datensatz, welcher für die praktische Analyse verwendet wird, handelt es sich um Daten des Darmouth Flood Observatory in Colorado. Die Daten selbst repräsentieren globale Überschwemmungsereignisse, kategorisiert nach ihrem Auslöser, ihrer Stärke, der Anzahl an zwangsläufig umgesiedelten Bewohnern oder nach Toten. Der Datensatz selbst wird seit dem Jahr 1985 geführt und wird mit jedem neuen Überflutungsereignis upgedatet. Zusätzlich wird pro Ereignispunkt eine Information zum zugehörigen Staat und dem Start- und Enddatum des Ereignisses geführt. Der Datensatz wurde gewählt, da er die Anforderungen der Elemente Thematik und Globenwürdigkeit aus Kapitel 2.2 erfüllt. Zusätzlich handelt es sich bei den Daten um Vektorpolygone, welche ein zentrales Element dieser Masterarbeit sind.

Die Formate, in welchem die Daten zur Verfügung stehen, sind einerseits in einer Excel Tabelle (*.xlsx) und Polygone im Shapefile-Format.

Ziel ist es, die Daten im Zeitverlauf darzustellen. Auch können die Kategorien in Form einer Punktsignatur zusätzlich visualisiert werden. Abbildung 56 zeigt den *.shp Ausgangsdatensatz. Ein Ausschnitt des zugehörigen Excel-Files wurde bereits in Kapitel 7.4.6 abgebildet.

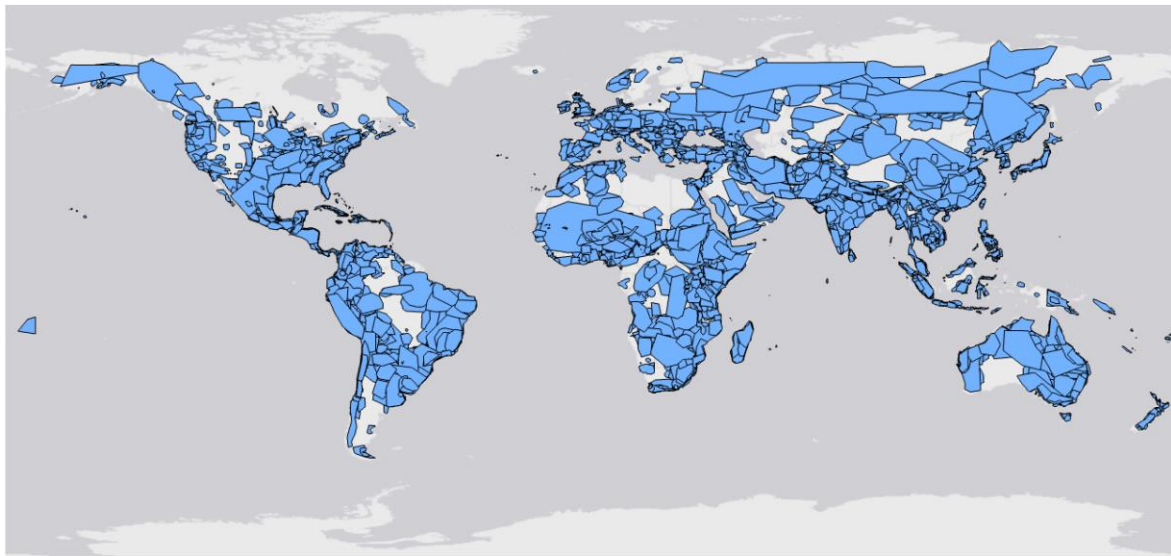


Abbildung 56: Ausgangsdatensätze der praktischen Formatevaluation ([GLO-18b])

Wie in Abbildung 56 ersichtlich, werden alle Polygone, welche seit dem Jahr 1985 visualisiert wurden, in einer Datei gesammelt. Dies impliziert, dass sich diverse Polygone von Überschwemmungsereignissen überlagern. Mit dem aktuellen Stand am 21.01.2019 enthält der Datensatz insgesamt 4056 Polygone. Verwendet man den kompletten Datensatz für die praktische Analyse dieser Masterarbeit, so würde der Zeitrahmen, unter den momentanen Möglichkeiten der Umsetzung, gesprengt werden. Aus diesem Grund wird ein Ausschnitt der Daten herangezogen. Mit durchschnittlich 100 – 150 Überschwemmungen pro Jahr ist es ausreichend, den Datenauszug auf ein halbes Jahr zu beschränken. Hierfür wird die erste Hälfte des Jahres 2017 gewählt.

Folgend wird nun in Kapitel 8.1.2 die Global Story umgesetzt. Durch die Dokumentation des Arbeitsprozesses wird auch auf Probleme hingewiesen, welche es gilt im Zuge der zukünftigen, direkten Vektorintegration von Linien- und Polygondaten am sphärischen Display zu verhindern.

8.1.2 Umsetzung einer Global Story für Global Flood Events

In den folgenden Unterkapiteln werden die Datensätze und deren Umsetzung für eine Global Story diskutiert. Als Hintergrundkarte für die Global Story wird die EarthByNight Plattkarte der NASA verwendet (siehe Abb. 57). (vgl. [EAR-17])

Da die Polygone und Punktdaten in ihrer Animation oft sehr verteilt am Globus erscheinen, bringt diese Basiskarte mehr Leben in das Globenbild. Durch den zusätzlichen, dunklen Farbraum der Karte kann im Zuge der Visualisierung von Polygonen und Punkten mehr mit hellen und kontrastreichen Farben gearbeitet werden.

Um die Hintergrundkarte korrekt am Hyperglobus zu visualisieren, wird die Bildgröße in Adobe Photoshop auf 4096 px * 2048 px festgelegt.



Abbildung 57: EarthByNight Plattkarte ([EAR-17])

8.1.2.1 Der Punktdatensatz

Der Punktdatensatz wird herangezogen, um die Attribute der Überschwemmungsereignisse mit den Flächenpolygonen zu kombinieren. Dies bietet NutzerInnen einen Mehrwert durch eine erweiterte Informationsvermittlung. Hierfür wird zunächst die Tabelle des Datensatzes angepasst. Der Datensatz muss angepasst werden, damit die Geoobjekte und ihre Attribute in OmniSuite importiert werden können. Ohne einen korrekten Aufbau der Datei können die Punktobjekte nicht am sphärischen Display visualisiert werden.

	Name Box	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	GlideNumber	Country	OtherCountry	long	lat	Area	Began	Ended	Validation	Dead	Displaced	MainCause	Severity
2	1	0	Algeria	0	5.23026	35.8142	92615.67	1/1/1985	1/5/1985	News	26	3000	Heavy rain	1
3	2	0	Brazil	0	-45.3489	-18.7111	678499	1/15/1985	2/2/1985	News	229	80000	Heavy rain	2
4	3	0	Philippines	0	122.974	10.0207	12846.03	1/20/1985	1/21/1985	News	43	444	Torrential rain	1
5	4	0	Indonesia	0	124.606	1.01489	16542.12	2/4/1985	2/18/1985	News	21	300	Torrential rain	1
6	5	0	Mozambique	0	32.3491	-25.8693	20082.21	2/9/1985	2/11/1985	News	19	0	Heavy rain	2
7	6	0	Comoros	0	43.36	-11.6516	1035.61	2/16/1985	2/28/1985	News	2	35000	Tropical cyclone	1
8	7	0	New Zealand	0	175.734	-37.2305	7871.37	2/17/1985	2/18/1985	News	4	200	Heavy rain	1
9	8	0	Indonesia	0	108.14	-7.04008	77091.11	2/19/1985	2/23/1985	News	10	2000	Heavy rain	1
10	9	0	USA	0	-85.1742	40.6691	210528	2/22/1985	3/1/1985	News	7	2250	Rain and snow	2

Tabelle 20: Tabelle des Datensatzes (Ausgangsform) (eigene Darstellung)

Die Datei (siehe Tab. 20) enthält Attribute für jedes Flut-Ereignis. In einem ersten Schritt wird der gewählte Zeitraum selektiert und in ein Libre-Office Dokument kopiert. Beim Zeitraum handelt es sich, wie bereits in Kapitel 8.1.1 genannt, um Daten zwischen dem 1.1.2017 bis 30.06.2017. Nachdem die Daten in ein neues Dokument kopiert wurden gilt es, die Daten für eine Implementierung in OmniSuite aufzuarbeiten.

Zuerst werden die Spalten der Latitude [lat] und Longitude [long] in dieser Reihenfolge in die ersten beiden Spalten des Dokumentes kopiert. Die folgenden Spalten können in einer beliebigen Reihenfolge mit gewählten Attributen gefüllt werden. Neben der Verortung der Punkte, sollen die folgenden Attribute durch den Punktdatensatz dargestellt werden:

- **Verortung** [lat/long]: Punkt
- **Betroffenes Land** [Country]: Beschriftung
- **Grund der Überflutung** [MainCause]: Farbe
- **Stärke der Überflutung** [Severity]: Größe der Punkte
- **Zeitpunkt der Überflutung** [Began/Ended]: Zeitliche Animation der Punkte

Nachdem diese Auswahl getroffen wurde, können die nicht benötigten Spalten entfernt werden. *(Natürlich können auch andere Attribute gewählt werden, abhängig von der Fragestellung)* Generell werden alle Spaltennamen klein geschrieben. Die Spalte [Country] wird in [name] umbenannt, Severity in [value]. Zusätzlich werden Leerräume zwischen Ländernamen durch ein %20 Zeichen ersetzt. Die Ursachen der Überschwemmung werden zusammengefasst, gruppiert und mit einer Zahl pro Kategorie ausgewiesen [category], um diese im CSS-File anzusprechen. (siehe Tab. 21)

- **Heavy Rain / Snowmelt** → Kategorie 1
- **Torrential Rain** → Kategorie 2
- **Monsoon** → Kategorie 3
- **Storm / Tropical Storm / Cyclone / Hurricane** → Kategorie 4

Heavy Rain	1
Tropical Storm Beatriz	4
Torrential Rain	2
Monsoonal Rain	3

Tabelle 21: Kategorien der Überflutungsursachen (eigene Darstellung)

In einem weiteren Schritt wird die Animationszeit der Punkte berechnet. Diese wird auch für die später folgende Animation der Polygone benötigt. Die Start- und Endzeit einer Überschwemmung sind im Datumsformat gegeben. Basierend darauf kann die Animationszeit berechnet werden. Die Autorin wählt für die Animation der Punkte der ersten Hälfte des Jahres 2017 56 Sekunden. (4 Sekunden werden getrennt für die Einleitung der Global Story verwendet.) Zusätzlich wird die Anzahl an Tagen benötigt. Diese beträgt 184. *(1.1.2017 – 3.7.2017, da die letzte Überschwemmung im Juni 2016 bis zum 3. Juli andauert.)* Es werden 2 neue Spalten [start] und [end] in der Tabelle angelegt. Die Formel für die Berechnung der Start- und Endzeit in Sekunden lautet wie folgt: (vgl. [GLO-18a: 92])

$$=\text{DAYS}(\text{E2}, \text{DATE}(2017,1,1)) / 184 * 55$$

Formel 2: Formel zur Berechnung der Animationszeiten ([GLO-18a: 92])

Die Funktion DAYS berechnet die Zeit zwischen zwei Daten: dem Tag des Auftretens der Überschwemmung und dem ersten Tag des Jahres. Die Anzahl an Tagen seit dem ersten Tag des Jahres wird durch die Gesamtzahl an Tagen (184) dividiert. Diese Zahlenwerte befinden sich zwischen 0 und 1. Dieses Ergebnis wird noch mal 55 multipliziert, der Länge der Animation. (vgl. [GLO-18a: 92]) (*E2 in der Formel bezieht sich auf die Spalte, von welcher das Datum bezogen wird.*) Die Formel selbst wird für alle Attribute der Spalten [start] und [end] verwendet. Sind die Animationszeiten berechnet, wird der Typ der Spalte in Text geändert. (*Rechtsklick in der Spalte → Format Cells*). Nach diesem Schritt können die Spalten [Began] und [Ended] aus der Tabelle entfernt werden. Die nun enthaltenen Spalten mit Attributinformationen zu jedem Ereignis sind [lat], [long], [name], [category], [start], [end] und [value].

1	lat	long	category	name	start	end	value
2	35.0839	-119.043	1	USA	6.28	7.08	1
3	3.93172	102.461	1	Malaysia	7.25	12.05	2
4	7.59046	124.062	2	Philippines	10.00	12.05	1.5
5	-5.25083	-77.573	1	Peru	10.00	12.05	2
6	-11.0042	-75.1481	3	Peru	10.00	25.15	2
7	38.804	-120.825	2	USA	14.28	20.01	1
8	-18.6428	29.022	2	Zimbabwe	14.28	20.01	1.5
9	-27.2717	30.5561	1	South%20Africa	15.05	20.01	1.5
10	-7.0293	107.162	1	Indonesia	16.14	20.01	1
11	3.72258	-72.6737	1	Colombia	17.03	20.01	2

Abbildung 58: Tabelle des aufbereiteten Datensatzes (Finale Form) (eigene Darstellung)

Das Dokument wird nun in Libre-Office im *.csv Format exportiert, wobei als Trennzeichen {space} definiert wird. Dieses Dokument wird in ein Textformat (*.txt) umbenannt und ist für einen Import in OmniSuite bereit.

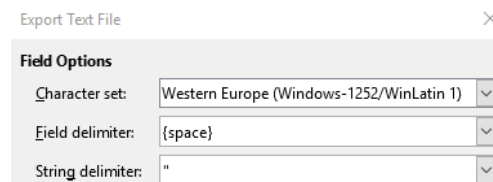


Abbildung 59: CSV-Export in Libre Office (eigene Darstellung)

Die fertige *.txt Datei lässt sich in Editoren öffnen und Spalten sind in dieser durch Leerzeichen getrennt. (siehe Abb. 60)

```

1 lat long name category start end value
2 -20.7778 34.5151 Mozambique 1 0.00 11.06 1.5
3 54.7053 9.58328 Germany 4 0.30 1.20 1.5
4 39.6236 -122.458 USA 1 0.90 1.20 1.5
5 -29.4094 -55.5512 Brazil 1 1.20 5.98 1
6 1.6911 -76.2863 Colombia 1 1.20 11.06 1
7 15.5805 -85.3672 Honduras 1 1.79 5.98 1
8 8.08572 125.027 Philippines 1 4.48 5.98 1
9 35.0839 -119.043 USA 1 5.68 5.98 1
10 3.93172 102.461 Malaysia 1 6.58 11.06 1
11 7.59046 124.062 Philippines 1 9.27 11.06 1.5
12 -5.25083 -77.573 Peru 1 9.27 11.06 1

```

Abbildung 60: Ausschnitt der *.txt Datei (eigene Darstellung)

Neben der *.txt Datei, wird für Punktgeometrien ein Stylesheet (CSS-Dokument) benötigt, in welchem die Visualisierung der Punkte am Hyperglobus definiert wird. Dieses Dokument liegt unter Material/Styles des definierten OmniSuite Verzeichnisses.

```

1  .category1 {
2    marker-symbol: ellipsoid;
3    marker-width: 25 * [value] * 2.1;
4    marker-height: 25 * [value] * 2.1;
5    marker-color: rgba(219, 241, 252, 0);
6    marker-outline-style: solid;
7    marker-outline-width: 4px;
8    marker-outline-color: rgba(219, 241, 252, 255);
9    text-name: [name];
10   text-dy: 5;
11   text-placement: S;
12 }

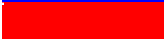
```

Abbildung 61: Ausschnitt einer Kategorie des CSS-Dokuments der *.txt Datei (eigene Darstellung)

Im Stylesheet werden die folgenden Punkte definiert:

- **marker-symbol:** ellipsoid → Form des Objekts
- **marker-width und marker-height:** Höhe und Breite des Ellipsoids (ident da Punkt) * [value] → Stärke der Überschwemmung pro Objekt
- **marker-color:** RGB Farbcode (0 - 255) + Angabe der Transparenz (0 - 255)
- **marker-outline-style:** Art der Außenlinie
- **marker-outline-width:** Stärke der Außenlinie
- **marker-outline-color:** RGBA Farbcode
- **text-name:** [name] → Definition der Spalte, aus welcher die Attribute pro Objekt am Globus als Text visualisiert werden sollen
- **text-dy:** 5 → Verschiebung des Textes entlang der Y-Achse in Pixel
- **text-placement:** S (Süd – unter dem Punktobjekt)

Da vier Kategorien definiert wurden, besteht die CSS-Datei aus insgesamt vier Blöcken (category1{} – category4{}). Der Unterschied in diesen liegt im RGBA Farbcode der Objektfüllung (marker-color). Hierfür wurde die folgende Farbwahl getroffen:

Kategorie 1	→	
Kategorie 2	→	
Kategorie 3	→	
Kategorie 4	→	

Nachdem das Textdokument und das Stylesheet erstellt sind, kann die Datei in die Globensoftware importiert werden. (siehe Abb. 62)

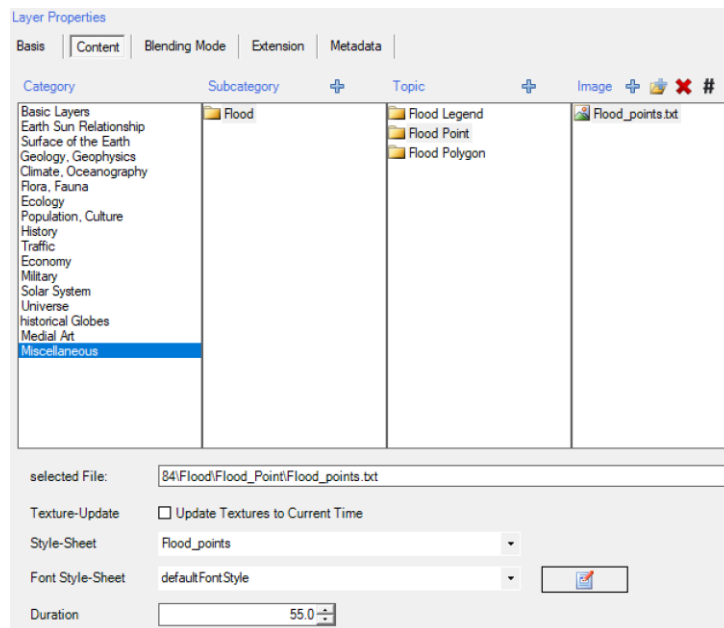


Abbildung 62: Import der *.txt Datei in OmniSuite (eigene Darstellung)

Unter Style-Sheet wird die gleich benannte CSS-Datei gewählt. Wichtig ist, dass diese sich im Verzeichnis der Globensoftware unter *Materials/Styles* befindet. Ist dies nicht der Fall, kann OmniSuite nicht auf die CSS-Datei zugreifen. Nachdem die Datei importiert und der Stylesheet ausgewählt ist, ist die Erstellung des Punktlayers für den Hyperglobus finalisiert. (siehe Abb. 63)



Abbildung 63: *.txt Dateivisualisierung in OmniSuite (eigene Darstellung)

8.1.2.2 Der Polygon-Datensatz

Bevor mit der Umsetzung und Visualisierung der Flächen durch die Globensoftware begonnen wird, muss der Polygon-Datensatz in einem ersten Schritt im GIS aufbereitet und exportiert werden. Für diesen Schritt wird die Software ArcGIS Pro verwendet. *(Dies ist eine persönliche Wahl, der Schritt kann auch mit der Open-Source Software QGIS durchgeführt werden.)*

Bereits in Kapitel 8.1 wurde erwähnt, dass mit Daten des Jahres 2017 gearbeitet wird. Wie in Abbildung 64 zu sehen, überlappen sich einige Überschwemmungsregionen flächenmäßig. Da nach dem Datenexport noch eine weitere kartographische Adaption der Daten in einem Bildbearbeitungsprogramm stattfindet, dürfen sich Polygone nicht überlappen. Aus diesem Grund werden die Daten etappenweise exportiert. In der Attributtabelle werden jeweils die Flood-Events (Objekte) eines Monats gewählt und als gesonderter Layer exportiert.

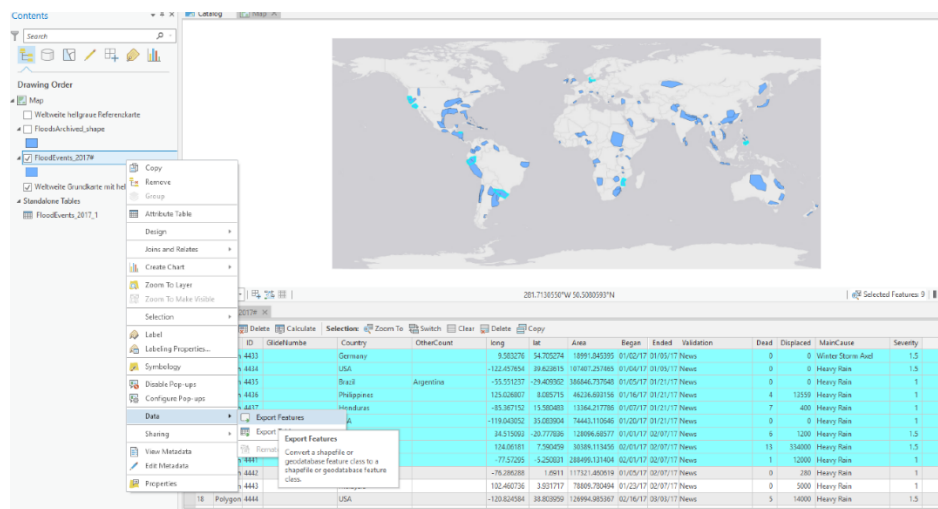


Abbildung 64: Export der Polygone der Flood Events im Jahr 2017 (Datengrundlage: [GLO-18b])

Es muss darauf geachtet werden, dass sich auch in diesem kurzen Zeitraum keine Objekte überlappen. Sollte dies doch der Fall sein, so muss der Objektexport noch einmal gesplittet werden. Eine andere Option wäre der Export des Layouts im *.svg Format. In dieser Datei würden auch überlappende Polygone objektweise abgespeichert werden. Jedoch impliziert dieser Schritt weitere Arbeitsschritte in einer Software für Vektordaten, wie unter anderem Adobe Illustrator oder Inkscape. Die Autorin entscheidet sich daher dafür, die Vektordaten bereits im GIS aufzusplitten. (siehe Abb.65)

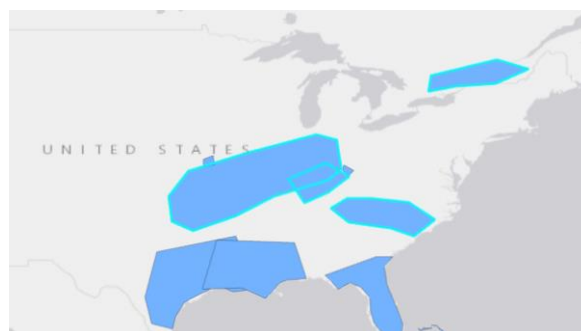


Abbildung 65: Überlappende Polygone Monat Mai 2017 ([GLO-18b])

Bevor mit der Bearbeitung im Bildbearbeitungsprogramm begonnen werden kann, müssen die Daten in der korrekten Ausdehnung als Rasterbild exportiert werden. In einem ersten Schritt wird das projizierte Koordinatensystem „Plate Carree (world)“ eingestellt. Danach wird im Layout View die Seitenausdehnung definiert. Da ArcGIS nicht mit der Einheit Pixel arbeitet, wird die Größe des Kartenlayouts auf das Verhältnis (Höhe: Breite) 40,96 cm: 20,48 cm festgelegt. Zusätzlich wird die Position des Rahmens auf X:0 und Y:0 gesetzt, um die exportierten Layer in der linken, unteren Ecke der Platkarte zu platzieren. Mit der zusätzlichen Breitendefinition (Width) des Kartenrahmens (40,96 cm) wird garantiert, dass sich die Vektordaten exakt in den definierten Maßen der Platkarte befinden. (siehe Abb. 66)

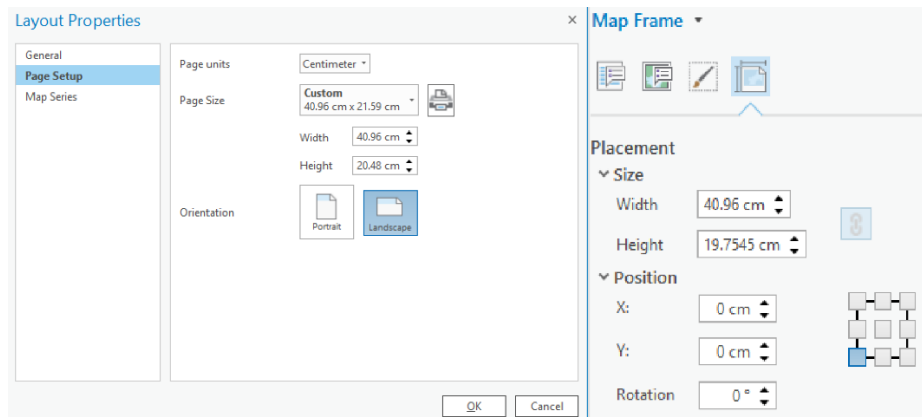


Abbildung 66: ArcGIS Pro Rasterexport Einstellungen (eigene Darstellung)

Sind diese Parameter gesetzt, kann der Export der Rasterbilder (*.png) für jeden zuvor erstellten Vektorlayer durchgeführt werden. Bei diesem Datensatz handelt es sich somit um die gesplitteten Monatspolygone von Überflutungen des Jahres 2017.

Die nächsten Arbeitsschritte finden nun in der Bildbearbeitungssoftware Adobe Photoshop statt. Dafür wählt die Autorin Adobe Photoshop, weil bereits in der Lehrveranstaltung Multimediale Technologie und Geokommunikation mit dieser Software gearbeitet wurde. Da es sich bei Zentimeter um eine andere Einheit als Pixel handelt, muss für alle exportierten Bilder in Photoshop die Größe unter *Image* -> *Image Size* auf 4096 px : 2048 px geändert werden. In einem weiteren Schritt wird der weiße Hintergrund auf Transparent geändert. Hierfür wird mit dem Zauberstab-Tool  gearbeitet, um Bereiche einer Farbe zu selektieren. Ist der weiße Bereich selektiert, kann dieser über die Taste „DEL“ oder „Entf“ gelöscht werden. Nun ist der Hintergrund transparent. (siehe Abb. 67)

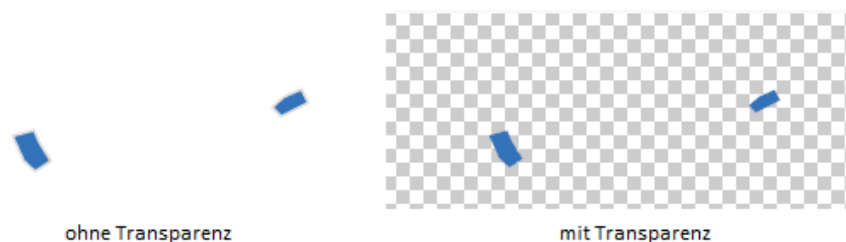


Abbildung 67: Rasterbild ohne/mit Transparenz (eigene Darstellung)

Mit dem Zauberstab-Tool kann die Selektion über *Rechtsklick -> Select Inverse* umgekehrt werden. Über Strg + C werden alle Polygone pro Rasterbild, welches aus ArcGIS Pro exportiert wurde, in eine neue Layer-Ebene der Photoshop Datei kopiert. (siehe Abb. 68)

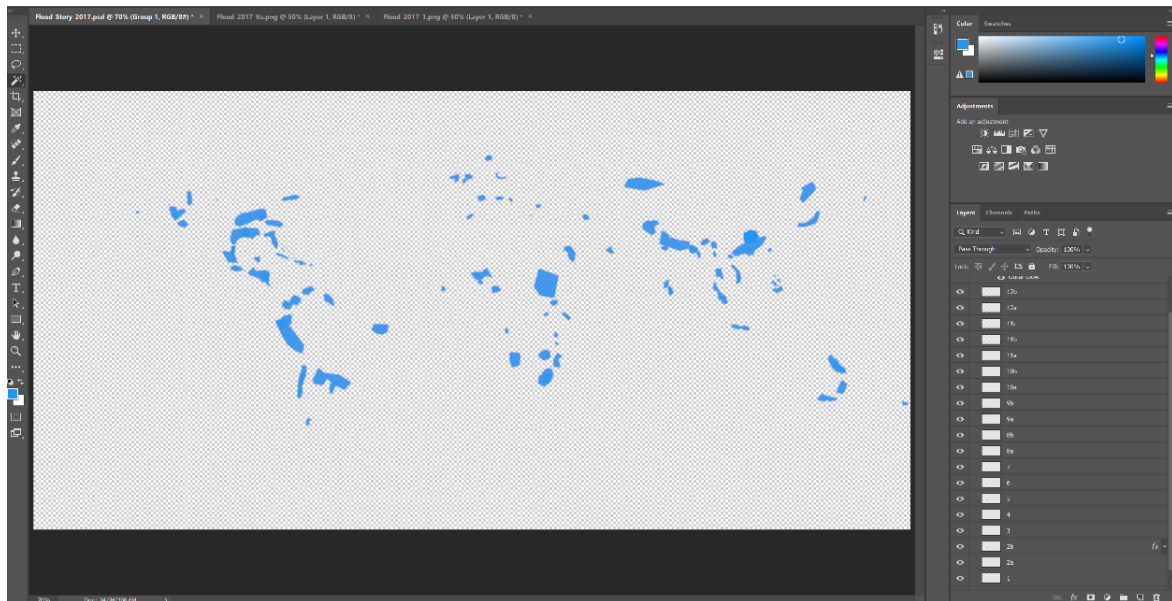


Abbildung 68: Polygone aufbereitet in Layer-Ebenen (eigene Darstellung)

Nach einer kurzen graphischen Bearbeitung der Layer, bei welcher sie eine idente Farbgebung bekommen (grau für Überflutungsregionen), wurde eine leichte Transparenz (Opacity: 70%) definiert, um Landflächen unter den Flutpolygone nicht zu verdecken. *(Natürlich können an diesem Punkt noch diverse graphische Adaptionen durchgeführt werden. Jedoch liegt der Fokus auf der Darstellung der arbeitsintensiven Erstellung einer Global Story durch das Verfahren der indirekten Rastervisualisierung am Hyperglobus.)*

Nachdem die Grundlage für eine Visualisierung am sphärischen Display geschaffen ist, können die Polygone nach ihrem zeitlichen Auftreten animiert werden. Für diesen Schritt wird die Tabelle des bearbeiteten Punktdatensatzes benötigt. Basierend auf diesem werden die Polygone in Photoshop animiert. Hierfür wird die Funktion der Zeitleistenanimation in Photoshop angewandt.

Die Timeline in Photoshop ist in Frames unterteilt. Unter Timeline Frame Rate lässt sich die Framerate einstellen. Für Visualisierungen am Hyperglobus werden Bildsequenzen idealerweise mit 30 frames per second exportiert. In Abbildung 69 sieht man, dass die Timeline zwischen Sekunden in 30 Frames unterteilt ist. Mit der [start] und [end] Berechnung in der Tabelle aus Kapitel 8.1.2 wird eine Animation schwierig, da diese mit hundertstel Sekunden angegeben sind. Daher müssen die Werte nach dem Komma (10.xx) in einer weiteren Spalte auf die 30 Frames pro Sekunde umgerechnet werden. Dies garantiert eine korrekte und übereinstimmende Animation in Photoshop.

Hierfür werden die Spalten [startcomma] und [endcomma] im Tabellendokument mit den bereits berechneten Animationszeiten der Punktobjekte eingefügt. Über die folgende Formel werden die hundertstel Sekunden auf 30 Frames pro Sekunde umgerechnet.

$$=\text{ROUND}(\text{MOD}(\text{G2},1) * 30,0)$$

Formel 3: Formel zur Berechnung der Polygonanimation

Der Befehl MOD nimmt den Restwert einer Zahl bei Division. Da durch 1 dividiert wird, bleibt immer 0 und die Kommastellen der Animationszeit über. Dieser Wert wird mit 30 multipliziert und durch den ROUND Befehl gerundet. Für die Animation in Photoshop wird somit die Ganzzahl aus Spalte [start]/[end] für die Sekunden verwendet. Die Frames zwischen Sekunden werden aus [startframes]/[endframes] bezogen. Basierend auf diesen berechneten Zeitangaben können die Flächen nun in Photoshop animiert werden.

G	H	I	J
start	startframes	end	endframes
0.00	0.00	11.06	2.00
0.30	9.00	1.20	6.00
0.90	27.00	1.20	6.00
1.20	6.00	5.98	29.00
1.20	6.00	11.06	2.00

Tabelle 22: Tabelle mit berechneten Frames per Second (eigene Darstellung)

Zuerst jedoch werden die Polygone jedes Layers in der Reihenfolge ihres Auftritts pro Monat gesplittet. Dieser Schritt wird wieder mit dem Zauberstab-Tool durchgeführt (*Eine weitere Möglichkeit für die Animation in Photoshop ist das sogenannte Mask-Tool. An dieser Stelle wird nicht näher auf diese Möglichkeit eingegangen und mit der ersten Option gearbeitet.*) Nachdem die Polygone in einzelne Layer kopiert und nach ihrer Reihenfolge benannt wurden, wird die Animation durchgeführt.

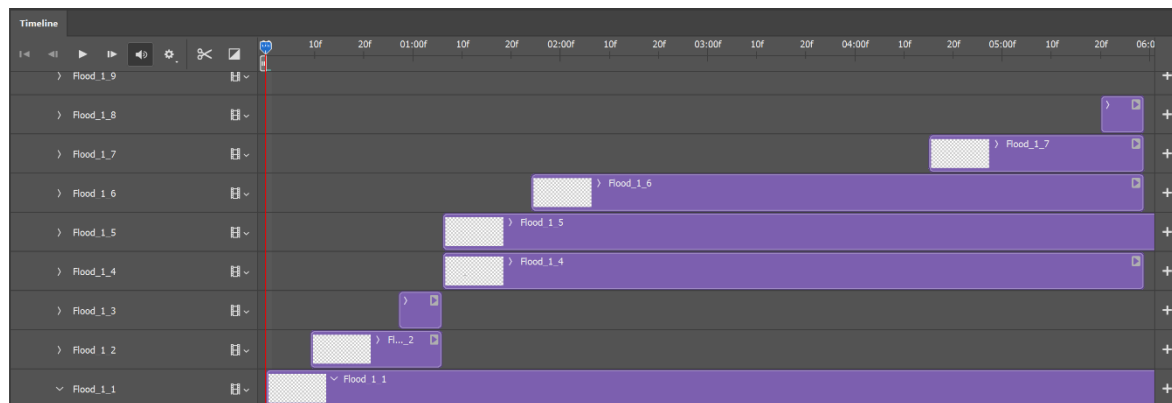


Abbildung 69: Animation der einzelnen Polygone in Photoshop (eigene Darstellung)

Ist die Animation durchgeführt, kann diese als Bildsequenz aus Photoshop exportiert werden. Die Bildsequenzen werden im PNG-Format gespeichert. Dies ist notwendig, da der Alpha-Kanal (Transparenz) mitgespeichert werden muss (siehe Kapitel 3.1.1). Es wird die Renderoption: *Alpha Channel – Straight Unmatted* ausgewählt. Dadurch wird ermöglicht, dass in der Globensoftware eine Kombination aus Hintergrundkarte, Polygonen und Punktobjekten erstellt werden kann. (siehe Abb. 70)

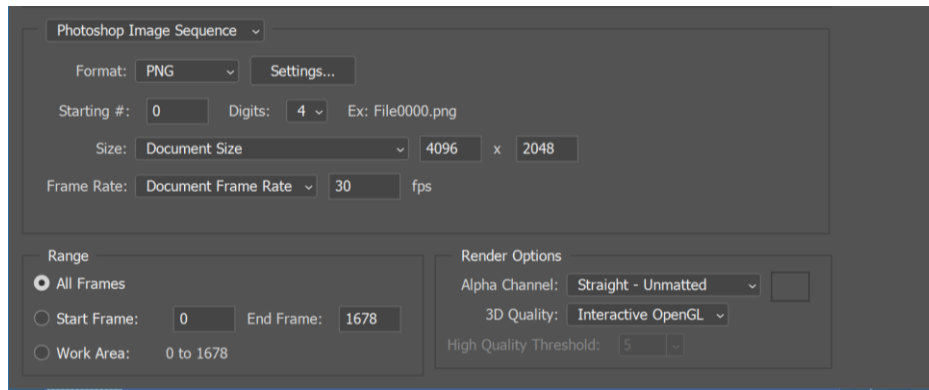


Abbildung 70: Render-Einstellung in Photoshop (eigene Darstellung)

Nachdem die Bildsequenz erstellt wurde, werden auch die gerenderten Bilder über dem MaterialEditor die Globensoftware importiert. Da die Bilder im *.png Format exportiert erstellt wurden und OmniSuite aus Performancegründen das DDS-Format empfiehlt, findet während dem Importvorgang eine Konvertierung in DXT5 statt. (siehe Abb. 71)

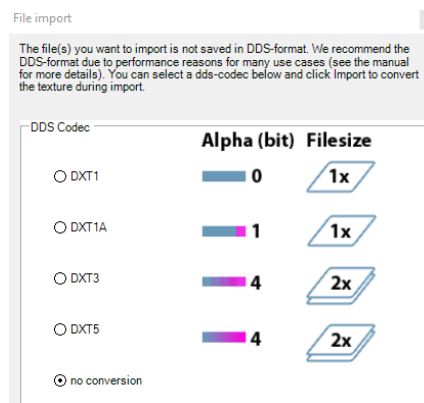


Abbildung 71: DDS-Konvertierung in OmniSuite (eigene Darstellung)

Ist der Import vollständig, wird das Zeitintervall von 1680 Bilder in 56 Sekunden definiert. (siehe Abb. 72)

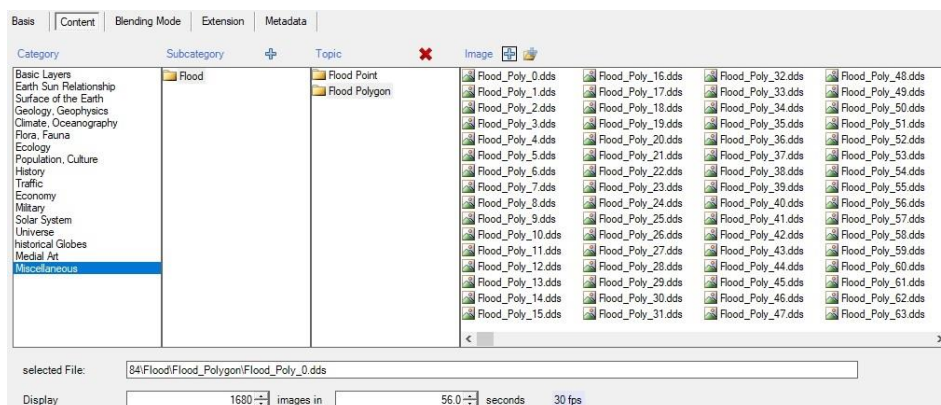


Abbildung 72: Rasterimport in OmniSuite (eigene Darstellung)

Aus den importierten Punkt- (Kap. 8.1.2.1) und Polygondaten wird nun ein Material erstellt. Im ersten Layer liegt die Hintergrundkarte (EarthByNight). Polygone werden in Layer 2 definiert, Punkte in Layer 3. Dieses Material wird im StoryEditor als Material 1 verwendet.

Um den Endnutzern auch Information über die visualisierten Informationen am Hyperglobus zu bieten, wird eine zusätzliche Legende in Photoshop erstellt (siehe Abb. 73). In dieser ist auch eine Zeitleiste enthalten, welche den Verlauf von Flutereignissen im ersten Halbjahr 2017 zeitlich festhält. Die Animation dieser Zeitleiste beruht auf der Zeitberechnung aus Kapitel 8.1.2.1. Ist diese vollständig, wird sie auch im kleinstmöglichen Format 2ⁿ (*Image -> Canvas Size -> 1024:512*) als Bildsequenz gerendert und als DXT5 Format OmniSuite importiert.



Abbildung 73: Legende der Global Story (eigene Darstellung)

Auch die Legende wird mit Alpha-Kanal exportiert. Wurde sie im Format DXT5 die Globensoftware importiert und das Zeitintervall definiert, wird unter *Layer Properties -> Extension* die Ausdehnung auf der Hintergrundkarte definiert. Da die Sequenzen der Legende in einer geringeren Größe gerendert wurden, wird die Ausdehnung dementsprechend festgelegt. Über die Option *repeat Texture horizontally* wird die Legende insgesamt drei Mal um den Globus angeordnet (siehe Abb. 74). Die Legende wird als eigenes Material abgespeichert.

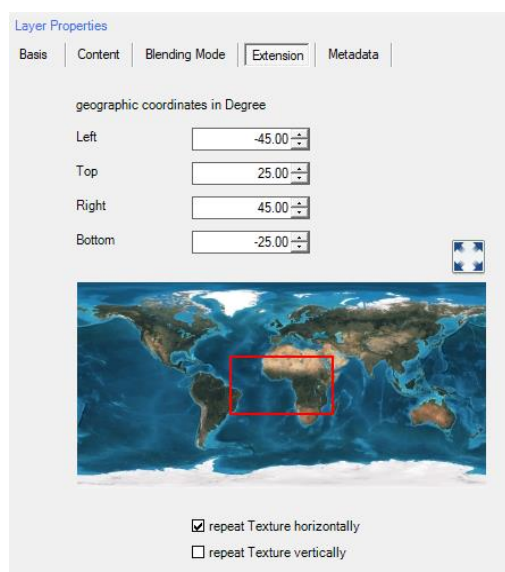


Abbildung 74: Definition der Legendenausdehnung in OmniSuite (eigene Darstellung)

8.1.3 Erstellen der Global Story für den Testdatensatz

Nachdem Material 1 und 2 erstellt wurden, wird die Global Story im OmniSuite StoryEditor umgesetzt. Die Zeit (Duration) wird für 56 Sekunden festgelegt. In der Spur Material wird das am Ende von Kapitel 8.1.2 erstellte Material *FloodMain* und in der Spur Material 2 das umgesetzte Material *FloodLegend* festgelegt. Zusätzlich werden für die Spur Material 1 über die Rotation gewählte Positionen am Globus zu bestimmten Zeitpunkten definiert. (siehe Abb. 75)

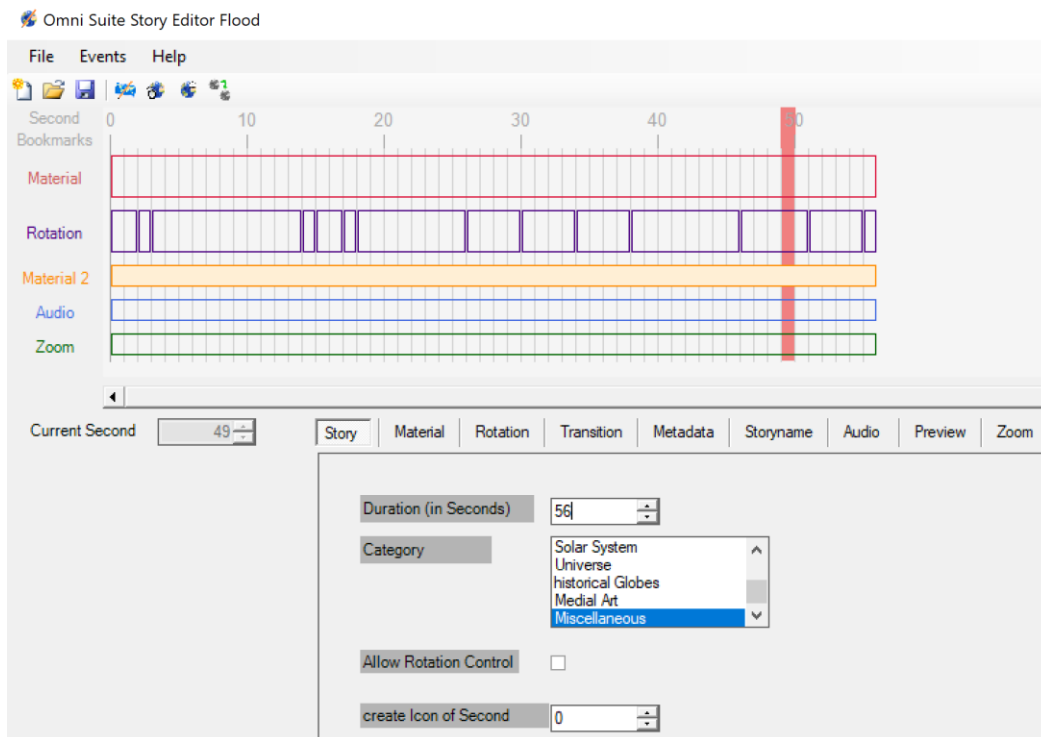


Abbildung 75: Erstellung der Global Story im StoryEditor (eigene Darstellung)

Da Materiale und Rotationen festgelegt wurden, ist die Global Story fertig erstellt. (Abb.)

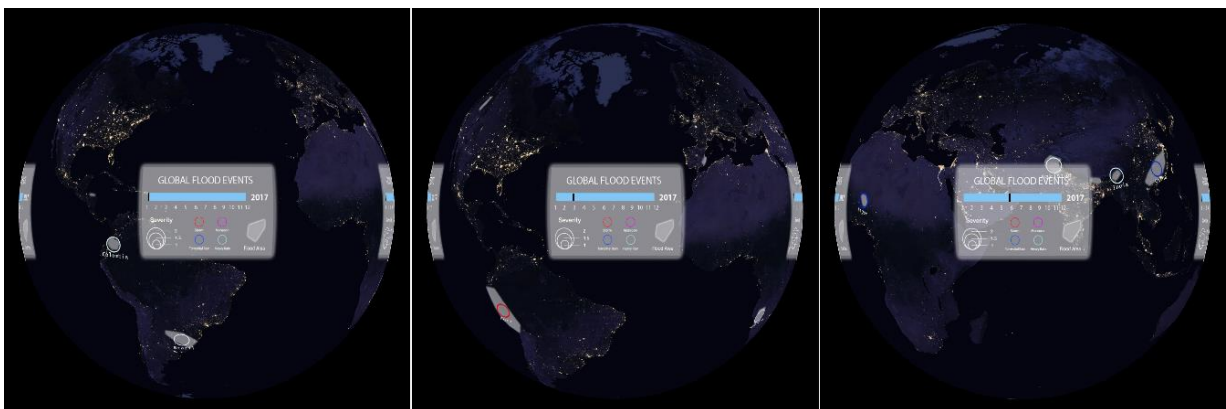


Abbildung 76: Global Story zu verschiedenen Zeitpunkten (eigene Darstellung)

8.1.4 Analyse der Erstellung einer Global Story nach dem aktuellen Ansatz

Während der Prozesse in Kapitel 8.1.2.1 – 8.1.2.2 wurde verdeutlicht, worin der Unterschied zwischen einer externen, asynchronen und internen, direkten Visualisierung am sphärischen Display liegt.

Der Punktdatensatz stand in einem Tabellenformat zur Verfügung. Nach einer Aufarbeitung der Tabellenattribute konnten seine Flut-Objekte direkt am Globus visualisiert werden. Kategorisierungen, zeitliche Animationen und Klassifizierungen wurden objektbasiert in einem CSS-Dokument definiert. Vorteile sind, dass der Zeitaufwand gering gehalten wird und Punkte objektbasiert angesprochen werden. Im Zuge eines Quiz könnten Überschwemmungs-Orte zusätzlich, je nach ihren Eigenschaften, interaktiv von NutzerInnen angesprochen werden. Diese Aspekte zeigen die großen Vorteile der direkten, synchronen Visualisierung am Hyperglobus auf.

Für die Visualisierung von Polygonen am sphärischen Display bedarf es weit mehrerer Schritte. Zuerst mussten diese im GIS aufbereitet und in der korrekten Ausdehnung exportiert werden. Danach wurden Layer in einer Bildbearbeitungssoftware graphisch aufbereitet. Um eine Animation durchzuführen, wurde jedes Flut-Objekt in eine eigene Ebene kopiert. Dieser Prozess und jener der Polygon-Animation sind zeitkonsumierend. Zusätzlich werden manuell umgesetzte Animationszeitpunkte nie die Genauigkeit jener Datensätze besitzen, welche direkt, synchron am Hyperglobus visualisiert werden. Ein weiterer Aspekt ist, dass der Ausgangsdatsatz im Shapefile-Format diverse Attribute enthielt. All jene Objektinformation ging durch die Konvertierung in ein Raster-Format verloren. (siehe Tab. 23)

Datensatz	Programme und Schritte		Zeitaufwand		Dateien + Speicherbedarf		Objektbezug	
Punkt	Tabellenadaptierung Umspeichern der Tabelle (*.txt) Notepad++	±	~3h	+	2 Dateien (*.txt + *.css) 2.46 KB/1.27 KB	+	ja	+
Polygon	Aufbereitung im GIS Bildbearbeitungsprogramm Animation	±	~8h	-	1680 Bilder (*.png) 85 MB	-	nein	-

Tabelle 23: Aspekte der Umsetzung der Global Story (eigene Darstellung)

Verdeutlicht wird die Problematik der Rastervisualisierung am sphärischen Display dadurch, dass es sich bei dem Datensatz um einen kleinen Datenauszug handelt. Die Originaldaten bestehen aus mehr als 4700 Polygonen. Dies nach dem momentanen externen, synchronen Ansatz im Zeitverlauf am Hyperglobus umzusetzen ist eine (zeitlich) schwer bewältigbare Aufgabe.

Daher werden nun in Kapitel 8.2 die in Kapitel 7.5 gewählten Vektorformate praktisch auf ihre Vor- und Nachteile für eine geplante Implementierung am Hyperglobus untersucht.

8.2 Praktische Analyse der gewählten Formate

Wie in Kapitel 7.5 diskutiert, werden die Formate *.kml, *.svg und *.geojson für die folgende praktische Formatanalyse herangezogen. Hierzu wird der Test-Polygondatensatz aus Kapitel 8.1.2 in die jeweiligen Formate konvertiert und evaluiert. *(Das OmniSuite Punkt-Format wird an dieser Stelle nicht weiter untersucht, da Punkte bereits mit der internen, synchronen Methode am Hyperglobus visualisiert werden können. Das OmniSuite Punktformat wurde bereits im Kapitel 7.4.6 vorgestellt und in Kapitel 8.1.2.1 praktisch eingesetzt.)*

Bevor die praktische Analyse beginnt, werden an dieser Stelle die Fragestellungen dieser aufgelistet:

Wie können ...

- die **graphische Aufbereitung**,
- **Animation** und
- **Interaktion**

... in einer praktischen Analyse mit einem Testdatensatz umgesetzt werden?

Zusätzlich fließen die folgenden, noch offenen Forschungsfragen in praktische Analyse ein:

- Existieren für die jeweiligen Vektorformate **Programmbibliotheken in C++**?
- Welche Unterschiede gibt es in der **Speichergröße** der jeweiligen Formate?

Für die Ergebnisse der theoretischen Analyse und Basisinformation zu den nun folgenden Vektorformaten wird auf Kapitel 7 verwiesen. Die folgenden Konvertierungen pro Format werden in der Software ArcGIS Pro durchgeführt. Dieser Prozess ist jedoch, wie in Tabelle 18 ersichtlich, genauso in QGIS möglich.

8.2.1 KML

Der Testdatensatz wird in ArcGIS Pro unter *Toolboxes → Conversion Tools → Layer To KML* in das KML-Format konvertiert. Da die KML-Datei von jenem Tool nur im *.kmz-Format konvertiert werden kann, wird das FME Tool *Data Interoperability → Quick Export → KML* verwendet. (siehe Abb. 77)

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <xml xmlns="http://www.opengis.net/kml/2.2">
3   <Document id="root_doc">
4     <Schema name="Flood 2017 KML" id="Flood 2017 KML">
5       <SimpleField name="ID" type="int"/></SimpleField>
6       <SimpleField name="GlideNumber" type="string"/></SimpleField>
7       <SimpleField name="Country" type="string"/></SimpleField>
8       <SimpleField name="OtherCount" type="string"/></SimpleField>
9       <SimpleField name="long" type="float"/></SimpleField>
10      <SimpleField name="lat" type="float"/></SimpleField>
11      <SimpleField name="Area" type="float"/></SimpleField>
12      <SimpleField name="Began" type="string"/></SimpleField>
13      <SimpleField name="Ended" type="string"/></SimpleField>
14      <SimpleField name="Validation" type="string"/></SimpleField>
15      <SimpleField name="Dead" type="int"/></SimpleField>
16      <SimpleField name="Displaced" type="int"/></SimpleField>
17      <SimpleField name="MainCause" type="string"/></SimpleField>
18      <SimpleField name="Severity" type="float"/></SimpleField>
19    </Schema>
20    <Folder name="Flood 2017 KML">
21      <Placemark>
22        <Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>0</fill></PolyStyle></Style>
23        <ExtendedData><SchemaData schemaUri="#Flood 2017 KML">
24          <SimpleData name="ID">4433</SimpleData>
25          <SimpleData name="Country">Germany</SimpleData>
26          <SimpleData name="long">9.583276000000000</SimpleData>
27          <SimpleData name="lat">54.705274000000003</SimpleData>
28          <SimpleData name="Area">18991.845394600000673</SimpleData>
29          <SimpleData name="Began">2017/01/02</SimpleData>
30          <SimpleData name="Ended">2017/01/05</SimpleData>
31          <SimpleData name="Validation">News</SimpleData>
32          <SimpleData name="Dead">0</SimpleData>
33          <SimpleData name="Displaced">0</SimpleData>
34          <SimpleData name="MainCause">Winter Storm Axel</SimpleData>
35          <SimpleData name="Severity">1.500000000000000</SimpleData>
36        </SchemaData></ExtendedData>
37        <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13.532608,54.306792 13.33466,53.662268 10.414927,53.415802 9.425187,54.457616 9.227239,55.251098 9.524161
38      </Placemark>

```

Abbildung 77: Ausschnitt aus der KML-Datei des Testdatensatzes (eigene Darstellung)

8.2.1.1 Graphische Aufbereitung - KML

Zur kartographischen Stildefinition von KML-Objekten gibt es, wie in Kapitel 7.4.2 bereits erläutert, mehrere Varianten. Nachdem jedes Objekt in einer KML-Datei als Placemark definiert ist, besitzt jedes Objekt eine Stilinformation. Im konvertierten Testdatensatz ist diese Information bereits für jedes Objekt vergeben. (siehe Abb. 78)

```

<Placemark>
  <Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>0</fill></PolyStyle></Style>
  <ExtendedData><SchemaData schemaUri="#Flood 2017 KML">
    <SimpleData name="ID">4433</SimpleData>
    <SimpleData name="Country">Germany</SimpleData>
    <SimpleData name="long">9.583276000000000</SimpleData>
    <SimpleData name="lat">54.705274000000003</SimpleData>
    <SimpleData name="Area">18991.845394600000673</SimpleData>
    <SimpleData name="Began">2017/01/02</SimpleData>
    <SimpleData name="Ended">2017/01/05</SimpleData>
    <SimpleData name="Validation">News</SimpleData>
    <SimpleData name="Dead">0</SimpleData>
    <SimpleData name="Displaced">0</SimpleData>
    <SimpleData name="MainCause">Winter Storm Axel</SimpleData>
    <SimpleData name="Severity">1.500000000000000</SimpleData>
  </SchemaData></ExtendedData>
  <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13.532608,54.306792 13.33466,53.662268
</Placemark>

```

Abbildung 78: KML Style-Definition im Testdatensatz nach Konvertierung (eigene Darstellung)

Zusätzlich können mehrere Objekte einer KML-Datei mit demselben Style ausgewiesen werden. Hierfür wird am Anfang der Datei eine Style-ID definiert. Wie bei der Umsetzung der Global-Story sollen die Polygone mit einem transparenten Grau eingefärbt werden. Die KML Farbcodes können über die Website Zonum (vgl. [KML-12]) abgerufen werden. Im 8-Digit Hex Code selbst wird die Transparenz mitdefiniert.

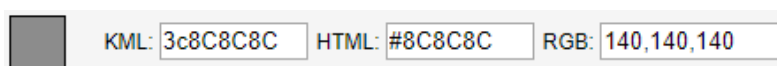


Abbildung 79: KML 8-Digit Farbcode im Vergleich ([KML-12])

Ist die Style-ID am Anfang des Dokumentes definiert, kann der Style-Tag jedes Objektes über die Funktion *Suchen* → *Ersetzen* durch die `<styleUrl>` ersetzt werden. (siehe Abb. 80)

```

21  <Style id="transGreyPoly">
22    <LineStyle>
23      <width>1</width>
24      <color>3c8C8C8C</color>
25    </LineStyle>
26    <PolyStyle>
27      <color>3c8C8C8C</color>
28    </PolyStyle>
29  </Style>
30  <Placemark>
31    <styleUrl>#transGreyPoly</styleUrl>
32    <ExtendedData><SchemaData schemaUrl="#Flood_2017_KML">
33      <SimpleData name="ID">4433</SimpleData>
34      <SimpleData name="Country">Germany</SimpleData>
35      <SimpleData name="long">9.583276000000000</SimpleData>
36      <SimpleData name="lat">54.705274000000003</SimpleData>
37      <SimpleData name="Area">18991.845394600000873</SimpleData>
38      <SimpleData name="Began">2017/01/02</SimpleData>
39      <SimpleData name="Ended">2017/01/05</SimpleData>
40      <SimpleData name="Validation">News</SimpleData>
41      <SimpleData name="Dead">0</SimpleData>
42      <SimpleData name="Displaced">0</SimpleData>
43      <SimpleData name="MainCause">Winter Storm Axel</SimpleData>
44      <SimpleData name="Severity">1.500000000000000</SimpleData>
45    </SchemaData></ExtendedData>
46    <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13.532608,54.306792
47  </Placemark>

```

Abbildung 80: `<styleURL>` im KML Objektcode (eigene Darstellung)

Möchte man die Polygonobjekte farblich kategorisieren, können Objekte mit einem bestimmten Attribut (Severity) kategorisiert werden. Google Earth selbst besitzt keine Möglichkeit, Objekte automatisch nach einem Attribut zu kategorisieren oder klassifizieren. Da der Datensatz jedoch sowieso aus einer GIS Software exportiert wird, kann die Klassifikation bereits vor der Konvertierung in der Software QGIS durchgeführt werden. Das Python Plugin MMQIS bietet die Möglichkeit, Vektordatensätze zu manipulieren. Die Option *Import/Export* → *Google Maps Google KML* konvertiert den Vektordatensatz ins KML-Format und übernimmt auch vorgenommene (farbliche) Klassifizierungen. Wie in Abbildung 81 zu sehen, werden jedoch nur Polygonfüllungen übernommen. Die Linienfarben der Polygone werden standardmäßig in Schwarz (`<color>40000000</color>`) gespeichert.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://earth.google.com/kml/2.2">
3 <Document>
104 <Style id="style10">
105 <LineStyle>
106 <color>40000000</color>
107 <width>3</width>
108 </LineStyle>
109 <PolyStyle>
110 <color>ff02f7f3</color>
111 <fill>1</fill>
112 <outline>1</outline>
113 </PolyStyle>
114 </Style>
115 <Style id="style11">
116 <LineStyle>
117 <color>40000000</color>
118 <width>3</width>
119 </LineStyle>
120 <PolyStyle>
121 <color>ff02f7f3</color>
122 <fill>1</fill>
123 <outline>1</outline>
124 </PolyStyle>
125 </Style>
126 <Style id="style12">
127 <LineStyle>
128 <color>40000000</color>
129 <width>3</width>
130 </LineStyle>
131 <PolyStyle>
132 <color>ff02f7f3</color>
133 <fill>1</fill>
134 <outline>1</outline>
135 </PolyStyle>
136 </Style>
137 <Style id="style13">
138 <LineStyle>
139 <color>40000000</color>
175 <Placemark>
176 <name>NULL</name>
177 <ExtendedData>
178 <Data name="Name"><displayName>Name</displayName><value><![CDATA[NULL]]></value></Data>
179 <Data name="description"><displayName>description</displayName><value><![CDATA[NULL]]></value></Data>
180 <Data name="timestamp"><displayName>timestamp</displayName><value><![CDATA[NULL]]></value></Data>
181 <Data name="begin"><displayName>begin</displayName><value><![CDATA[NULL]]></value></Data>
182 <Data name="end"><displayName>end</displayName><value><![CDATA[NULL]]></value></Data>
183 <Data name="altitudeMode"><displayName>altitudeMode</displayName><value><![CDATA[NULL]]></value></Data>
184 <Data name="tessellate"><displayName>tessellate</displayName><value><![CDATA[~1]]></value></Data>
185 <Data name="extrude"><displayName>extrude</displayName><value><![CDATA[0]]></value></Data>
186 <Data name="visibility"><displayName>visibility</displayName><value><![CDATA[~1]]></value></Data>
187 <Data name="drawOrder"><displayName>drawOrder</displayName><value><![CDATA[NULL]]></value></Data>
188 <Data name="icon"><displayName>icon</displayName><value><![CDATA[NULL]]></value></Data>
189 <Data name="ID"><displayName>ID</displayName><value><![CDATA[4433]]></value></Data>
190 <Data name="GlideNumbe"><displayName>GlideNumbe</displayName><value><![CDATA[NULL]]></value></Data>
191 <Data name="Country"><displayName>Country</displayName><value><![CDATA[Germany]]></value></Data>
192 <Data name="OtherCount"><displayName>OtherCount</displayName><value><![CDATA[NULL]]></value></Data>
193 <Data name="long"><displayName>long</displayName><value><![CDATA[9.583274]]></value></Data>
194 <Data name="lat"><displayName>lat</displayName><value><![CDATA[54.705274]]></value></Data>
195 <Data name="Area"><displayName>Area</displayName><value><![CDATA[18991.8453946]]></value></Data>
196 <Data name="Began"><displayName>Began</displayName><value><![CDATA[2017/01/02]]></value></Data>
197 <Data name="Ended"><displayName>Ended</displayName><value><![CDATA[2017/01/05]]></value></Data>
198 <Data name="Validation"><displayName>Validation</displayName><value><![CDATA[News]]></value></Data>
199 <Data name="Dead"><displayName>Dead</displayName><value><![CDATA[0]]></value></Data>
200 <Data name="Displaced"><displayName>Displaced</displayName><value><![CDATA[0]]></value></Data>
201 <Data name="MainCause"><displayName>MainCause</displayName><value><![CDATA[Winter Storm Axel]]></value></Data>
202 <Data name="Severity"><displayName>Severity</displayName><value><![CDATA[1.5]]></value></Data>
203 </ExtendedData>
204 <styleUrl>#style9</styleUrl>
205 <MultiGeometry>
206 <Polygon>
207 <outerBoundaryIs>
208 <LinearRing>
209 <tessellate>1</tessellate>
210 <coordinates>
211 13.532608,54.306792,0.000000
212 13.33466,53.622268,0.000000
213 10.414927,53.415802,0.000000

```



Abbildung 81: Klassifiziertes KML-File (exportiert aus QGIS/visualisiert in Google Earth)

Als weitere Option kann die graphische Stildefinition eines Geoobjektes auch in einer externen KML-Datei abgespeichert werden. (siehe Abb. 82) Hierfür müssen der Testdatensatz im KML-Format und dessen KML-Referenzdatei im selben Ordner abgespeichert sein.

<pre> 23 <Folder><name>Flood_2017_KML</name> 24 <styleUrl>Flood_2017_KML_StyleUrl_Stylesheet.kml#transGreyPoly</styleUrl> 25 <Placemark> 26 <styleUrl>#transGreyPoly</styleUrl> 27 <ExtendedData><SchemaData schemaUrl="#Flood_2017_KML"> 28 <SimpleData name="ID">4433</SimpleData> 29 <SimpleData name="Country">Germany</SimpleData> 30 <SimpleData name="long">9.583274000000000</SimpleData> 31 <SimpleData name="lat">54.705274000000003</SimpleData> 32 <SimpleData name="Area">18991.845394600000873</SimpleData> 33 <SimpleData name="Began">2017/01/02</SimpleData> 34 <SimpleData name="Ended">2017/01/05</SimpleData> 35 <SimpleData name="Validation">News</SimpleData> 36 <SimpleData name="Dead">0</SimpleData> 37 <SimpleData name="Displaced">0</SimpleData> 38 <SimpleData name="MainCause">Winter Storm Axel</SimpleData> 39 <SimpleData name="Severity">1.500000000000000</SimpleData> 40 </SchemaData></ExtendedData> 41 <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13. 42 </Placemark> </pre>	<pre> 1 <?xml version="1.0" encoding="UTF-8"?> 2 <kml xmlns="http://www.opengis.net/kml/2.2" 3 xmlns:atom="http://www.w3.org/2005/Atom" 4 xmlns:gx="http://www.google.com/kml/ext/2.2" 5 xmlns:kml="http://www.opengis.net/kml/2.2"> 6 <Document> 7 <name>Flood style</name> 8 <Style id="transGreyPoly"> 9 <LineStyle> 10 <color>40000000</color> 11 <width>3</width> 12 </LineStyle> 13 <PolyStyle> 14 <color>ffe7dec3</color> 15 <fill>1</fill> 16 <outline>1</outline> 17 </PolyStyle> 18 </Style> 19 </Document> 20 </kml> </pre>
--	--

Abbildung 82: KML externe Stildefinition (eigene Darstellung)

In der KML-Hauptdatei des Testdatensatzes wird im Tag `<styleUrl>` über das Rautensymbol (#) auf die KML-Referenzdatei verwiesen, in welcher der Style definiert wurde. Zusätzlich wird ein Name für den Style vergeben.

Durch die `FadeIn` und `FadeOut` Option lassen sich die Objekte des Testdatensatz in einem definierten Zeitintervall einblenden. Als Beispiel werden 3 Sekunden für das Einblenden in einem fließenden Übergang gewählt. In diesem Zeitintervall lässt sich das Auf- und Ausblenden der Überflutungspolygone steuern. Dieser Fall wurde für ein Objekt des Testdatensatzes im KML-Dokument umgesetzt. (siehe Abb. 83 und 84)

```

10 <Style id="style1" <!-- Definierter Polygon-Stil -->
11   <LineStyle>
12     <color>00FFFFFF</color> <!-- Begin Füllfarbe: Transparent -->
13     <width>1</width> <!-- Begin Füllfarbe: Transparent -->
14   </LineStyle>
15   <PolyStyle>
16     <color>00FFFFFF</color> <!-- Begin Füllfarbe: Transparent -->
17   </PolyStyle>
18 </Style>
19
20
21 <gx:Tour>
22   <name>Start the fade-in of the object</name>
23   <gx:Playlist>
24     <gx:Wait> <gx:duration>1</gx:duration></gx:Wait>
25
26     <gx:AnimatedUpdate>
27       <gx:duration>3</gx:duration> <!-- Länge des Einblendens -->
28       <Update>
29         <targetHref></targetHref>
30         <Change>
31           <Style targetId="style1" <!-- Referenz Polygon-Stil -->
32             <LineStyle>
33               <color>64D2D2D2</color> <!-- Finale Füllfarbe: Transparentes Grau -->
34             </LineStyle>
35             <PolyStyle>
36               <color>64D2D2D2</color> <!-- Finale Füllfarbe: Transparentes Grau -->
37             </PolyStyle>
38           </Style>
39         </Change>
40       </Update>
41     </gx:AnimatedUpdate>
42     <gx:Wait> <gx:duration>3</gx:duration></gx:Wait>
43   </gx:Playlist>
44 </gx:Tour>

```

Abbildung 83: Definition zum Ein- und Ausblenden von KML-Objekten (Datengrundlage: [KML-o.J.])



Abbildung 84: FadeIn von KML-Objekt in Google Earth Pro (eigene Darstellung)

Zusätzlich kann auch über die Tags `<minFadeExtent>` und `<maxFadeExtent>` das Einblenden von Objekten in einem KML-Dokument gesteuert werden. Dieses Einblenden bezieht sich jedoch nicht auf eine Zeitspanne, sondern auf den Zoomfaktor in Google Earth Pro. (siehe Abb. 85)



Abbildung 85: FadeIn durch Zoom in Google Earth Pro (eigene Darstellung)

8.2.1.2 Animation und Interaktion – KML

Der Fragestellung und der Global Story nach zu urteilen, ist die TimeSpan Funktion jene, welche für diese praktische Analyse untersucht wird. Damit werden die Flutpolygone in Google Earth nach ihrem Auftreten animiert. Für Testzwecke wird der TimeSpan-Tag (<timeSpan>) für drei Objekte des Testdatensatzes umgesetzt.

Da die Zeitinformation selbst auf dem XML-Zeitformat aufbaut, muss jenes im Format **yyyy-mm-ddThh:mm:sszzzz** vorliegen. Jedoch kann die Zeitinformation auch nur als Datum definiert sein (vgl. [TIM-18]). Die Zeitinformation des Testdatensatzes liegt in dieser Form vor. Dies bedeutet, dass die die Spalten [Began] und [Ended] umformatiert werden müssen, da das Datumsformat dd/mm/yyyy ist. Dazu muss unter den *Computereinstellungen* → *Regions- und Spracheinstellungen* → *Datum und Zeit* → *Änderung von Datums- und Zeitformaten* das Datumsformat angepasst werden. (siehe Abb. 86) Nach dieser Änderung kann eine erneute Konvertierung des Testdatensatzes ins KML-Format im GIS erfolgen.

Formats

First day of week: Monday
 Short date: 2018-12-10
 Long date: 10 December, 2018
 Short time: 10:25 PM
 Long time: 10:25:02 PM
[Change date and time formats](#)

Abbildung 86: Zeit- und Datumseinstellungen Windows 10 (eigene Darstellung)



Abbildung 87: KML Objektanimation in Google Earth Pro (eigene Darstellung)

Die Animationszeit zwischen animierten Objekten wird automatisch im Hintergrund von Google Earth Pro interpoliert. Es kann nicht, wie beim OmniSuite Punktformat, ein gewünschtes Zeitintervall definiert werden.

KML-Objekte können, wenn sie als Placemark eindeutig definiert sind, auch interaktiv objektbasiert angesprochen werden. Dies gilt auch für definierte Folder, welche zum Beispiel Objekte einer Kategorie enthalten. So können die Objekte des Testdatensatzes zum Beispiel basierend auf dem Attribut „Severity“ (Stärke) eingefärbt werden.

Die Option Animationen und Interaktivität mit dem KML-Format umzusetzen ist somit gegeben.

8.2.1.3 Formatbibliotheken in C++ – libkml

Um KML-Daten in C++ durch KML-Parser einzulesen und zu visualisieren, steht die Open-Source-Programmbibliothek **libkml** zur Verfügung. Es handelt sich um eine Bibliothek zum Analysieren, Serialisieren und zum programmgesteuerten Zugriff auf ein KML-Dokumentobjektmodell (DOM). Die Bibliothek wurde für C++ umgesetzt und es gibt unter anderem Bindungen zu den Programmiersprachen Python und Java. Die Bibliothek unterliegt einer BSD-Lizenz (Berkeley Software Distribution). Dies bedeutet, dass die Software frei verwendet, kopiert, verbreitet und verändert werden darf. Bedingung ist, dass durch einen Copyright-Vermerk immer auf die ursprüngliche Programmbibliothek verwiesen wird. Die Programmbibliothek selbst wurde erfolgreich auf Betriebssystemen von Windows, Mac OS X 10.4 – 10.5 und Linux-Versionen getestet. (vgl. [INT-08])

libkml setzt sich aus den folgenden Elementen zusammen:

“The KML base library is internal to the libkml system. There is no public API to this module, but this must be linked with any use of libkml...The KML DOM is the API to KML data which is a direct representation of the XML. Everything you can do in XML you can do in the DOM API. KML DOM requires KML Base...The KML Engine is the API to KML algorithms such as semantics and behavior specified by the KML standard beyond the basic XML file format. KML Engine requires KML DOM and KML Base...The KML Convenience library consists of useful and common utility code not directly part of the KML standard. KML Convenience requires KML Engine, KML DOM and KML Base...The KML Regionator library consists of code useful for creating Region-based NetworkLink hierarchies. KML Regionator depends on KML Convenience, KML Engine, KML DOM, and KML Base.”([LIB-08])

Im folgenden C++ Code wird ein KML-Placemark geparkt und es werden die Werte des <name> Elements als Output zur Verfügung gestellt. Detaillierte Informationen und Dokumentationen zur Verwendung von libkml werden von Google (vgl. [LIB-08]) frei zur Verfügung gestellt. (siehe Abb. 88)

```

26 // parsekml.cc
27 // This program parses a KML Placemark from a memory buffer and prints
28 // the value of the <name> element on standard output.
29
30 #include <iostream>
31 #include <string>
32 #include "kml/dom.h" // The KML DOM header.
33
34 int main() {
35     // Parse KML from a memory buffer.
36     std::string errors;
37     kml::ElementPtr element = kml::Parse(
38         "<Placemark><name>hi</name></Placemark>",
39         &errors);
40
41     // Convert the type of the root element of the parse.
42     const kml::PlacemarkPtr placemark = kml::AsPlacemark(element);
43
44     // Access the value of the element.
45     std::cout << "The Placemark name is: " << placemark->get_name()
46         << std::endl;
47
48     return 0;
49 }
50

```

Abbildung 88: C++ KML Parsing Beispiel ([LIB-08])

8.2.2 SVG

Dem Ansatz des Rasterexports in ArcGIS Pro wird der Testdatensatz im Layout-View als *.svg exportiert. Dafür müssen die Layout Anforderungen aus Kapitel 6.1.2.2 eingehalten werden. Unter *Share* → *Layout* wird der Datensatz als *.svg Datei abgespeichert und konvertiert. Wie in Abbildung 89 zu sehen, werden keine geographischen Koordinaten verwendet.

```

1 <?xml version="1.0" encoding="utf-8" standalone="no"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
3 <svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" width="1161pt" height="581pt" viewBox="0 0 1161 581" xml:space="p">
4 <g color-interpolation="linearRGB" transform="scale(0.24,0.24)">
5 <g>
6 <rect x="0" y="0" width="4837.7856" height="2418.8928" fill="#FFFFFF" fill-opacity="1" />
7 </g>
8 <g id="Map Frame">
9 <g id="FloodEvents_2017_5">
10 <clipPath id="cp_2">
11 <path d="M4837.7856 2418.8928 L4837.7856 85.687785 L0 85.687785 L0 2418.8928 Z" />
12 </clipPath>
13 <g clip-path="url(#cp_2)">
14 <g transform="matrix(4.1666667 0 0 -4.1666667 2418.8928 1252.2903)">
15 <path d="M-186.66476 -100.55922 L-189.53765 -99.918076 L-189.21844 -97.999129 L-189.85686 -95.120143 L-189.21844 -88.403581 L-184.43029 -84.567765 L-180.59977 -84"
16 </g>
17 </g>
18 <g clip-path="url(#cp_2)">
19 <g transform="matrix(4.1666667 0 0 -4.1666667 2418.8928 1252.2903)">
20 <path d="M-183.47266 14.585754 L-196.24106 22.268606 L-197.19869 30.585231 L-192.41054 30.582825 L-185.70713 23.862573 L-183.47266 14.585754 Z" fill="#848582" f
21 </g>
22 </g>
23 <g clip-path="url(#cp_2)">
24 <g transform="matrix(4.1666667 0 0 -4.1666667 2418.8928 1252.2903)">
25 <path d="M-114.2041 -9.7577787 L-112.28884 -9.7587398 L-113.56568 -19.033798 L-119.63067 -22.549117 L-127.61092 -21.265701 L-134.31433 -16.144704 L-131.12223 -9.1
26 </g>
27 </g>
28 <g clip-path="url(#cp_2)">
29 <g transform="matrix(4.1666667 0 0 -4.1666667 2418.8928 1252.2903)">
30 <path d="M-227.84285 158.86112 L-235.8231 155.34677 L-245.3994 154.71188 L-252.10281 153.43584 L-251.46439 157.27374 L-235.50389 161.10394 L-227.84285 158.86112 Z
31 </g>

```

Abbildung 89: Ausschnitt aus der SVG-Datei des Testdatensatzes (eigene Darstellung)

Abbildung 89 der SVG-Datei zeigt, dass alle Pfade (Geoobjekte) des Testdatensatzes ident benannt sind. Dies macht eine Klassifikation oder ein Gruppieren der Daten im Zuge der kartographischen Stilisierung unmöglich. Zusätzlich wurden die Geometrien jedes Objektes

gespeichert, nicht aber deren Sachattribute. Dadurch wäre theoretisch ein Bezug pro Geoobjekt vorhanden, jedoch kann nicht damit gearbeitet werden. Es können Kategorisierungen, Klassifizierungen und Animationen ohne Objektattribute nicht umgesetzt werden.

Da es sich bei SVG um ein Vektorformat zur Darstellung von zweidimensionalen Grafiken im WWW und nicht um ein Datenformat zum Austausch von Geodaten handelt, wird es in der praktischen Analyse nicht weiter bearbeitet. Es handelt sich um kein geeignetes Format für eine Visualisierung am sphärischen Display.

8.2.3 GeoJSON

Der Testdatensatz wird in ArcGIS Pro unter *Toolboxes* → *Data Interoperability Tools* → *Quick Export* in das GeoJSON Format konvertiert. (siehe Abb. 90)

```
1 {
2   "type": "FeatureCollection",
3   "name": "Flood_2017_GEOJSON",
4   "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG:6300" } },
5   "features": [
6     { "type": "Feature", "properties": { "ID": 4433, "GlideHumb": null, "Country": "Germany", "OtherCount": null, "long": 9.593276, "lat": 54.705274, "Area": 18991.845394600001, "Began": "2017/01/02", "Ended": "2017/01/05", "Valida"},
7     { "type": "Feature", "properties": { "ID": 4434, "GlideHumb": null, "Country": "USA", "OtherCount": null, "long": -122.457484, "lat": 39.423418, "Area": 107407.297468, "Began": "2017/01/04", "Ended": "2017/01/08", "Valida"},
8     { "type": "Feature", "properties": { "ID": 4435, "GlideHumb": null, "Country": "Brazil", "OtherCount": null, "long": -55.55137, "lat": -23.409362, "Area": 386464.73744800001, "Began": "2017/01/08", "Ended": "2017/01/12"},
9     { "type": "Feature", "properties": { "ID": 4436, "GlideHumb": null, "Country": "Philippines", "OtherCount": null, "long": 125.026907, "lat": 8.085715, "Area": 46236.693155599999, "Began": "2017/01/16", "Ended": "2017/01/21"},
10    { "type": "Feature", "properties": { "ID": 4437, "GlideHumb": null, "Country": "Morocco", "OtherCount": null, "long": -8.367132, "lat": 18.580493, "Area": 13944.217784, "Began": "2017/01/07", "Ended": "2017/01/21", "Valida"},
11    { "type": "Feature", "properties": { "ID": 4438, "GlideHumb": null, "Country": "USA", "OtherCount": null, "long": -119.043052, "lat": 35.083904, "Area": 74443.110446300003, "Began": "2017/01/20", "Ended": "2017/01/21", "Valida"},
12    { "type": "Feature", "properties": { "ID": 4439, "GlideHumb": null, "Country": "Morocco", "OtherCount": null, "long": 34.515093, "lat": -20.777836, "Area": 128094.68577, "Began": "2017/01/01", "Ended": "2017/01/05"},
13    { "type": "Feature", "properties": { "ID": 4440, "GlideHumb": null, "Country": "Philippines", "OtherCount": null, "long": 124.0421, "lat": 7.580459, "Area": 30389.1114641, "Began": "2017/02/01", "Ended": "2017/02/05"},
14    { "type": "Feature", "properties": { "ID": 4441, "GlideHumb": null, "Country": "Peru", "OtherCount": null, "long": -77.172395, "lat": -8.250831, "Area": 288499.13140399999, "Began": "2017/02/01", "Ended": "2017/02/07", "Valida"},
15    { "type": "Feature", "properties": { "ID": 4442, "GlideHumb": null, "Country": "Colombia", "OtherCount": null, "long": -76.280289, "lat": 1.4911, "Area": 117321.460416, "Began": "2017/02/08", "Ended": "2017/02/09", "Valida"},
16    { "type": "Feature", "properties": { "ID": 4443, "GlideHumb": null, "Country": "Malaysia", "OtherCount": null, "long": 102.460736, "lat": 3.9317, "Area": 78809.750483500002, "Began": "2017/01/23", "Ended": "2017/02/07", "Valida"},
17    { "type": "Feature", "properties": { "ID": 4444, "GlideHumb": null, "Country": "USA", "OtherCount": null, "long": -120.824854, "lat": 38.403959, "Area": 124994.983267, "Began": "2017/03/14", "Ended": "2017/03/19", "Valida"},
18    { "type": "Feature", "properties": { "ID": 4445, "GlideHumb": null, "Country": "Chile", "OtherCount": null, "long": -70.246879, "lat": -30.939461, "Area": 189731.0257199999, "Began": "2017/02/14", "Ended": "2017/03/03", "Valida"},
19    { "type": "Feature", "properties": { "ID": 4446, "GlideHumb": null, "Country": "South Africa", "OtherCount": null, "long": 30.556134, "lat": -27.271723, "Area": 224179.61355100001, "Began": "2017/02/17", "Ended": "2017/03/03", "Valida"},
20    { "type": "Feature", "properties": { "ID": 4447, "GlideHumb": null, "Country": "Indonesia", "OtherCount": null, "long": 107.142002, "lat": -7.029304, "Area": 40233.827487900002, "Began": "2017/02/21", "Ended": "2017/03/03", "Valida"},
21    { "type": "Feature", "properties": { "ID": 4448, "GlideHumb": null, "Country": "Colombia", "OtherCount": null, "long": -75.479738, "lat": 3.722879, "Area": 135846.77802300001, "Began": "2017/02/23", "Ended": "2017/03/03", "Valida"},
22    { "type": "Feature", "properties": { "ID": 4449, "GlideHumb": null, "Country": "Zimbabwe", "OtherCount": null, "long": 29.022037, "lat": -18.642758, "Area": 163760.90567199999, "Began": "2017/03/14", "Ended": "2017/03/20"},
23    { "type": "Feature", "properties": { "ID": 4450, "GlideHumb": null, "Country": "Peru", "OtherCount": null, "long": -78.148087, "lat": -11.094239, "Area": 510942.3472499999, "Began": "2017/02/02", "Ended": "2017/02/07"},
24    { "type": "Feature", "properties": { "ID": 4451, "GlideHumb": null, "Country": "Australia", "OtherCount": null, "long": 150.740046, "lat": -31.486412, "Area": 180000.273911, "Began": "2017/03/13", "Ended": "2017/03/22", "Valida"},
25    { "type": "Feature", "properties": { "ID": 4452, "GlideHumb": null, "Country": "Burundi", "OtherCount": null, "long": 19.038572, "lat": -1.987297, "Area": 18863.134417400001, "Began": "2017/03/16", "Ended": "2017/03/22", "Valida"},
26    { "type": "Feature", "properties": { "ID": 4453, "GlideHumb": null, "Country": "Spain", "OtherCount": null, "long": -1.395236, "lat": 39.049941, "Area": 28021.282038700003, "Began": "2017/03/19", "Ended": "2017/03/22", "Valida"},
27    { "type": "Feature", "properties": { "ID": 4454, "GlideHumb": null, "Country": "USA", "OtherCount": null, "long": -116.123191, "lat": 45.543433, "Area": 75112.786445299993, "Began": "2017/03/18", "Ended": "2017/03/22", "Valida"},
28    { "type": "Feature", "properties": { "ID": 4455, "GlideHumb": null, "Country": "Mali", "OtherCount": null, "long": 17.046184, "lat": -20.549767, "Area": 249828.283281, "Began": "2017/03/09", "Ended": "2017/03/14"},
29    { "type": "Feature", "properties": { "ID": 4456, "GlideHumb": null, "Country": "Colombia", "OtherCount": null, "long": -76.113053, "lat": 2.187014, "Area": 24684.301610300001, "Began": "2017/04/01", "Ended": "2017/04/05"},
30    { "type": "Feature", "properties": { "ID": 4457, "GlideHumb": null, "Country": "Korea", "OtherCount": null, "long": 121.456325, "lat": -3.717519, "Area": 24002.5827839, "Began": "2017/03/21", "Ended": "2017/04/08", "Valida"},
31    { "type": "Feature", "properties": { "ID": 4458, "GlideHumb": null, "Country": "Australia", "OtherCount": null, "long": 145.481159, "lat": -21.974973, "Area": 258468.48799000001, "Began": "2017/03/23", "Valida"},
32    { "type": "Feature", "properties": { "ID": 4459, "GlideHumb": null, "Country": "Bangladesh", "OtherCount": null, "long": 90.622833, "lat": 24.317833, "Area": 14281.3045604, "Began": "2017/03/30", "Ended": "2017/04/18", "Valida"}
33 ]
34 }
```

Abbildung 90: Ausschnitt aus der GEOJSON-Datei des Testdatensatzes (eigene Darstellung)

Wie bereits in Kapitel 7.4.3 erwähnt, besitzt ein GeoJSON Objekt keinerlei Stilinformation, sondern beschreibt lediglich die in ihm definierten Geoobjekte mit ihren Attributen. In der Standard-Spezifikation von GeoJSON ist keine allgemein gültige Empfehlung zur Stilisierung definiert. Um GeoJSON Objekte kartographisch zu stylen und aufzubereiten, stehen mehrere Optionen zur Auswahl. Hierzu zählen Mapbox und CSS. Bei der Auswahl der Methode muss jedoch bezüglich der Anwendung und des Zielproduktes, einer Visualisierung am Hyperglobus, differenziert werden. Da Mapbox für Kartenprodukte im Web oder in mobilen Applikationen eingesetzt wird, ist der Anwendungsbereich für die Fragestellung dieser Forschungsarbeit fraglich.

Da es im Bereich von CSS und Mapbox Unterschiede in der Umsetzung und im Umfang an Möglichkeiten gibt, werden für die praktische Analyse beide Varianten untersucht und verglichen.

8.2.3.1 GeoJSON – CSS

8.2.3.1.1 Graphische Aufbereitung

Variante 1 beschreibt die Stilisierung von Geoobjekten des Testdatensatzes manuell nach dem Ansatz der *simplestyle* Spezifikation. Hierfür wird einem GeoJSON-Objekt des

Testdatensatzes in den Properties ein Stil zugewiesen. Dieser Schritt muss vom Nutzer manuell durchgeführt werden. (siehe Abbildung 91)

```

1  {
2    "type": "Feature",
3    "properties": {
4      "ID": 4433,
5      "GlideNumbe": null,
6      "Country": "Germany",
7      "OtherCount": null,
8      "long": 9.583276,
9      "lat": 54.705274,
10     "Area": 18991.845394600001,
11     "Began": "2017/01/02",
12     "Ended": "2017/01/05",
13     "Validation": "News",
14     "Dead": 0,
15     "Displaced": 0,
16     "MainCause": "Winter Storm Axel",
17     "Severity": 1.5,
18     "fill": "#DCCDCD",
19     "stroke-width": 1,
20     "fill-opacity": 0.6
21   },
22   "geometry": {
23     "type": "Polygon",
24     "coordinates": [[
25       "type": "MultiPolygon",
26
27     "coordinates":
28     [
29       [
30         [ 13.532608, 54.306792 ],
31         [ 13.33466, 53.662268 ],
32         [ 10.414927, 53.415802 ],
33         [ 9.425187, 54.457616 ],
34         [ 9.227239, 55.251098 ],
35         [ 9.524161, 55.994746 ],
36         [ 9.920057, 55.944961 ],
37         [ 9.623135, 54.953381 ],
38         [ 10.019031, 54.407731 ],
39         [ 10.761336, 54.109839 ],
40         [ 11.058258, 53.911344 ],
41         [ 12.493381, 54.009796 ],
42         [ 13.532608, 54.306792 ]
43       ]
44     ]
45   }
46 }

```

Abbildung 91: Kartographische Aufbereitung von GeoJSON per Geoobjekt (eigene Darstellung)

In der *simplestyle* Spezifikation ist festgelegt, dass Farben über das Format #RGB oder #RRGGBB interpretiert werden (HTML Farbcode). Andere Farbformate (wie rgba oder rgb) werden nicht unterstützt. (vgl. [SIM-17])

Wie bereits in Kapitel 7.4.3 beschrieben, können Geoobjekte der Testdaten auch in GeoJSON kategorisiert werden. Hierfür müssen in der Datei Feature Collections gebildet und mit einer Klasse ausgewiesen werden. Als Beispiel werden zwei Datensätze der Kategorie Torrential Rain gewählt. Auch dieser Schritt muss manuell durchgeführt werden. Auf die Stilisierung der Objekte wird, basierend auf der erstellten Klasse in der GeoJSON Datei, mittels einer CSS-Stildefinition zugegriffen. (siehe Abb. 92 und 93)

```

1  {
2    "type": "FeatureCollection",
3    "name": "Flood_2017_GEOJSON",
4    "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG:4030" } },
5    "className": {
6      "baseVal": "torrentialrain"
7    },
8    "features": [
9      { "type": "Feature",
10       "properties": {
11         "ID": 4433,
12         "GlideNumbe": null,
13         "Country": "Germany",
14         "OtherCount": null,
15         "long": 9.583276,
16         "lat": 54.705274,
17         "Area": 18991.845394600001,
18         "Began": "2017/01/02",
19         "Ended": "2017/01/05",
20         "Validation": "News",
21         "Dead": 0,
22         "Displaced": 0,
23         "MainCause": "Torrential Rain",
24         "Severity": 1.5
25       },
26       "geometry": {
27         "type": "MultiPolygon",
28         "coordinates": [ [ [
29           [ 13.532608, 54.306792 ],
30           [ 13.33466, 53.662268 ],
31           [ 10.414927, 53.415802 ],
32           [ 9.425187, 54.457616 ],
33           [ 9.227239, 55.251098 ],
34           [ 9.524161, 55.994746 ],
35           [ 9.920057, 55.944961 ],
36           [ 9.623135, 54.953381 ],
37           [ 10.019031, 54.407731 ],
38           [ 10.761336, 54.109839 ],
39           [ 11.058258, 53.911344 ],
40           [ 12.493381, 54.009796 ],
41           [ 13.532608, 54.306792 ]
42         ] ] ]
43       }
44     },
45     { "type": "Feature",
46       "properties": {
47         "ID": 4490,
48         "GlideNumbe": null,
49         "Country": "Nepal",
50         "OtherCount": null,
51         "long": 86.426961,
52         "lat": 27.294978,
53         "Area": 54260.541468299998,
54         "Began": "2017/06/30",
55         "Ended": "2017/07/03",
56         "Validation": "News",
57         "Dead": 11,
58         "Displaced": 6000,
59         "MainCause": "Torrential Rain",
60         "Severity": 1.0
61       },
62       "geometry": {
63         "type": "MultiPolygon",
64         "coordinates": [ [ [
65           [ 87.812587, 27.740706 ],
66           [ 88.307457, 27.641284 ],
67           [ 88.505405, 26.351937 ],
68           [ 87.416691, 26.352484 ],
69           [ 84.348497, 27.246582 ],
70           [ 84.942341, 28.238013 ],
71           [ 87.812587, 27.740706 ]
72         ] ] ]
73       }
74     }
75   ]
76 }

```

Abbildung 92: Gruppierung von GeoJSON Geoobjekten (eigene Darstellung)

```

1  <style type="text/css">
2
3  .torrentialrain{
4      stroke:#DCDCDC;
5      stroke-width:1;
6      fill:#DCDCDC;
7      opacity: 0.6;
8  }
```

Abbildung 93: CSS File zur Stilisierung der gruppierten Geoobjekte (eigene Darstellung)

8.2.3.1.2 Animation und Interaktion

Zum aktuellen Stand der Forschung wurden noch keine GeoJSON Objekte mit CSS animiert. Jedoch können die Animationszeiten („when“ beziehungsweise „begin“, „end“) in den Properties jedes Objektes definiert werden. Im Falle des Testdatensatzes sind dies die Attribute „Began“ und „Ended“. Diese Attribute bei eine zukünftige Animationsentwicklung eingesetzt werden.

Interaktivität ist in jenem Sinn gegeben, dass GeoJSON Testobjekte durch ihre FeatureID oder die jeweiligen Attribute angesprochen werden können.

8.2.3.1.3 Programmbibliotheken in C++ - RapidJSON

Hinsichtlich des Parsens von GeoJSON Objekten in C++ gibt es Formatbibliotheken für JSON, nicht jedoch direkt für GeoJSON. Nachdem GeoJSON auf JSON basiert, sind Formatbibliotheken auch für Objekte des speziellen Geodaten austauschformates anwendbar.

RapidJSON ist eine Formatbibliothek in C++ mit SAX (Simple API for XML) und DOM API, welche zum Parsen und Erstellen von JSON-Objekten verwendet wird. RapidJSON arbeitet schnell und die Performance der Bibliothek ist vergleichbar mit der C++ Funktion `strlen()`. Für Zwecke der Beschleunigung wird unter anderem SSE2³ unterstützt. Die Formatbibliothek ist eigenständig, dies bedeutet, dass sie nicht von externen Bibliotheken abhängig ist oder STL⁴ (C++ Standard Template Library) abhängig ist. Des Weiteren ist RapidJSON Unicode-freundlich. Es werden UTF-8, UTF-16, UTF-32 (LE & BE) und deren interne Erkennung, Validierung und Transkodierung unterstützt. So kann RapidJSON eine UTF-8-Datei lesen und die JSON-Zeichenfolgen in UTF-16 im DOM umcodieren. Zusätzlich werden auch Ersatzzeichen und „\u0000“ (Nullzeichen) unterstützt. Im folgenden Beispielcode (siehe Abb. 94) wird eine JSON-Zeichenfolge in ein DOM geparkt, eine Änderung im DOM vorgenommen und jenes DOM wieder als JSON String ausgegeben. (vgl. [RAP-o.J.])

³ Bei SSE2 handelt es sich um eine x86 Befehlserweiterung, welche von Intel im Zuge des Prozessors Pentium 4 eingeführt wurde. Dadurch werden Leistungen erhöht, wenn idente Operationen für mehrere Datenobjekte durchgeführt werden. Die Verarbeitung von Gleitkommazahlen wird durch SSE2 mit doppelter Genauigkeit (64-bit statt 32-bit) ermöglicht. (vgl. [INT-18])

⁴ Die Standard Template Library (STL) ist eine Gruppe von C++ - Template-Klassen, die allgemeine Programmierdatenstrukturen und -funktionen wie Listen, Stacks, Arrays usw. bereitstellen. Es handelt sich um eine Bibliothek von Containerklassen, Algorithmen und Iteratoren. (vgl. [THE-o.J.])

```

1 // rapidjson/example/simpledom/simpledom.cpp`
2 #include "rapidjson/document.h"
3 #include "rapidjson/writer.h"
4 #include "rapidjson/stringbuffer.h"
5 #include <iostream>
6 using namespace rapidjson;
7 int main() {
8     // 1. Parse a JSON string into DOM.
9     const char* json = "{\"project\":\"rapidjson\",\"stars\":10}";
10    Document d;
11    d.Parse(json);
12    // 2. Modify it by DOM.
13    Value& s = d["stars"];
14    s.SetInt(s.GetInt() + 1);
15    // 3. Stringify the DOM
16    StringBuffer buffer;
17    Writer<StringBuffer> writer(buffer);
18    d.Accept(writer);
19    // Output {"project":"rapidjson","stars":11}
20    std::cout << buffer.GetString() << std::endl;
21    return 0;
22 }

```

Abbildung 94: C++ RapidJSON Example ([RAP-o.J.])

8.2.3.2 GeoJSON - Mapbox

8.2.3.2.1 Graphische Aufbereitung

„A Mapbox style is a document that defines the visual appearance of a map: what data to draw, the order to draw it in, and how to style the data when drawing it. A style document is a [JSON](#) object with specific root level and nested properties.“ ([MAP-18a])

Die graphische Stilisierung von GeoJSON-Objekten wird in Mapbox auf zwei Arten definiert. Variante 1 beschreibt die graphische Aufbereitung mit CartoCSS in einer Desktopanwendung. Diese Methode wird in Mapbox Studio Classic eingesetzt. Durch die Stylesheet-Sprache wird das Erscheinungsbild von GeoJSON-Objekten definiert. Die jeweiligen Stile sind durch geschwungene Klammern {} begrenzt und enthalten Stileigenschaften wie auch Werte. In Mapbox Studio Classic wird jeder Layer als Selektor behandelt. Zusätzlich wird in Mapbox Studio Classic zwischen Style und Source unterschieden. Styles lassen sich nur für vordefinierte Stylelayer über CartoCSS definieren. Importiert man einen Datensatz (-> *Source*), zum Beispiel die Flutpolygone des Testdatensatzes im GeoJSON-Format, so kann nur zwischen der Sichtbarkeit und Nicht-Sichtbarkeit der Polygone unterschieden werden. CartoCSS Stilisierungen können darauf nicht zugreifen. (siehe Abb. 95)

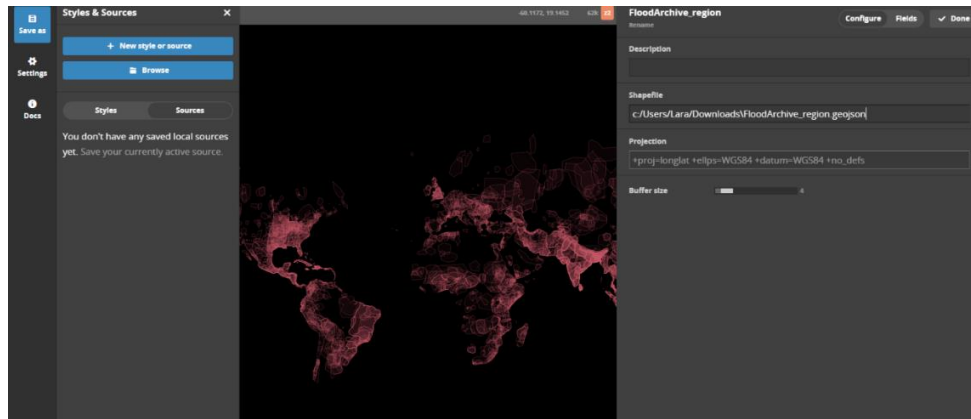


Abbildung 95: Flutpolygone in Mapbox Studio Classic (eigene Darstellung)

Um trotzdem ein Beispiel einer kartographischen Umsetzung durch CartoCSS aufzugreifen, bildet Abb. 96 die Stildefinition eines geöffneten Styles dar. Der kartographische Stil kann durch CartoSS im Fenster Style beliebig variiert werden.

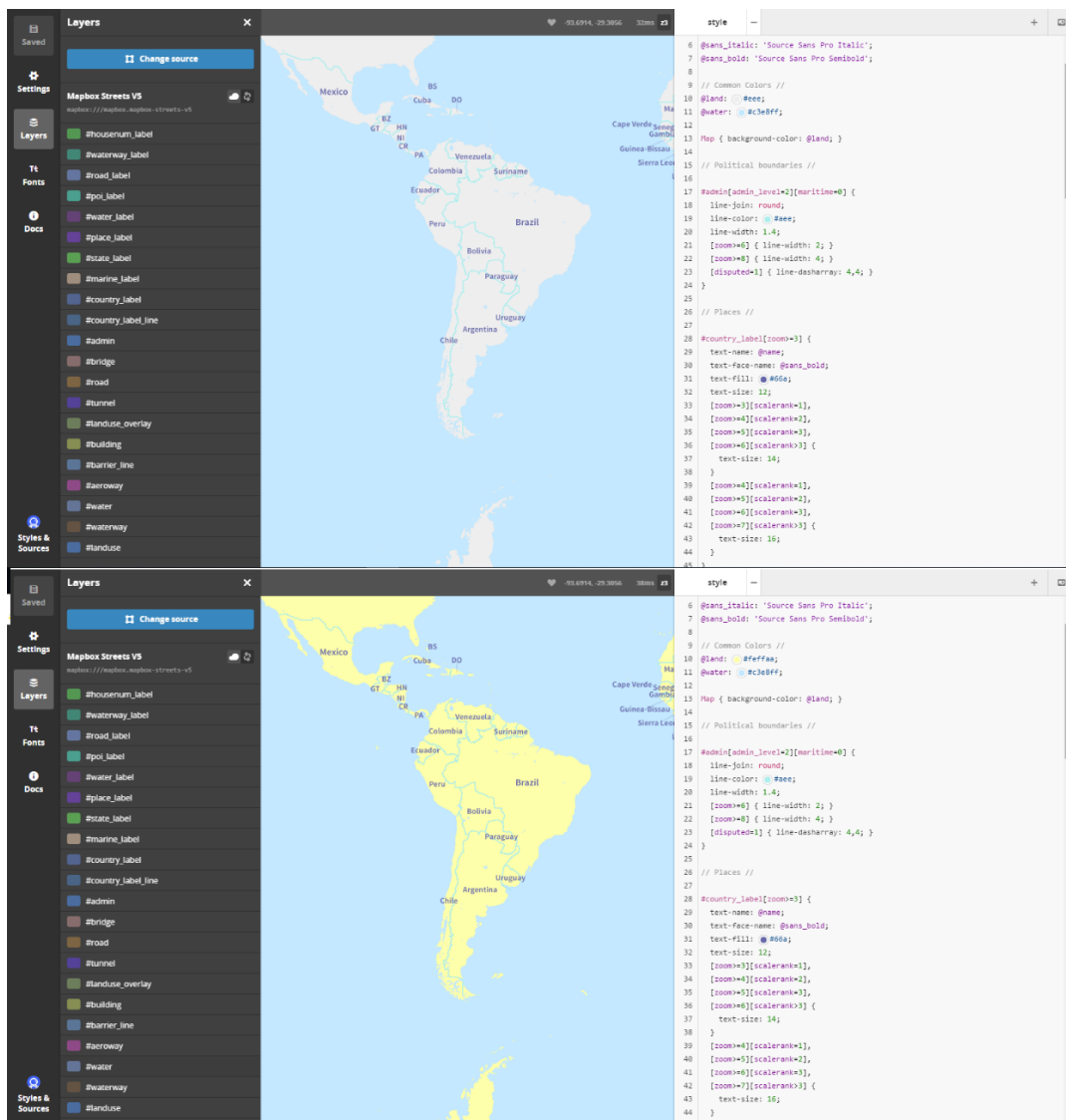


Abbildung 96: Variation von Geoobjekten in Mapbox Studio Classic mit CartoCSS (eigene Darstellung)

Da es sich bei Mapbox Studio Classic um eine Desktop-Anwendung handelt und zusätzlich um eine Mapbox Registrierung zur Benützung vorhanden sein muss, ist die Methode der Visualisierung von GeoJSON Objekten am sphärischen Display nicht geeignet.

Option 2 definiert die Stilisierung von Geoobjekten in der Mapbox GL JS Webmap. Die interaktive Online-Karte für das WWW wird in einem HTML-Dokument, basierend auf Mapbox GL JavaScript, umgesetzt. Hierfür werden zwei Polygone des Testdatensatzes im GeoJSON-Format verwendet und in eine Webmap integriert. Um auf das Mapbox Webmap Service zuzugreifen, sind die Registrierung auf der Website und die Erstellung eines ‘API Access Tokens’ notwendig. Dieser Token ist auch im Quellcode der Web-Map integriert. (siehe Abb. 97)

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset='utf-8' />
5    <title>Add GeoJSON Flood Polygons</title>
6    <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />
7    <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.js'></script>
8    <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.css' rel='stylesheet' />
9    <style>
10     body { margin:0; padding:0; }
11     #map { position:absolute; top:0; bottom:0; width:100%; }
12   </style>
13 </head>
14 <body>
15
16   <div id='map'></div>
17   <script>
18     mapboxgl.accessToken = 'pk.eyJJIjoic2xrMTg0IiwiaSI6ImNhcWlkZDRzMzBkZmc0OW55eDR4cTM4YXQifQ.zPkBQsqBwXVfMZLlsRq7AQ';
19     var map = new mapboxgl.Map({
20       container: 'map',
21       style: 'mapbox://styles/mapbox/streets-v9',
22       center: [7.5, 50.5],
23       zoom: 5
24     });
25
26     map.on('load', function () {
27
28       map.addLayer({
29         'id': 'maine',
30         'type': 'fill',
31         'source': {
32           'type': 'geojson',
33           'data': {
34             'type': 'FeatureCollection',
35             'name': "Flood_2017_GEOJSON",
36             "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::4030" } },
37             'features': [
38               {
39                 'type': 'Feature',
40                 "properties": {
41                   "ID": 4433,
42                   "GlideNumbe": null,
43                   "Country": "Germany"

```

Abbildung 97: Ausschnitt aus dem Quellcode der Mapbox WebMap (Datengrundlage: [ADD-o.J.])

Geoobjekte können in Mapbox nur im Quellcode intern kartographisch aufbereitet werden. Abbildung 98 zeigt die Definition des Styles für GeoJSON Flood Polygone. Diese befindet sich im JavaScript Abschnitt der HTML-Datei und basiert auf der Mapbox Style Specification, welche ein Teil der GL JS Bibliothek ist. Die kartographische Stilisierung ist im Schlüssel ‘paint’ angegeben: Zur Farbdefinition werden Hexadecimal, RGB oder RGBA-Farbcodes verwendet. Die Transparenz wird über Dezimalwerte von 0 – 1 definiert.

```

111   'paint': {
112     'fill-color': 'rgb(125,125,125)',
113     'fill-opacity': 0.8
114   }
115   });
116   });
117 </script>

```

Abbildung 98: Interne CSS-Stilisierung einer HTML Mapbox Datei (Datengrundlage: [ADD-o.J.])

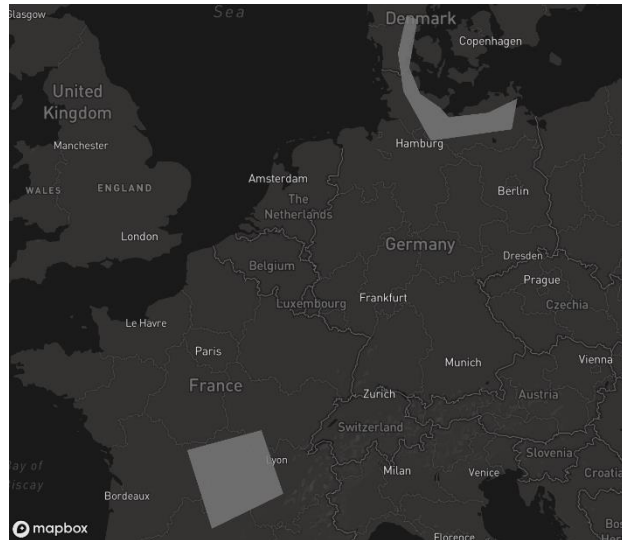


Abbildung 99: GeoJSON Objekte des Testdatensatzes in einer Mapbox Webmap (eigene Darstellung)

Die soeben diskutierte Variante beschreibt die Aufbereitung aller GeoJSON Objekte einer Webmap mit einem definierten Stil. Doch können Geoobjekte ebenso kategorisiert gestylt werden, durch Mapbox's Visualisierungsoption 'Date-Driven Styling'. Durch diese Option wird die Kartenvisualisierung um eine Dimension erweitert. Sowohl Größe und Farbe können dadurch attributbasiert gesteuert werden ([DAT-o.J.a]). Für Testzwecke werden Polygone basierend auf einem Attribut eingefärbt. Hierfür wird das Attribut „Severity“ mit den möglichen Werten 1.0, 1.5 oder 2.0 ausgewählt. (siehe Abb. 100) (Auf die attributbasierte Größenveränderung wird nicht näher eingegangen, da dies im Wesentlichen für Punktdatensätze wichtig ist. Hierfür gibt es bereits das entwickelte OmniSuite Punktfomat.)

```

198     'layout': {},
199     'paint': {
200       'fill-color': [
201         'interpolate',
202         ['linear'],
203         ['number', ['get', 'Severity']],
204         1.0, 'rgb(246, 249, 24)',
205         1.5, 'rgb(216, 0, 0)',
206         2.0, 'rgb(229, 0, 206)'
207       ],
208       'fill-opacity': 0.8
209     }
210   };
211 };
212 </script>

```

Abbildung 100: Data-Driven Styling in der Mapbox Webmap (Datengrundlage: [ADD-o.J.])

Data-Driven Styling arbeitet mit Interpolationen zwischen Paaren von Eingabe- und Ausgabewerten. Dadurch wird garantiert, dass die Interpolation zwischen Flood-Polygonen linear und glatt erfolgt. (vgl. [DAT-o.J.a]) Über **get** wird die Eigenschaft **severity** abgerufen. Da es sich bei den Werten um Zahlen handelt, werden über **number** Zahlenwerte erwartet. Zusätzlich findet eine lineare Interpolierung durch die definierten **RGB** (oder Hex) Farbwerte statt, um den jeweiligen Werten ihre Farbe zuzuweisen. Abbildung 101 zeigt die attributbasierte Einfärbung in der Mapbox Webmap.

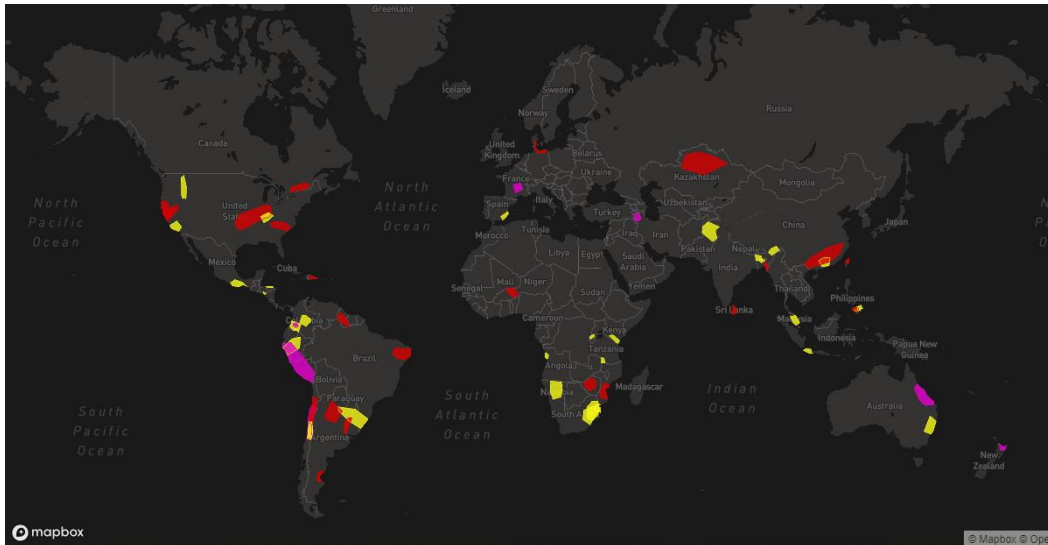


Abbildung 101: Attributbasierte Kategorisierung von Flood-Polygonen mit Mapbox (eigene Darstellung)

Auch die Definition eines FadeIn von Objekten in Mapbox GL JS ist möglich. Hierfür muss die **fill-opacity** auf 0 gesetzt werden. Zusätzlich wird ein Zeitraum für die **fill-opacity-transition** definiert. Nachdem die Webmap geladen wurde, wird eine Funktion aufgerufen, welche nach 3 Sekunden die definierte Transparenz (0.8) übernimmt. Der Übergang der Transparenzwerte ist fließend, somit ein FadeIn. (siehe Abb. 102 und Abb. 103)

```

199 'paint': {
200   'fill-color': [
201     'interpolate',
202     ['linear'],
203     ['number', ['get', 'Severity']],
204     1.0, 'rgb(246, 249, 24)',
205     1.5, 'rgb(216, 0, 0)',
206     2.0, 'rgb(229, 0, 206)'
207   ],
208   'fill-opacity': 0,
209   'fill-opacity-transition': {duration: 2000}
210 }
211 };
212 };
213
214 // so after the map is initialized we call a function which runs as example after 3
215 // the circle-opacity-transition is set as 2000, what means that changes to this va
216 setTimeout(function() {
217   map.setPaintProperty('geojson', 'fill-opacity', 0.8);
218   },
219   3000
220 );
221 </script>

```

Abbildung 102: Definition eines FadeIn von Flood-Objekten in Mapbox GL JS (eigene Darstellung)

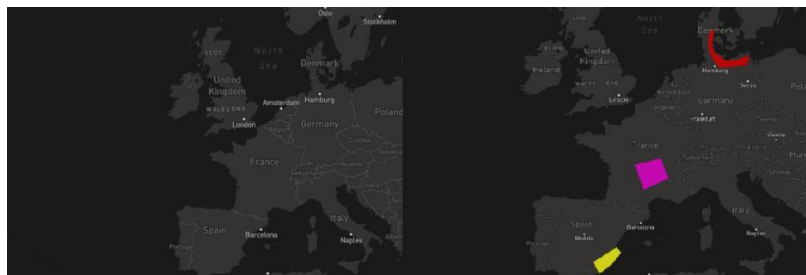


Abbildung 103: Sichtbarkeit der Flood-Polygone zur Sekunde 0 und 3 (eigene Darstellung)

Diese soeben beschriebenen internen Methoden zur kartographischen Aufbereitung von GeoJSON Objekten können nicht in einem externen CSS-Dokument definiert werden. Die Stilisierung kann nur innerhalb der Style-Spezifikation des JavaScript Abschnitts im HTML-Dokument vorgenommen werden.

8.2.3.2.2 Animation und Interaktion – Mapbox GL JS

Mapbox GL JS bietet die Möglichkeit, GeoJSON Objekte zu animieren. Animationen können entlang Ellipsen oder Pfaden stattfinden. Zusätzlich ist auch ein zeitbasiertes Auftreten von Objekten möglich. Diese Option ist jedoch beschränkt und nicht direkt für die Testdaten umzusetzen, da die gegebene Zeitinformation im Datumsformat angegeben ist. (“begin“/“end“). Zusätzlich sind zeitbasierte Animationen in Mapbox GL JS immer Slider basiert. Dies bedeutet, dass ohne die interaktive Betätigung des Sliders keine Änderung von Objekten über die Zeit sichtbar ist. (siehe Abb. 104) Dies macht die Animationsfunktion in Mapbox nur geringfügig geeignet für eine potentielle Visualisierung am Hyperglobus.

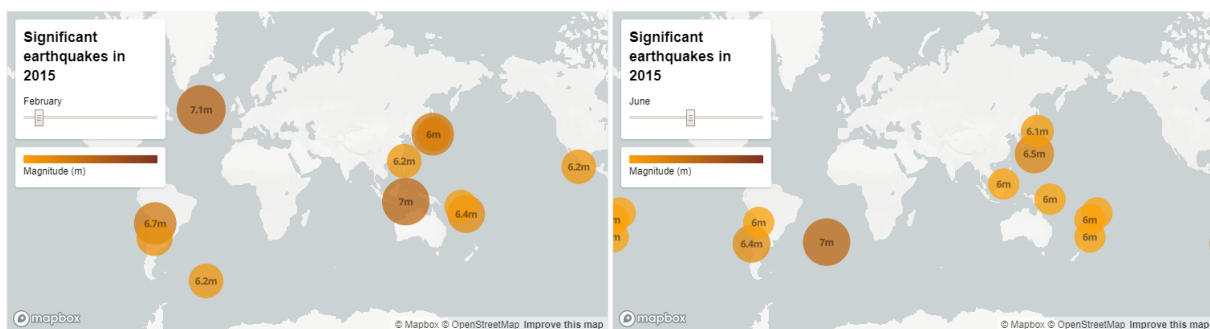


Abbildung 104: Slider-basierte Animation in Mapbox GL JS (eigene Darstellung)

Interaktivität ist in Mapbox GL JS dadurch gegeben, dass GeoJSON Objekte attributbasiert angesprochen werden können. (siehe Kapitel 7.4.3)

8.2.3.2.3 Programmbibliotheken in C++ - Mapbox GL JS

Zum derzeitigen Stand gibt es keine C++ Formatbibliothek, um Geoobjekte einer Mapbox GL JS Webmap zu parsen.

Tabelle 24 gibt einen Überblick über die Ergebnisse der praktischen Formatanalyse.

	Graphisches Erscheinungsbild			Animation		Interaktion		C++ Bibliothek		Speicherbedarf		
KML	intern		• <style> Tag • Shared styles -> <Style id=".."> • Graphische Klassifikation nach Export in QGIS → <styleUrl>	+	• TimeStamp: <when> Placemark • TimeSpan: <begin><end> Placemark → UTC Zeitformat • FadeIn/FadeOut: KML-Tour	+	• pro Objekt -> eindeutiger <Placemark> Tag	++	• libkml • Analyse und programmgesteuerter Zugriff auf KML-Objekte • frei verfügbar -> BSD Lizenz	+	69 KB	-
	extern		• Shared styles -> <Style id=".."> in externer KML-Datei auf identer Orderebene	+								
GeoJSON	CSS	intern	• Properties (fill/stroke-width/fill-opacity/...)	+	keine Dokumentation zu bisheriger Umsetzung	-	• pro Objekt -> z.B. ID oder Attribute • <i>pro Objekt -> manuell erstellte Klassen</i> in GeoJSON Datei	++	• RapidJSON • Parsen und Erstellen von GeoJSON Objekten • Unicode	+	36 KB	+
		extern	• manuelle Klassenerstellung in GeoJSON Datei • Stilisierung über externe CSS-Datei	+								
	Mapbox	intern	• Mapbox GL JavaScript • Automatische Klassifizierung der Objekte • FadeIn Effekt	++	• interaktive Animation durch TimeSlider	±	• pro Objekt -> z.B. ID oder Attribute	+	• keine Bibliothek zum Parsen von Mapbox GL JS Objekten in C++	-		
		extern	nicht umsetzbar.	-	nicht umsetzbar.	-	nicht umsetzbar.	-				
SVG	intern		• <style> Tag im SVG Dokument • Attribute der Geoobjekte werden nicht gespeichert -> keine objektbasierte Kategorisierung	--	nicht im Rahmen der Masterarbeit untersucht.							
	extern		• Stildefinition in externer CSS-Datei	+								

Tabelle 24: Praktische Formatanalyse der Vektorformate (eigene Darstellung)

9 Diskussion der Formatanalysen

Nachdem die drei gewählten Vektorformate in der praktischen Analyse analysiert wurden, gilt es die Ergebnisse der theoretischen und praktischen Formatanalyse zu diskutieren und eine **Empfehlung für den Hyperglobus** zu geben. Diese Punkte werden, basierend auf den in Kapitel 1.3 definierten Arbeitsfragen, diskutiert.

- *Welche Features sind für eine Visualisierung am sphärischen Display **essenziell**?*

In Kapitel 2 wurden die zentralen Elemente für eine Visualisierung am Hyperglobus definiert. Hierbei handelt es sich um die **graphische Aufbereitung, Globenwürdigkeit, Verzerrungsproblematik, Animation und Interaktion** von Geoobjekten. Es gilt, diese Punkte für eine Visualisierung am Hyperglobus, anhand der gegebenen Geodaten, zu überprüfen. Nur dadurch kann gesichert werden, **dass ein Vektorformat für eine Darstellung am sphärischen Display geeignet ist.**

- *Wie werden **kartographische** Stilinformationen im jeweiligen Format mitgespeichert?*

Die graphische Aufbereitung eines Shapefiles lässt sich extern in einem *.lyr (ArcGIS Pro) oder *.qrl (QGIS) Datei speichern. Der Quellcode der Layerdefinitionsdatei (*.lyr) ist jedoch wie das Shapefile selbst verschlüsselt. Die *.qrl Datei hingegen lässt sich in sämtlichen Editoren öffnen. Dadurch kann die graphische Aufbereitung von Geoobjekten außerhalb einer GIS-Software adaptiert werden.

Die graphische Aufbereitung der Objekte in KML kann intern sowohl pro Geoobjekt oder auch für Objektgruppen durchgeführt werden. Bei der graphischen Gestaltung pro Geoobjekt kann, durch eine am Anfang der Datei definierte StyleID, eine <styleUrl> vergeben werden. Dadurch muss der kartographische Stil nicht pro Objekt definiert bzw. wiederholt werden. Der Schritt der Objektkategorisierung in KML muss manuell vom Nutzer durchgeführt werden. Dies bringt, je nach Größe des Datensatzes, einen großen Aufwand mit sich. Als Alternative bietet sich ein Export der bereits kategorisierten Daten aus dem PlugIn MMQGIS in QGIS. In der KML-Datei des Exports sind die Geoobjekte nicht in Gruppen kategorisiert, jedoch wird jedem Objekt automatisch der in QGIS definierte Farbstil zugewiesen. Weiters lässt sich die kartographische Stilisierung der Objekte auch in einem externen KML-Dokument definieren. Das FadeIn von Objekten in KML kann nur im Zuge einer zeitlichen Animation und nicht objektbasiert eingesetzt werden.

Es gibt keine eindeutige Spezifikation für die graphische Aufbereitung des GeoJSON Formates. Diese ist abhängig vom Einsatzgebiet und der Anwendungsart. Vergleicht man die Optionen CSS und Mapbox, so bieten sich momentan mehr Visualisierungsmöglichkeiten in Mapbox GL JS. Geoobjekte einer GeoJSON-Datei können sowohl intern (Properties) als auch extern, durch die manuelle Bildung von Objektklassen, stilisiert und kategorisiert werden. Jedoch wird diese Form der Visualisierung nicht von Programmen wie QGIS oder ArcGIS Pro unterstützt und kann nicht überprüft werden. Die Aufbereitung der GeoJSON Objekte in Mapbox bietet hier

weitaus mehr Möglichkeiten. Objekte können einfärbig oder auch kategorisiert visualisiert werden. Dafür müssen, verglichen mit dem KML-Format, keine Objektgruppen gebildet werden. Durch die Definition der zu kategorisierenden Attribute wird automatisch auf die Objekteigenschaften zugegriffen. Dies reduziert den Zeitaufwand wesentlich. Bei der graphischen Gestaltung von GeoJSON Objekten in Mapbox kann das FadeIn durch eine JavaScript Funktion definiert werden.

Bei den Formaten GML und WKT handelte es sich um Formate zur Definition von Geoobjekten, weswegen eine graphische Aufbereitung des Erscheinungsbilds nicht umsetzbar ist. Aus diesem Grund wurden sie nicht weiter in der praktischen Analyse untersucht.

- *Welche **Animations- und Interaktionsmöglichkeiten** bieten sich im Rahmen der Vektorformate?*

Shapefiles lassen sich nur in einer GIS-Software animieren. Da der Quellcode eines Shapefiles zusätzlich verschlüsselt ist, können Animationen auch nicht in diesem ergänzt werden. Zusätzlich ist durch den verschlüsselten Quellcode auch keine Interaktion mit Geoobjekten eines Datensatzes möglich.

In KML werden Objekte durch die Angabe von <begin> und <end> im XML-Zeitformat automatisch animiert. Die Dauer der Animation kann jedoch nicht gesteuert werden. Diese Form der Animation ist leicht anzuwenden, da die Datumsangaben nicht in einem Kalkulationsprogramm in Sekunden umgerechnet werden müssen. Hingegen ist das zeitliche Abstimmen der Animation mit einem zweiten Datensatz (wie in Kapitel 8.1.3) nur schwer umsetzbar.

Bezüglich der Animation von GeoJSON Objekten mit CSS gibt es zum momentanen Stand noch keinerlei Dokumentation. Jedoch ist die Zeitinformation in der Datei in den Objekteigenschaften angegeben. Animationen können in Mapbox GL JS durchgeführt werden, jedoch ist deren Anwendbarkeit für das sphärische Display begrenzt, da Animationen interaktiv und Slider-basiert sind. Dies bedeutet, dass ohne das interaktive Benutzen keinerlei Animation ersichtlich ist.

Auch für das SVG-Format wurde die mögliche Animation von Objekten untersucht. Durch Elemente, welche im Zuge der SMIL definiert wurden, können Objekte animiert werden. (siehe Kapitel 7.4.4) Jedoch wurde das Vektorformat, aufgrund fehlender Attributdaten von Geoobjekten, nicht weiter in der praktischen Analyse beleuchtet.

Ein weiteres Format, welches untersucht wurde und Geodaten animieren kann, ist das OmniSuite Punktformat. Dieses kann, über seine implementierte Animationsfunktion Punkte und ihr Aufscheinen am sphärischen Display zeitlich steuern. Jedoch ist die direkte, synchrone Integration von Punktdaten bereits am sphärischen Display implementiert. Die bereits implementierte Animationsweise des OmniSuite Punktformates aus Kapitel 7.4.6 kann weder das KML- noch das GeoJSON-Format erfüllen

Für die Formate GML und WKT war zum Zeitpunkt der theoretischen Formatanalyse keine Animation von Geoobjekten bekannt.

Bis auf das Shapefile war der Quellcode von allen Vektorformaten in Editoren lesbar. Objekte der Vektorformate können, basierend auf einer eindeutigen ID oder Objektattributen, interaktiv angesprochen werden.

- *Werden die Vektorformate von **Programmen unterstützt** und gibt es (open-source) **Programmbibliotheken** in Programmiersprachen für diese?*

Die Formate SHP, GML, KML und GeoJSON können in kommerziellen und Open-Source GIS-Softwareprodukten ins jeweilige Format konvertiert werden. Das SVG-Format wird in GIS-Programmen soweit unterstützt, dass Geoobjekte im Kartenlayoutmodus ins SVG-Format exportiert werden können. Jedoch werden keine Geoattribute mitgespeichert. Das WKT-Format bildet immer nur den Code eines Geoobjektes ab und das OmniSuite Punktformat wird von keinem GIS-Programm unterstützt.

Die Untersuchung von Programmbibliotheken in C++ wurde für die Formatauswahl aus Kapitel 7 in der praktischen Analyse durchgeführt. Da das SVG-Format wegen seiner im Format fehlenden Geoattributdaten nicht näher evaluiert wurde, wurden geeignete Formatbibliotheken für das KML und GeoJSON Format gesucht.

Das KML- und GeoJSON-Format besitzen die Formatbibliotheken libkml und RapidJSON, welche das Parsen von Geoobjekten in C++ ermöglichen. Kritisch zu betrachten ist, dass es keine Bibliothek gibt, um GeoJSON Objekte einer Mapbox GL JS Webmap zu parsen. Somit ist der potentielle Zugriff von OmniSuite auf die benötigten Objektdaten nicht garantiert.

- *Welche Unterschiede gibt es im **Speicherbedarf** der jeweiligen Formate?*

Der Speicherbedarf der untersuchten Vektorformate hängt von deren Dateistruktur ab. Ein Großteil der untersuchten Formate basiert auf der XML-Struktur. Da diese tagbasiert ist, ist der Speicherbedarf verglichen mit dem Schlüssel:Wert Prinzip in der Dateistruktur von JSON größer.

- *Welche Vektorformate **eignen** sich und können diese Features **mitspeichern**?*

Sowohl das KML- als auch das GeoJSON-Format besitzen **Vor- und Nachteile, wodurch sie für eine Visualisierung am sphärischen Display nicht vollständig geeignet** sind.

Beide Formate können in GIS-Systemen konvertiert werden. Auch können graphische Stilinformationen in beiden Formaten extern abgespeichert werden. Dies bietet den Vorteil der Wiederverwendung von Stildefinitionen. Weitere Features aus Kapitel 2.2, **wie**

Globenwürdigkeit, Raumbezug, Animation und Interaktion können von beiden Formaten größtenteils abgedeckt werden.

Der Vorteil des KML-Formates ist, **dass Animationen von Geoobjekten ohne Interaktivität des Nutzers durchgeführt werden.** Hingegen kann der Zeitraum der Animation nicht beeinflusst werden. Ohne die Option MMQGIS zu berücksichtigen, ist auch die manuelle Kategorisierung von Objekten in KML zeitaufwendig und nicht als ideal für große Datensätze am Hyperglobus zu sehen. Durch die Automatisierung dieses Vorganges, basierend auf einem Skript, kann der Zeitaufwand dieses Schrittes reduziert werden. Auch ist **der Speicherbedarf einer KML-Datei, verglichen mit jenem einer GeoJSON-Datei, größer.**

Als wesentlicher Vorteil des GeoJSON-Formates ist sicherlich zu sehen, **dass dessen Einsatzbereich sehr vielfältig ist.** Dies zeigt sich allein dadurch, dass es keine definierte Stil-Spezifikation gibt und die kartographische Aufbereitung des Formates von der Anwendung abhängt. Diese Annahme wird durch die differenzierte Aufbereitung mit CSS oder in Mapbox unterstrichen. Abgesehen von der idealen graphischen Aufbereitung von GeoJSON Objekten in Mapbox GL JS ist die Eignung dieser Anwendungsform kritisch zu hinterfragen. Sowohl die **interaktive Animationsform via Slider und die nicht existente C++ Bibliothek zum Parsen der Objekte einer Mapbox GL JS Webmap,** stellen die Möglichkeit und den Mehrwert einer Implementation in OmniSuite in Frage. Zusätzlich darf nicht außer Acht gelassen werden, dass ein Zugriff auf Mapbox nur möglich ist, wenn ein aktiver AccessToken in der Datei eingebaut wurde und eine Registrierung besteht.

Um Geoobjekte mit ihrer graphischen Stilisierung am sphärischen Display abzubilden, muss OmniSuite auf die **Koordinaten des Geoobjektes, die graphische Stilisierung und die Animationszeit** zugreifen können. All diese Informationen können aus **den beiden analysierten Vektorformaten** bezogen werden.

Am Ende dieser Masterarbeit kann **leider keine eindeutige Empfehlung** für eines der untersuchten Vektorformate gegeben werden. Sowohl das KML- wie auch GeoJSON-Format besitzen Vor- und Nachteile bezüglich einer zukünftigen Visualisierung am sphärischen Display.

Die Forschungsfrage „*Welche Vektorformate eignen sich anhand ihrer Vor- und Nachteile für eine Visualisierung am sphärischen Display?*“ konnte somit **nicht** vollständig beantwortet werden. Der große Bereich an Vektorformaten konnte jedoch zumindest auf zwei Formate eingeschränkt werden. Eine vollständige Empfehlung zur Implementierung jener stellt diese Arbeit jedoch nicht dar.

Zusammenfassend wäre es empfehlenswert, **ein eigenes Vektorformat für den Hyperglobus zu entwickeln, welches Polygon- und Liniendatensätze auf diesem abbilden kann.** Dies wurde von Kristen J. auch schon im Zuge des OmniSuite Punktformats erfolgreich umgesetzt. Eine Möglichkeit hierfür wäre eine Konvertierung von einem herkömmlichen Vektorformat, wie KML oder GeoJSON, in ein spezielles, für den Hyperglobus entwickeltes Format.

Option 2 beschreibt eine Implementierung des GeoJSON Formates in der Globensoftware. Da jedoch noch keine Animationsfunktion im Format integriert ist, **müsste eine CSS Ergänzung für das FadeIn/FadeOut und die Animation von Geoobjekten** entwickelt werden. Nachdem

das Format jedoch anwendungsflexibel ist, ist eine Umsetzung dieser Funktionen als realistisch anzusehen.

Als dritte Alternative kann auch das KML-Format als ein potentieller Kandidat für eine Implementierung in OmniSuite gesehen werden. Jedoch ist die **Visualisierung von Animationen durch die automatische Interpolierung zwischen gegebenen Zeitpunkten zum momentanen Stand nicht zufriedenstellend**. Die korrekte Nutzung des Features Animation wäre somit, verglichen mit dem OmniSuite Punktformat, nicht gegeben.

Schlussendlich liegt es an Kristen J. und Riedl A. zu beurteilen, **ob KML, GeoJSON oder ein eigens erstelltes Format** am Besten für eine Implementierung am sphärischen Display geeignet ist. Es muss überprüft werden, ob die Entwicklung eines neuen Formates oder die Adaptierung der bestehenden Formate ein zufriedenstellenderes Resultat erzielen. Diese Forschungsarbeit hat durch ihre Formatanalyse eine Grundlage für diese Entscheidung gebildet.

10 Zusammenfassung

Im Rahmen dieser Masterarbeit wurde versucht Vektorformate festzulegen, welche für eine zukünftige Implementierung am sphärischen Display geeignet sind. Wesentliche Faktoren hierfür waren, ob ein Vektorformat graphisch aufbereitet werden kann, ob seine Geoobjekte animiert und interaktiv angesprochen werden können und ob Formatbibliotheken in C++ vorhanden sind, um das Format in der Globensoftware zu implementieren.

In einer ersten Untersuchung wurden die Formate auf ihre Geometrietypen, den Raumbezug, ihre potentielle graphische Stilisierung und ihre Dateistruktur untersucht. Auch wurde überprüft, in welchem Bereich die jeweiligen Vektorformate Einsatz finden und wie diese verbreitet sind. Basierend auf dieser theoretischen Evaluierung wurde eine Formatauswahl für die praktische Formatanalyse getroffen. So wurden unter anderem Vektorformate, welche nur die Geometrie von Geoobjekten abbilden und keine graphische Aufbereitung zulassen, von der praktischen Analyse ausgeschlossen.

Um die Bedeutsamkeit einer direkten, synchronen Visualisierung von Vektordaten zu unterstreichen, wurde mit einem Datensatz des Darmouth Flood Observatory eine Global Story nach dem aktuellen Ansatz umgesetzt. Während dem Arbeitsprozess wurde deutlich, dass die Bearbeitung großer Datensätze zur Darstellung eines Zeitverlaufs, nach dem momentanen Ansatz der indirekten Rastervisualisierung, einen nahezu nicht zu bewältigenden zeitlichen Aufwand darstellt. Zusätzlich verhindert der fehlende Objektbezug den Einsatz von Interaktivität.

In der praktischen Analyse wurden die gewählten Vektorformate KML, SVG und GeoJSON hinsichtlich der genannten Forschungsfragen untersucht. Diese Evaluierung der Formate hat große Unterschiede aufgedeckt. So kann das SVG-Format nicht eingesetzt werden, da es sich um ein graphisches Vektorformat handelt und keine Geoattribute abgespeichert werden. Dadurch wurde der Fokus auf das KML- und GeoJSON-Format gelegt. Es hat sich herausgestellt, dass keines der beiden Vektorformate die Voraussetzungen für eine Visualisierung am Hyperglobus vollständig erfüllt. Die graphische Stilisierung von Geoobjekten kann in beiden Formaten sowohl intern als auch extern durchgeführt werden. Hingegen sind die Ergebnisse der Objektanimation nicht ideal, da Animationen im KML-Format automatisch zwischen gegebenen Daten interpoliert werden und es vom GeoJSON keinerlei Dokumentation zu einer bisherigen Umsetzung gibt. Formatbibliotheken in C++ sind für beide Formate existent. In Fragen des Speicherbedarfs ist jener des GeoJSON-Formats geringer.

Zusammenfassend konnte, basierend auf den genannten Faktoren, keine direkte Empfehlung für die Implementierung eines der analysierten Vektorformate gegeben werden. Die Formate GeoJSON und KML könnten aufgrund der Basisfeatures implementiert werden, jedoch sind die Animationsfunktionen beider Formate als nicht ideal anzusehen. Basierend darauf konnte keine vollständige Formatempfehlung gegeben werden.

Alternativ ist als Resultat dieser Masterarbeit die Entwicklung eines weiteren Vektorformates für das sphärische Display anzudenken. Dieses könnte sich auf die Abbildung von Linien- und Polygondaten konzentrieren. Durch die Entwicklung eines eigenen Formates kann dieses spezifisch an die Anforderungen bezüglich Geometrie, Raumbezug, graphischer Aufbereitung, Animation und Interaktion angepasst werden.

Es wird sich zeigen, ob es in naher Zukunft noch Entwicklungen im Bereich der Geoobjektanimation mit den Formaten KML oder GeoJSON geben wird. Wenn nicht, kann durch die eigene Vektorformatentwicklung der HRG eine vielversprechende Lösung für die direkte, synchrone Visualisierung von Polygon- und Vektordaten am sphärischen Display geschaffen werden.

Wie bereits bei Punktobjekten, wird bei der Implementierung des zukünftigen Vektorformates auch die zunehmende Verzerrung von Linien in Polnähe berücksichtigt werden müssen.

Abgesehen von dieser Problematik, werden sich durch die direkte Integration der Vektordaten am Hyperglobus viele Vorteile, bezogen auf den Zeitaufwand und die objektbasierte Animation wie auch Interaktion von Polygon- und Liniendaten, ergeben. Dadurch werden sich am Globenmedium der Zukunft neue Möglichkeiten in der Umsetzung von Global Stories bieten.

11 Literaturverzeichnis

- [ADD-o.J.] Add a GeoJSON polygon. Mapbox. Online unter: <https://docs.mapbox.com/mapbox-gl-js/example/geojson-polygon/>, (Stand: 20.01.2019)
- [AGU-14] A Guide to SVG Animations (SMIL). CSS-Tricks. 2014. Online unter: <https://css-tricks.com/guide-svg-animations-smil/>, (Stand: 18.01.2019)
- [BAU-08] BAUMANN, P.: Konzeptuelle Modellierung von Geodiensten. In: Informatik-Spektrum, Volume 31. Springer Verlag Heidelberg. 2008. S. 435-450.
- [BAR-95] BARTELME, N.: Geoinformation. Modelle Strukturen Funktionen. Springer Verlag Heidelberg Berlin New York. 1995. 414 S.
- [BIL-17] Bildformate für die Verwendung in OmniSuite. LV Multimediatechnologie und Geokommunikation. 2017. 11 S.
- [BRO-13] BROWN P., STEWART I., HAEML N., POOLEY A., Rasmus A., Shah M.: EXT_texture_compression_s3tc. 2013. Online unter: https://www.khronos.org/registry/OpenGL/extensions/EXT/EXT_texture_compression_s3tc.txt, (Stand: 10.12.2018)
- [BUR-06] BURGGRAF, D. S.: Geography Markup Language. In: Data Science Journal. Volume 5. 2006. S. 178-203.
- [CAR-o.J.] CARTOGRAPHIC STYLESHEET. User Guide. Instructions for using Ordnance Survey's cartographic stylesheets. Online unter: <https://www.ordnancesurvey.co.uk/docs/user-guides/cartographic-stylesheet-user-guide.pdf>, (Stand: 20.11.2018)
- [COZ-11] COZZI, P., RING, K.: 3D Engine Design for Virtual Globes, CRC Press, Taylor & Francis Group, Boca Raton. 499 S.
- [CRI-14] CRICKARD, P.: Leaflet.js Essentials – Packt Publishing. Birmingham. 2014. 165 S.
- [DAS-o.J.] Das SVG-Dateiformat. Inkscape. Online unter: <https://inkscape.org/de/entwickeln/das-svg-format/>, (Stand: 05.08.2018)
- [DAT-18] Data Catalog. THE WORLD BANK. Online unter: [https://datacatalog.worldbank.org/search?search_api_views_fulltext_op=AND&query=world&sort_by=search_api_relevance&sort_order=DESC&page=0%2C0&f\[0\]=field_wbddd_data_type%3A295](https://datacatalog.worldbank.org/search?search_api_views_fulltext_op=AND&query=world&sort_by=search_api_relevance&sort_order=DESC&page=0%2C0&f[0]=field_wbddd_data_type%3A295) (Stand: 20.11.2018)
- [DAT-o.J.] data.gv.at – offene Daten Österreichs. Online unter: <https://www.data.gv.at/suche/> (Stand: 25.07.2018)
- [DAT-o.J.a] data-driven styling. Mapbox. Online unter: <https://www.mapbox.com/help/define-data-driven-styling/>, (Stand: 05.01.2019)
- [DGF-12] DGfK – Sektion für Bayern: Quo Vadis Globus? – Die digitale Wiedergeburt des Globus. 2012. Online unter: <http://kartographie.com/quo-vadis-globus-die-digitale-wiedergeburt-des-globus/>, (Stand: 14.05.2018)

- [DIG-08] JENKINS, N.: Earth Browser 3 Preview. 2008. Online unter: <http://www.digitalgeography.co.uk/archives/category/earth-browser/>, (Stand: 02.05.2018)
- [DOR-96] DORFFNER, L.: Der digitale Behaim-Globus – Visualisierung und Vermessung des historisch wertvollen Originals. In: FELDMANN, H., HAMMER, M.C., CAVELTI, A., DÜRST, A., KLÖTI, T., OEHLI, M.: Cartographica Helvetica. Murten, Bern. CARTOGRAPHICA HELVETICA. 1996. S. 20-24.
- [EAR-17] Earth at night: Flat Maps. Online unter: <https://earthobservatory.nasa.gov/features/NightLights/page3.php>, (Stand: 04.12.2018)
- [ESR-98] ESRI Shapefile Technical Description. ESRI. 1998. Online unter: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>; (Stand: 04.08.2018)
- [ESR-o.J.] ESRI PRJ File. Online unter: https://vsp.pnnl.gov/help/index.htm#vsample/ESRI_PRJ_File.htm, (Stand: 05.08.2018)
- [EVJ-07] EVJEN, B., KENTT, S., THANGARATHINAM, T., KAY, M., VERNET, A., ERFUSON, S.: Professional XML. Indianapolis. 2007. 856 S.
- [FIL-05] File format. Techtarget. 2005. Online unter: <https://whatis.techtarget.com/definition/file-format>, (Stand: 15.01.2019)
- [GDA-17] GDAL. FAQ-General. 2017. Online unter: <https://trac.osgeo.org/gdal/wiki/FAQGeneral#WhatdoesOGRstandfor>, (Stand: 25.07.2018)
- [GEO-14] Geojson CSS. Openstreetmap Wiki. 2014. Online unter: https://wiki.openstreetmap.org/wiki/Geojson_CSS, (Stand: 20.10.2018)
- [GEO-18] Geography Markup Language. 2018. Online unter: <http://www.opengeospatial.org/standards/gml> (Stand: 26.07.2018)
- [GEO-18a] Geoprocessing considerations for shapefile output. ArcGIS. 2018. Online unter: <http://desktop.arcgis.com/en/arcmap/latest/manage-data/shapefiles/geoprocessing-considerations-for-shapefile-output.htm> , (Stand: 05.08.2018)
- [GEO-o.J.] GeoJSON. Online unter: www.geojson.org, (Stand: 17.12.2018)
- [GEO-o.J.a] GeoJSON. Mapbox. Online unter: <https://docs.mapbox.com/help/glossary/geojson/>, (Stand: 10.01.2019)
- [GLO-18a] GLOBOCCESS: OmniSuite 5.0.0. 2018. 130 S.
- [GLO-18b] Global Active Archive of Large Flood Events. Dartmouth Flood Observatory. 2018. Online unter: <http://floodobservatory.colorado.edu/Archives/index.html>, (Stand: 24.11.2018)
- [HAK-02] HAKE, G., GRÜNREICH, D.: Kartographie. 7. Auflage. De Gruyter. Berlin. 1994. 600 S.
- [HAR-09] HARCKE, R.: Geoinformationssysteme – Geodaten. 2009. Online unter: <http://romanharcke.de/geoinformationssysteme-geodaten-kapitel-4/>, (Stand: 21.06.2018)
- [HOW-17] How To: Create a time animation from a shapefile. ESRI. 2017. Online unter: <https://support.esri.com/en/technical-article/000012836>, (Stand: 16.01.2019)
- [HOW-17a] How CSS works. Mozilla Developer. 2017. Online unter: https://developer.mozilla.org/de/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works, (Stand: 20.01.2019)
- [HTM-18] HTML5 SVG. 2018. Online unter: https://www.w3schools.com/hTml/html5_svg.asp/ (Stand: 06.08.2018)

- [INT-03] Introduction to JPEG. Comp.compression Frequently Asked Questions. 2003. Online unter: http://www.quaretec.com/u/vilo/edu/200203/Tekstialgoritmid_I/Loengud/Loeng7_Compression/www.faqs.org/faqs/compression-faq/part2/section-6.html (Stand: 19.01.2019)
- [INT-08] Introducing libkml: a library for reading, writing, and manipulating KML. Google Open Source Blog. 2008. Online unter: <https://opensource.googleblog.com/2008/03/introducing-libkml-library-for-reading.html> (Stand: 12.12.2018)
- [INT-16] Interaktives Auswählen von Features. ArcGIS for Desktop. 2016. Online unter: <http://desktop.arcgis.com/de/arcmap/10.3/map/working-with-layers/selecting-features-interactively.htm>, (Stand: 20.01.2019)
- [INT-18] Intel Streaming SIMD Extensions Technology. Intel. 2018. Online unter: <https://www.intel.com/content/www/us/en/support/articles/000005779/processors.html> (Stand: 01.01.2019)
- [INT-o.J.] Introducing JSON. Online unter: <https://www.json.org/>, (Stand: 07.08.2018)
- [JOB-07] JOBST, M., GERMANCHIS, T.: The Employment of 3D in Cartography – An Overview. In: Cartwright W., Peterson M., Gartner G. (Herausgeber): Multimedia Cartography. 2nd Edition. Springer. Heidelberg. S. 217-228.
- [KIE-18] KIESLIN, M., KORDASCH, S.L., STEPANEK, K.: Tiefseekabel – Die Welt hängt am Draht. Global Story. LV Multimediatechnologie und Geokommunikation. 2018.
- [KML-12] KML Color. Zonum Solutions. 2012. Online unter: http://www.zonums.com/gmaps/kml_color/, (Stand: 10.12.2018)
- [KML-17] KML. Google. 2017. Online unter: <https://developers.google.com/kml/faq>, (Stand: 30.07.2018)
- [KML-19] KML. OGC. 2019. Online unter: <http://www.opengeospatial.org/standards/kml> (Stand: 22.01.2019)
- [KML-o.J.] KML Reference. Google. Online unter: <https://developers.google.com/kml/documentation/kmlreference#placemark>, (Stand: 02.08.2018)
- [KML-o.J.a] KML – Keyhole Markup Language. GDAL. Online unter: https://www.gdal.org/drv_kml.html, (Stand: 20.01.2019)
- [KRI-12] KRISTEN, J.: Entwicklung von interaktiven kartographischen 3D-Applikationen am Beispiel eines taktilen Hyperglobus. Diplomarbeit. Universität Wien. 2012. 114 S.
- [KRI-14] KRISTEN, J.: Earthquakes 1914 – 2014. Global Story. 2014.
- [KRI-18] KRISTEN, J.: 6m Sea Level Rise. Global Story. 2018.
- [LAK-04] LAKE, R., BURGGRAF, D. TRNINIC, M., RAE, L.: Geography Mark-Up Language: Foundation of the Geo-Web. John Wiley & Sons. Ltd. England. 2004. 408 S.
- [LAK-05] LAKE, R.: The application of geography markup language (GML) to the geological sciences. In: Computers & Geosciences, Volume 31. 2005. S. 1081-1094.
- [LAN-13] DE LANGE, Norbert: Geoinformatik in Theorie und Praxis. 3. Auflage. Berlin Heidelberg, Springer Spektrum. 2013. 454 S.
- [LIB-08] libkml. Google Code Archive. 2008. Online unter: <https://code.google.com/archive/p/libkml/downloads>, (Stand: 12.12.2018)
- [LIE-12] LIEM, J.: Vektordaten am taktilen Hyperglobus – über Verfahren zur Aufbereitung, Visualisierung geographischer Vektordatensätze am sphärischen Display. Diplomarbeit. Universität Wien. 2012. 109 S.

- [MAP-18] Mapbox. OpenStreetMap Wiki. 2018. Online unter: <https://wiki.openstreetmap.org/wiki/Mapbox> (Stand: 11.12.2018)
- [MAP-18a] Mapbox Style Specification. Mapbox. 2018. Online unter: <https://www.mapbox.com/mapbox-gl-js/style-spec/>, (Stand: 03.01.2019)
- [MAP-18b] Mapbox GL JS. Mapbox. 2018. Online unter: <https://www.mapbox.com/mapbox-gl-js/api/>, (Stand: 04.01.2018)
- [NAT-18] Natural Earth I (1:10m). Natural Earth. Free vector and raster map data at 1:10m, 1:50m, and 1:110m scales. 2018. Online unter: <https://www.naturalearthdata.com/downloads/10m-raster-data/10m-natural-earth-1/>, (Stand: 14.01.2019)
- [OGC-07] OGC, WILSON, T. (Ed.): KML 2.2 – An OGC Best Practice. 2007. 153 S. Online unter: http://portal.opengeospatial.org/files/?artifact_id=23689&passcode=zvufhhkyzat8xbgbkzv, (Stand: 30.07.2018)
- [OGC-08] OGC Approves KML as Open Standard. OGC. 2008. Online unter: <https://www.opengeospatial.org/pressroom/pressreleases/857>, (Stand: 18.12.2018)
- [OGC-11] OGC, HERRING, J. (Hrsg.): OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture (Version 1.2.1). 2010. 93 S. Online unter: http://portal.opengeospatial.org/files/?artifact_id=25355 (Stand: 20.07.2018)
- [OGC-12] OGC® Geography Markup Language (GML) — Extended schemas and encoding rules. OGC. 2012. 91 Seiten. Online unter https://portal.opengeospatial.org/files/?artifact_id=46568, (Stand: 26.07.2018)
- [OGC-15] OGC, BURGGRAU, D. (Hrsg.): OGC KML 2.3. 2015. Online unter: <http://docs.opengeospatial.org/is/12-007r2/12-007r2.html>, (Stand: 30.07.2018)
- [OGC-15a] OGC - Geographic information: Well-Known text representation of coordinate reference systems. OGC. 2015. Online unter: <http://docs.opengeospatial.org/is/12-063r5/12-063r5.html#8>, (Stand: 15.07.2018)
- [OGR-o.J.] OGR Vector Formats. GDAL. Online unter: http://www.gdal.org/ogr_formats.html, (Stand: 25.07.2018)
- [OPT-16] Optimale Darstellungen auf einem sphärischen Display. Lehrveranstaltung Multimediatechnologie und Geokommunikation. 2016. 7 S.
- [PUN-17] Punktsignaturen in Photoshop einfügen und zur Darstellung am Globus entzerren. Tutorial. LV Multimediatechnologie und Geokommunikation. 2017. 4 S.
- [RAP-o.J.] RapidJSON. RapidJSON Documentation. Online unter: <http://rapidjson.org/index.html>, (Stand: 01.01.2019)
- [RIE-99] RIEDL, A.: Digital Globes. In: CARTWRIGHT, William, PETERSON, Michael P., GARTNER, Georg (Ed.): Multimedia cartography. Springer. Berlin, New York. 1999. S. 11 – 30.
- [RIE-00] RIEDL, A.: Virtuelle Globen in der Geovisualisierung. Mit einem Nachwort Globenforschung in Österreich von I. Kretschmer. Wien, Institut für Geographie und Regionalforschung der Universität Wien. Hrsg. von I. Kretschmer und K. Kriz Wien, Institut für Geographie und Regionalforschung der Universität Wien. 2000 (=Wiener Schriften zur Geographie und Kartographie, Band 13) 158 S.
- [RIE-11] RIEDL, A.: Entwicklungsgeschichte digitaler Globen. In: WAWRIK, Franz: Der Globusfreund 57/58. Wissenschaftliche Zeitung für Globenkunde. Internationale Coronelli-Gesellschaft für Globenkunde. Wien. 2011. S. 153-163.

- [RIE-12] RIEDL, A.: Skriptum zur Einführung in die Geoinformation (EGI), Universität Wien. 2012.
- [SCA-18] Scalable Vector Graphics (SVG) 2. W3C Candidate Recommendation 4th October 2018. W3C. 2018. Online unter: <https://www.w3.org/TR/SVG2/>, (Stand: 06.08.2016)
- [SCH-17] SCHUPPAR, B.: Geometrie auf der Kugel – Alltägliche Phänomene rund um Erde und Himmel. Springer Verlag. Berlin, Heidelberg. 256 S.
- [SIM-17] simplestyle-spec. Github. 2017. Online unter: <https://github.com/mapbox/simplestyle-spec/tree/master/1.1.0>, (Stand: 11.12.2018)
- [SMU-17] SMUTNY, T.: Optimierungsvorschläge zur Real Time Visualisierung von Punktsignaturen gezeigt am Beispiel von Luftfahrzeugdaten. Masterarbeit. Universität Wien. 2017. 123 S.
- [SRI-13] SRIPARASA, S. S.: JavaScript and JSON essentials– Packt Publishing. Birmingham. 2013. 120 S.
- [SVG-18] SVG Gradients. An ellipse with a vertical linear gradient from yellow to red. W3Schools. 2018. Online unter: https://www.w3schools.com/graphics/tryit.asp?filename=trysvg_linear, (Stand: 06.08.2018)
- [THE-o.J.] The C++ Standard Template Library (STL). Online unter: <https://www.geeksforgeeks.org/the-c-standard-template-library-stl/>, (Stand: 01.01.2019)
- [TIM-18] Time and Animation. Keyhole Markup Language. 2018. Online unter: <https://developers.google.com/kml/documentation/time>, (Stand: 10.12.2018)
- [WAS-16] Was sind Raster-Daten? ArcMap. 2016. Online unter: <http://desktop.arcgis.com/de/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm/>, (Stand: 16.07.2018)
- [WEB-03] WEBER, C.: XML-Grundlagen. 2003. Online unter: <http://www.lgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-kl.de/courses/seminar/WS0203/ausarbeitung1.pdf>, (Stand: 20.01.2019)
- [WEL-o.J.] Well-Known Text (WKT). Vertica Analytics Platform. Online unter: https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/AnalyzingData/Geospatial/Spatial_Definitions/WellknownTextWKT.htm?TocPath=Analyzing%20Data%7CGeospatial%20Analytics%7COGC%20Spatial%20Definitions%7CSpatial%20Object%20Representations%7C____1, (Stand: 20.01.2019)
- [WHA-15] What is a Shapefile? GIS Lounge. 2015. Online unter: <https://www.gislounge.com/what-is-a-shapefile/>, (Stand: 15.01.2019)
- [WHA-16] What is a Shapefile? 2016. Online unter: <http://desktop.arcgis.com/de/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm>, (Stand: 05.08.2018)
- [WHA-18] What is a DXF File? 2018. Online unter: <https://www.coreldraw.com/en/pages/dxf-file/> (Stand: 05.08.2018)
- [WIC-02] WICKENBURG, S., ROOCH, A., GROSS, J.: Das JPEG-Verfahren. 2002. Online unter: https://www.mathematik.de/spudema/spudema_beitraege/beitraege/rooch/nkap04.html, (Stand: 19.01.2019)
- [WIN-12] WINTNER, S.: Digital Storytelling angewandt auf einen taktilen Hyperglobus. Diplomarbeit. Wien. 2012. 159 S.

12 Anhang

12.1 XML

XML korrekt

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes">
2 <Straße Typ="Ortsstraße">
3   <LineColor>
4     <Farbe>schwarz</Farbe>
5   </LineColor>
6   <LineWidth>2</LineWidth>
7   <LineType>MultiLineString</LineType>
8 </Straße>

```

XML inkorrekt

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes">
2 <Straße Typ="Ortsstraße">
3   <LineColor>
4     <Farbe>schwarz
5   </LineColor>
6   </Farbe>
7   <LineWidth>2
8   <LineType>MultiLineString
9   </LineWidth>
10  </LineType>
11 </Straße>

```

XML – Dateistruktur

```

2 <!-- Das ist ein XML-Beispiel -->
3
4 <studenten>
5   <student>
6     <matrikelnummer>01107335</matrikelnummer>
7     <vorname>Sara Lena</vorname>
8     <nachname>Kordasch</nachname>
9     <lehrveranstaltungen>
10      <lv>Multimediatechnologie</lv>
11      <lv>Modellierung</lv>
12      <lv>ANVIS</lv>
13    </lehrveranstaltungen>
14  </student>
15 </studenten>
16
17 <studenten>
18   <student>
19     <matrikelnummer>01107418</matrikelnummer>
20     <vorname>Maxi</vorname>
21     <nachname>Mustermann</nachname>
22     <lehrveranstaltungen>
23      <lv>Thematische Kartographie</lv>
24      <lv>AGI</lv>
25      <lv>MEGIDA</lv>
26    </lehrveranstaltungen>
27  </student>
28 </studenten>

```

12.2 JSON

12.2.1 JSON-Dateistruktur

```

2  /* Das ist ein JSON Beispiel */
3
4  {
5    "Studenten": {
6      "1": {
7        "Matrikelnummer": "01107335"
8        "Vorname": "Sara Lena"
9        "Nachname": "Kordasch"
10       "Lehrveranstaltungen": [
11         "Multimediatechnologie",
12         "Modellierung",
13         "ANVIS"
14       ]
15     },
16     "2": {
17       "Matrikelnummer": "01107418"
18       "Vorname": "Maxi"
19       "Nachname": "Mustermann"
20       "Lehrveranstaltungen": [
21         "Thematische Kartographie",
22         "AGI",
23         "MEGIDA"]
24     }
25   }
26 }

```

12.3 Stylesheets

CSS Beispiel – HTML

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <link rel="stylesheet" href="style.css">
5    </head>
6    <body>
7
8      <h1>Dies ist ein Beispiel.</h1>
9      <p>Die graphische Gestaltung dieses Textes wurde
10     im CSS Dokument definiert.</p>
11
12    </body>
13  </html>

```

```

1  body {
2    background-color: #D8D5D5;
3  }
4  h1 {
5    color: #242424;
6    opacity: 0.8;
7    font-family: arial;
8    font-size: 20px;
9    font-style: italic;
10   text-shadow:
11     -1px -1px 0 #FFFFFF,
12     1px -1px 0 #FFFFFF,
13     -1px 1px 0 #FFFFFF,
14     1px 1px 0 #FFFFFF;
15  }
16  p {
17    color: #0B1783;
18    font-family: arial;
19    font-size: 15px;
20    font-style: normal;
21    font-weight: bold;
22  }

```

Cartographic Stylesheet – ArcGIS Pro (*.lyr)

```

1  !DOCTYPE qgis-layer-definition
2  <!DOCTYPE qgis-layer-definition>
3  <qlr>
4  <layer-tree-group expanded="1" name="" checked="Qt::Checked">
5  <customproperties/>
6  <layer-tree-layer expanded="1" providerKey="ogr" id="floodevents_2017_mai_f295f8d9_50e0_455d_9b2d_0128e2ac6f9d"
7  source="..\Documents\Uni\Masterarbeit\flood_ausschnitt_2017\floodevents_2017_mai.shp" name="floodevents_2017_mai"
8  checked="Qt::Checked">
9  <customproperties/>
10 </layer-tree-layer>
11 </layer-tree-group>
12 <maplayers>
13 <maplayer maxScale="0" refreshOnNotifyMessage="" geometry="Polygon" labelsEnabled="0" minScale="1e+8"
14 simplifyDrawingHints="1" simplifyLocal="1" autoRefreshTime="0" autoRefreshEnabled="0" type="vector" simplifyMaxScale="1"
15 simplifyDrawingTol="1" refreshOnNotifyEnabled="0" simplifyAlgorithm="0" readOnly="0" hasScaleBasedVisibilityFlag="0">
16 <extent>
17 <xmin>-88.6580369999999931</xmin>
18 <ymin>-34.36751199999999784</ymin>
19 <xmax>115.0304339999999963</xmax>
20 <ymax>46.76355699999999871</ymax>
21 </extent>
22 <id>floodevents_2017_mai_f295f8d9_50e0_455d_9b2d_0128e2ac6f9d</id>
23 <datasource>..\Documents\Uni\Masterarbeit\flood_ausschnitt_2017\floodevents_2017_mai.shp</datasource>
24 <keywordList>
25 <value></value>
26 </keywordList>
27 <layername>floodevents_2017_mai</layername>
28 <srs>
29 <spatialrefsys>
30 <proj4>+proj=longlat +ellps=WGS84 +no_defs</proj4>
31 <srsid>3239</srsid>
32 <srid>4030</srid>
33 <authid>EPSG:4030</authid>
34 <description>Unknown datum based upon the WGS 84 ellipsoid</description>
35 <projectionacronym>longlat</projectionacronym>
36 <ellipsoidacronym>WGS84</ellipsoidacronym>
37 <geographicflag>true</geographicflag>
38 </spatialrefsys>
39 </srs>
40 <resourceMetadata>
41 <identifier></identifier>
42 <parentidentifier></parentidentifier>
43 <language></language>
44 <type></type>
45 <title></title>
46 <abstract></abstract>
47 <links>
48 <fees></fees>

```

Cartographic Stylesheet – QGIS (*.qlr)

```

<!DOCTYPE qgis-layer-definition>
<qlr>
<layer-tree-group expanded="1" name="" checked="Qt::Checked">
  <customproperties/>
  <layer-tree-layer expanded="1" providerKey="ogr" id="floodevents_2017_mai_f295f8d9_50e0_455d_9b2d_0128e2ac6f9d"
source="..\Documents\Uni\Masterarbeit\flood_ausschnitt_2017\floodevents_2017_mai.shp" name="floodevents_2017_mai"
checked="Qt::Checked">
    <customproperties/>
  </layer-tree-layer>
</layer-tree-group>
<maplayers>
  <maplayer maxScale="0" refreshOnNotifyMessage="" geometry="Polygon" labelsEnabled="0" minScale="1e+8"
simplifyDrawingHints="1" simplifyLocal="1" autoRefreshTime="0" autoRefreshEnabled="0" type="vector" simplifyMaxScale="1"
simplifyDrawingTol="1" refreshOnNotifyEnabled="0" simplifyAlgorithm="0" readOnly="0" hasScaleBasedVisibilityFlag="0">
    <extent>
      <xmin>-88.6580369999999931</xmin>
      <ymin>-34.36751199999999784</ymin>
      <xmax>115.0304339999999963</xmax>
      <ymax>46.76355699999999871</ymax>
    </extent>
    <id>floodevents_2017_mai_f295f8d9_50e0_455d_9b2d_0128e2ac6f9d</id>
    <datasource>..\Documents\Uni\Masterarbeit\flood_ausschnitt_2017\floodevents_2017_mai.shp</datasource>
    <keywordList>
      <value></value>
    </keywordList>
    <layername>floodevents_2017_mai</layername>
    <srs>
      <spatialrefsys>
        <proj4>+proj=longlat +ellps=WGS84 +no_defs</proj4>
        <srsid>3239</srsid>
        <srid>4030</srid>
        <authid>EPSG:4030</authid>
        <description>Unknown datum based upon the WGS 84 ellipsoid</description>
        <projectionacronym>longlat</projectionacronym>
        <ellipsoidacronym>WGS84</ellipsoidacronym>
        <geographicflag>true</geographicflag>
      </spatialrefsys>
    </srs>
    <resourceMetadata>
      <identifier></identifier>
      <parentidentifier></parentidentifier>
      <language></language>
      <type></type>
      <title></title>
      <abstract></abstract>
      <links>
      <fees></fees>

```

```

<encoding></encoding>
<crs>
  <spatialrefsys>
    <proj4></proj4>
    <srsid>0</srsid>
    <srid>0</srid>
    <authid></authid>
    <description></description>
    <projectionacronym></projectionacronym>
    <ellipsoidacronym></ellipsoidacronym>
    <geographicflag>false</geographicflag>
  </spatialrefsys>
</crs>
<extent/>
</resourceMetadata>
<provider encoding="System">ogr</provider>
<vectorjoins/>
<layerDependencies/>
<dataDependencies/>
<legend type="default-vector"/>
<expressionfields/>
<map-layer-style-manager current="default">
  <map-layer-style name="default"/>
</map-layer-style-manager>
<auxiliaryLayer/>
<renderer-v2 forceraster="0" type="singleSymbol" enableorderby="0" symbollevels="0">
  <symbols>
    <symbol clip_to_extent="1" type="fill" alpha="1" name="0">
      <layer enabled="1" pass="0" locked="0" class="SimpleFill">
        <prop v="3x:0,0,0,0,0,0" k="border_width_map_unit_scale"/>
        <prop v="164,113,88,255" k="color"/>
        <prop v="bevel" k="joinstyle"/>
        <prop v="0,0" k="offset"/>
        <prop v="3x:0,0,0,0,0,0" k="offset_map_unit_scale"/>
        <prop v="MM" k="offset_unit"/>
        <prop v="35,35,35,255" k="outline_color"/>
        <prop v="solid" k="outline_style"/>
        <prop v="0.26" k="outline_width"/>
        <prop v="MM" k="outline_width_unit"/>
        <prop v="solid" k="style"/>
      <data_defined_properties>
        <Option type="Map">
          <Option type="QString" value="" name="name"/>
          <Option name="properties"/>
          <Option type="QString" value="collection" name="type"/>
        </Option>
      </data_defined_properties>
    </layer>
  </symbol>
</symbols>
<rotation/>
<scales/>
</renderer-v2>
<customproperties>
  <property key="dualview/previewExpressions" value="&quot;ID&quot;"/>
</customproperties>
<blendMode>0</blendMode>
<featureBlendMode>0</featureBlendMode>
<layerOpacity>1</layerOpacity>
<fieldConfiguration>
  <field name="ID">
    <editWidget type="">
      <config>
        <Option/>
      </config>
    </editWidget>
  </field>
  <field name="GlideNumbe">
    <editWidget type="">
      <config>
        <Option/>
      </config>
    </editWidget>
  </field>
  <field name="Country">
    <editWidget type="">
      <config>

```

```

        <Option/>
    </config>
</editWidget>
</field>
<field name="OtherCount">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="long">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="lat">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Area">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Began">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Ended">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Validation">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Dead">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Displaced">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="MainCause">
    <editWidget type="">
        <config>
            <Option/>
        </config>
    </editWidget>
</field>
<field name="Severity">
    <editWidget type="">
        <config>

```

```

    </Option/>
  </config>
</editWidget>
</field>
</fieldConfiguration>
<aliases>
  <alias field="ID" name="" index="0"/>
  <alias field="GlideNumbe" name="" index="1"/>
  <alias field="Country" name="" index="2"/>
  <alias field="OtherCount" name="" index="3"/>
  <alias field="long" name="" index="4"/>
  <alias field="lat" name="" index="5"/>
  <alias field="Area" name="" index="6"/>
  <alias field="Began" name="" index="7"/>
  <alias field="Ended" name="" index="8"/>
  <alias field="Validation" name="" index="9"/>
  <alias field="Dead" name="" index="10"/>
  <alias field="Displaced" name="" index="11"/>
  <alias field="MainCause" name="" index="12"/>
  <alias field="Severity" name="" index="13"/>
</aliases>
<excludeAttributesWMS/>
<excludeAttributesWFS/>
<defaults>
  <default expression="" applyOnUpdate="0" field="ID"/>
  <default expression="" applyOnUpdate="0" field="GlideNumbe"/>
  <default expression="" applyOnUpdate="0" field="Country"/>
  <default expression="" applyOnUpdate="0" field="OtherCount"/>
  <default expression="" applyOnUpdate="0" field="long"/>
  <default expression="" applyOnUpdate="0" field="lat"/>
  <default expression="" applyOnUpdate="0" field="Area"/>
  <default expression="" applyOnUpdate="0" field="Began"/>
  <default expression="" applyOnUpdate="0" field="Ended"/>
  <default expression="" applyOnUpdate="0" field="Validation"/>
  <default expression="" applyOnUpdate="0" field="Dead"/>
  <default expression="" applyOnUpdate="0" field="Displaced"/>
  <default expression="" applyOnUpdate="0" field="MainCause"/>
  <default expression="" applyOnUpdate="0" field="Severity"/>
</defaults>
<constraints>
  <constraint field="ID" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="GlideNumbe" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Country" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="OtherCount" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="long" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="lat" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Area" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Began" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Ended" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Validation" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Dead" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Displaced" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="MainCause" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
  <constraint field="Severity" notnull_strength="0" unique_strength="0" constraints="0" exp_strength="0"/>
</constraints>
<constraintExpressions>
  <constraint field="ID" exp="" desc=""/>
  <constraint field="GlideNumbe" exp="" desc=""/>
  <constraint field="Country" exp="" desc=""/>
  <constraint field="OtherCount" exp="" desc=""/>
  <constraint field="long" exp="" desc=""/>
  <constraint field="lat" exp="" desc=""/>
  <constraint field="Area" exp="" desc=""/>
  <constraint field="Began" exp="" desc=""/>
  <constraint field="Ended" exp="" desc=""/>
  <constraint field="Validation" exp="" desc=""/>
  <constraint field="Dead" exp="" desc=""/>
  <constraint field="Displaced" exp="" desc=""/>
  <constraint field="MainCause" exp="" desc=""/>
  <constraint field="Severity" exp="" desc=""/>
</constraintExpressions>
<attributeactions>
  <defaultAction key="Canvas" value="{00000000-0000-0000-0000-000000000000}" />
</attributeactions>
<attributetableconfig sortExpression="" actionWidgetStyle="dropDown" sortOrder="0">
  <columns>
    <column type="field" width="-1" hidden="0" name="ID"/>

```

[illegible]

12.5 GML

GML – Quellcode intern

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <app:RoadSegment gml: id 0 "RS001">
3    <app:name>Universitätsstraße</app:name>
4    <gml:centerLine>
5      <gml:LineString gml:id="L001">
6        <gml:coordinates>1,2,3,3,4,4,0</gml:coordinates>
7      </gml:LineString>
8    </gml:centerLine>
9    <app:numberOfLanes>2</app:numberOfLanes>
10   <app:surfaceType>Asphalt</gml:surfaceType>
11 </app:RoadSegment>
12

```

Quellcode extern + Referenzdatei

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <app:RoadSegment gml: id 0 "RS001">
3    <app:name>Blumengasse</app:name>
4    <gml:centerLine xlink:href="#L002"/>
5    <app:numberOfLanes>2</app:numberOfLanes>
6    <app:surfaceType>Asphalt</gml:surfaceType>
7  </app:RoadSegment>

1  <?xml version="1.0" encoding="utf-8" ?>
2  <app:RoadSegment gml: id 0 "RS001">
3    <app:name>Blumengasse</app:name>
4    <gml:centerLine>
5      <gml:LineString gml:id="L002">
6        <gml:coordinates>1,2,3,3,4,4,0</gml:coordinates>
7      </gml:LineString>
8    </gml:centerLine>
9    <app:numberOfLanes>2</app:numberOfLanes>
10   <app:surfaceType>Asphalt</gml:surfaceType>
11 </app:RoadSegment>

```

12.6 GeoJSON

GeoJSON Dateistruktur

```

1  /* Das ist ein GeoJSON Beispiel */
2  {
3    "type": "FeatureCollection",
4    "features": [
5      {
6        "type": "Feature",
7        "geometry": {
8          "type": "LineString",
9          "coordinates": [
10             [
11               7.11227700501511,
12               50.724511372039
13             ],
14             [
15               7.11320378477021,
16               50.7232906914878
17             ]
18           ]
19         },
20         "properties": {
21           "geoTyp": "Linie"
22         }
23       },
24       {
25        "type": "Feature",
26        "geometry": {
27          "type": "Polygon",
28          "coordinates": [
29             [
30               [
31                 -0.289431323256268,
32                 0.00451046354601
33               ],
34               [
35                 -0.272488103953148,
36                 0.004461017869546
37               ],
38               [
39                 -0.271961932650073,
40                 -0.0237003885651537
41               ]
42             ]
43           ]
44         }
45       }
46     ]
47   }

```

CSS simplestyle-Spezifikation

```

1  {
2    "type": "FeatureCollection",
3    "features": [
4      {
5        "type": "Feature",
6        "geometry": {
7          "type": "Point",
8          "coordinates": [0, 0]
9        },
10       "properties": {
11         "title": "A title",
12         "description": "A description",
13         "marker-size": "medium",
14         "marker-symbol": "bus",
15         "marker-color": "#fff",
16         "stroke": "#555555",
17         "stroke-opacity": 1.0,
18         "stroke-width": 2,
19         "fill": "#555555",
20         "fill-opacity": 0.5
21       }
22     ]
23   }

```

CSS interne Stilisierung

```

{
  "type": "Feature",
  "properties": {
    "ID": 4433,
    "GlideNumbe": null,
    "Country": "Germany",
    "OtherCount": null,

```

```

        "long": 9.583276,
        "lat": 54.705274,
        "Area": 18991.845394600001,
        "Began": "2017\01\02",
        "Ended": "2017\01\05",
        "Validation": "News",
        "Dead": 0,
        "Displaced": 0,
        "MainCause": "Winter Storm Axel",
        "Severity": 1.5,
        "fill": "#DCDCDC",
        "stroke-width": 1,
        "fill-opacity": 0.6
    },
    "geometry": {
        "type": "Polygon",
        "coordinates": [[
            "type": "MultiPolygon",
            "coordinates":
            [
                [
                    [ 13.532608, 54.306792 ],
                    [ 13.33466, 53.662268 ],
                    [ 10.414927, 53.415802 ],
                    [ 9.425187, 54.457616 ],
                    [ 9.227239, 55.251098 ],
                    [ 9.524161, 55.994746 ],
                    [ 9.920057, 55.944961 ],
                    [ 9.623135, 54.953381 ],
                    [ 10.019031, 54.407731 ],
                    [ 10.761336, 54.109839 ],
                    [ 11.058258, 53.911344 ],
                    [ 12.493381, 54.009796 ],
                    [ 13.532608, 54.306792 ]
                ]
            ]
        ]
    }
}

```

CSS externe Stilisierung

```

{
    "type": "FeatureCollection",
    "name": "Flood_2017_GEOJSON",
    "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::4030" } },
    "className": {
        "baseVal": "torrentialrain"
    }
    "features": [
    { "type": "Feature",
        "properties": {
            "ID": 4433,
            "GlideNumbe": null,
            "Country": "Germany",
            "OtherCount": null,
            "long": 9.583276,
            "lat": 54.705274,
            "Area": 18991.845394600001,
            "Began": "2017\01\02",
            "Ended": "2017\01\05",
            "Validation": "News",
            "Dead": 0,
            "Displaced": 0,
            "MainCause": "Torrential Rain",
            "Severity": 1.5
        },
        "geometry": {
            "type": "MultiPolygon",
            "coordinates": [ [ [
                [ 13.532608, 54.306792 ],
                [ 13.33466, 53.662268 ],
                [ 10.414927, 53.415802 ],
                [ 9.425187, 54.457616 ],

```

```

        [ 9.227239, 55.251098 ],
        [ 9.524161, 55.994746 ],
        [ 9.920057, 55.944961 ],
        [ 9.623135, 54.953381 ],
        [ 10.019031, 54.407731 ],
        [ 10.761336, 54.109839 ],
        [ 11.058258, 53.911344 ],
        [ 12.493381, 54.009796 ],
        [ 13.532608, 54.306792 ]
      ] ]
    }
  },
  { "type": "Feature",
    "properties": {
      "ID": 4490,
      "GlideNumbe": null,
      "Country": "Nepal",
      "OtherCount": null,
      "long": 86.426951,
      "lat": 27.294975,
      "Area": 54260.541468299998,
      "Began": "2017\06\30",
      "Ended": "2017\07\03",
      "Validation": "News",
      "Dead": 11,
      "Displaced": 6000,
      "MainCause": "Torrential Rain",
      "Severity": 1.0
    },
    "geometry": {
      "type": "MultiPolygon",
      "coordinates": [ [ [
        [ 87.812587, 27.740706 ],
        [ 88.307457, 27.641284 ],
        [ 88.505405, 26.351937 ],
        [ 87.416691, 26.352484 ],
        [ 84.348497, 27.246582 ],
        [ 84.942341, 28.238013 ],
        [ 87.812587, 27.740706 ]
      ] ] ]
    }
  }
]
}

```

<style type="text/css">

```

.torrentialrain{
stroke:#DCDCDC;
stroke-width:1;
fill:#DCDCDC;
opacity: 0.6;
}

```

CSS kategorisierte graphische Aufbereitung

```

{ "type": "FeatureCollection",
  "className":{
    "baseVal":"Linie"
  },
  "features": [
    { "type": "Feature",
      "geometry": {
        "type": "Line",
        "coordinates": [0, 1]
      },
      "properties": {
        "title": "A title",
        "description": "A description",
      },
    },
    { "type": "Feature",
      "geometry": {
        "type": "Line",
        "coordinates": [1, 0]
      },
    }
  ]
}

```

```

    },
    "properties": {
      "title": "Another title",
      "description": "Another description",
    }
  }
}

<style type="text/css">

.Linie{
  stroke:#DDDDDD;
  stroke-width: 1;
  fill:#DDDDDD;
  opacity: 0.6;
}

```

12.7 SVG

SVG Codebeispiel Ellipse

```

<svg height="150" width="400">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%"
x2="100%" y2="0%">
      <stop offset="0%" style="stop-
color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%" style="stop-
color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="200" cy="70" rx="85" ry="55"
fill="url(#grad1)" />
</svg>

```

12.8 OmniSuite Punktformat

Punktdatensatz der Global Story

```

lat long name category start end value
-20.7778 34.5151 Mozambique 1 0.00 11.06 1.5
54.7053 9.58328 Germany 4 0.30 1.20 1.5
39.6236 -122.458 USA 1 0.90 1.20 1.5
-29.4094 -55.5512 Brazil 1 1.20 5.98 1
1.6911 -76.2863 Colombia 1 1.20 11.06 1
15.5805 -85.3672 Honduras 1 1.79 5.98 1
8.08572 125.027 Philippines 1 4.48 5.98 1
35.0839 -119.043 USA 1 5.68 5.98 1
3.93172 102.461 Malaysia 1 6.58 11.06 1
7.59046 124.062 Philippines 1 9.27 11.06 1.5
-5.25083 -77.573 Peru 1 9.27 11.06 1
-11.0042 -75.1481 Peru 1 9.27 23.91 2
38.804 -120.825 USA 1 13.75 18.23 1.5
-18.6428 29.022 Zimbabwe 4 13.75 18.23 1.5
-27.2717 30.5561 South%20Africa 4 14.05 18.23 1
-7.0293 107.162 Indonesia 1 15.24 18.23 1
3.72258 -72.6737 Colombia 1 15.84 18.23 1
-30.9395 -70.2489 Chile 1 16.14 18.23 2
-20.5698 17.0462 Namibia 1 20.03 23.91 1
-31.6956 150.76 Australia 1 21.22 23.91 1
38.0498 -1.33823 Spain 2 21.22 23.91 1
-1.9573 29.8386 Burundi 1 22.12 23.91 1
45.5453 -116.123 USA 1 22.72 23.91 1
-8.71752 13.6563 Angola 1 23.61 28.99 1

```

-21.975 148.682 Australia 4 25.71 28.99 2
24.3178 90.6828 Bangladesh 1 26.30 31.98 1
2.18701 -76.1131 Colombia 1 26.90 28.99 2
-10.3642 33.8718 Malawi 1 27.80 31.98 1
-38.2042 176.79 New%20Zealand 1 28.10 32.88 2
-45.9533 -67.9388 Argentina 1 28.10 32.88 1.5
-27.4231 -62.6279 Argentina 1 28.10 32.88 1.5
51.2542 70.0468 Kazakhstan 1 30.79 32.88 1.5
37.7043 46.1693 Iran 31.09 32.88 2
25.3101 89.5446 Bangladesh 1 31.39 36.17 1
18.9198 -69.853 Dominican%20Republic 33.48 36.17 1.5
34.9912 -81.5814 USA 1 34.38 36.17 1.5
37.5744 -91.1819 USA 1 34.97 36.17 1.5
45.5748 -74.4058 Canada 1 37.07 40.05 1.5
23.8607 112.952 China 1 37.66 40.05 1
-2.87879 37.8802 Kenya 1 37.96 40.05 1
-27.1217 -69.952 Chile 1 38.86 40.05 1.5
-28.3624 29.7149 South%20Africa 2 39.76 41.25 1
37.5473 -86.3816 USA 2 41.25 41.85 1
3.81464 -59.0153 Guyana 1 41.55 43.04 1.5
7.6124 80.8349 Sri%20Lanka 4 42.45 49.02 1.5
-31.8884 -57.4317 Uruguay 1 43.04 49.02 1.5
-8.09623 -38.2308 Brazil 1 43.04 47.83 1.5
16.8247 -96.4275 Mexico 4 45.14 46.33 1
24.3022 121.216 Taiwan 1 45.43 46.33 1.5
27.6131 94.9387 India 3 45.43 55.61 1
13.4982 -87.2229 Honduras 1 47.83 49.83 1
22.2842 92.1427 Bangladesh 3 48.13 49.83 1.5
7.76344 125.249 Philippines 1 48.42 49.83 1
13.7015 1.11137 Niger 1 48.72 55.61 1.5
45.4855 3.33829 France 2 48.72 50.14 1
-33.4188 -70.9912 Chile 1 49.02 51.66 1
25.9426 112.853 China 3 51.41 55.61 1.5
33.4014 72.3232 Pakistan 3 52.31 55.61 1
27.295 86.427 Nepal 2 53.80 55.61 1

CSS-Datei des Punktdatensatzes

```

1  .category1 {
2    marker-symbol: ellipsoid;
3    marker-width: 25 * [value] * 2.1;
4    marker-height: 25 * [value] * 2.1;
5    marker-color: rgba(219, 241, 252, 0);
6    marker-outline-style: solid;
7    marker-outline-width: 4px;
8    marker-outline-color: rgba(219, 241, 252, 255);
9    text-name: [name];
10   text-dy: 5;
11   text-placement: S;
12 }
13
14 .category2 {
15   marker-symbol: ellipsoid;
16   marker-width: 25 * [value] * 2.1;
17   marker-height: 25 * [value] * 2.1;
18   marker-color: rgba(2, 37, 237, 0);
19   marker-outline-style: solid;
20   marker-outline-width: 4px;
21   marker-outline-color: rgba(2, 37, 237, 255);
22   text-name: [name];
23   text-dy: 5;
24   text-placement: S;
25 }
26
27 .category3 {
28   marker-symbol: ellipsoid;
29   marker-width: 25 * [value] * 2.1;
30   marker-height: 25 * [value] * 2.1;
31   marker-color: rgba(2, 37, 237, 0);
32   marker-outline-style: solid;
33   marker-outline-width: 4px;
34   marker-outline-color: rgba(2, 37, 237, 255);
35   text-name: [name];
36   text-dy: 5;
37   text-placement: S;
38 }
39
40 .category4 {
41   marker-symbol: ellipsoid;
42   marker-width: 25 * [value] * 2.1;
43   marker-height: 25 * [value] * 2.1;
44   marker-color: rgba(221, 0, 214, 0);
45   marker-outline-style: solid;
46   marker-outline-width: 4px;
47   marker-outline-color: rgba(221, 0, 214, 255);
48   text-name: [name];
49   text-dy: 5;
50   text-placement: S;
51 }

```

12.9 KML

KML Dateistruktur

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <kml xmlns="http://www.opengis.net/kml/2.2">
3    <Document id="root_doc">
4      <Schema name="RoadSegment" id="RoadSegment">
5        <SimpleField name="id" type="float"></SimpleField>
6      </Schema>
7      <Folder><name>RoadSegment</name>
8        <Placemark>
9          <Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>0</fill></PolyStyle></Style>
10         <MultiGeometry><LineString><coordinates>
11           -139.765920467186,
12           -52.401883612923 -120.214196329255,
13           -23.5734612046021 -130.804713570634,
14           21.6340175048673 -150.356437708565,
15           -2.27186361135497 -146.283161846496,
16           -44.885424616852</coordinates></LineString></MultiGeometry>
17       </Placemark>
18     </Folder>
19   </Document></kml>

```

KML Remote-Objekte

```

1  <app:Bridge gml:id="B1">
2    <app:span>400</app:span>
3    <app:height>50</app:height>
4    <app:mobility>DrawBridge</app:mobility>
5    <gml:centerOf>
6      <gml:Point gml:id="P1" srsName="..">
7        <gml:pos>100.1 200.3</gml:pos>
8      </gml:Point>
9    </gml:centerOf>
10  </app:Bridge>
11  <app:Tower gml:id="T1">
12    <app:height>90</app:height>
13    <gml:centerOf
14      <xlink:href="#"TransportationBridges.xml#P1"/>
15  </app:Tower>

```

```

1  <app:Bridge gml:id="B1">
2    <app:span>400</app:span>
3    <app:height>50</app:height>
4    <app:mobility>DrawBridge</app:mobility>
5    <gml:centerOf>
6      <gml:Point gml:id="P1" srsName="..">
7        <gml:pos>100.1 200.3</gml:pos>
8      </gml:Point>
9    </gml:centerOf>
10  </app:Bridge>
11  <app:Tower gml:id="T1">
12    <app:height>90</app:height>
13    <gml:centerOf xlink:href="#P1"/>
14  </app:Tower>

```

KML Style URL

```

<?xml version="1.0" encoding="utf-8" ?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id="root_doc">
<Schema name="Flood_2017_KML" id="Flood_2017_KML">
  <SimpleField name="ID" type="int"></SimpleField>
  <SimpleField name="GlideNumbe" type="string"></SimpleField>
  <SimpleField name="Country" type="string"></SimpleField>
  <SimpleField name="OtherCount" type="string"></SimpleField>
  <SimpleField name="long" type="float"></SimpleField>
  <SimpleField name="lat" type="float"></SimpleField>
  <SimpleField name="Area" type="float"></SimpleField>
  <SimpleField name="Began" type="string"></SimpleField>
  <SimpleField name="Ended" type="string"></SimpleField>
  <SimpleField name="Validation" type="string"></SimpleField>
  <SimpleField name="Dead" type="int"></SimpleField>
  <SimpleField name="Displaced" type="int"></SimpleField>
  <SimpleField name="MainCause" type="string"></SimpleField>
  <SimpleField name="Severity" type="float"></SimpleField>
</Schema>
<Folder><name>Flood_2017_KML</name>
<Style id="transGreyPoly">
  <LineStyle>
    <width>1</width>
    <color>64AAAAAA</color>
  </LineStyle>
  <PolyStyle>
    <color>64AAAAAA</color>
  </PolyStyle>
</Style>
<Placemark>
  <styleUrl>#transGreyPoly</styleUrl>
  <ExtendedData><SchemaData schemaUrl="#Flood_2017_KML">
    <SimpleData name="ID">4433</SimpleData>
    <SimpleData name="Country">Germany</SimpleData>
    <SimpleData name="long">9.583276000000000</SimpleData>
    <SimpleData name="lat">54.705274000000003</SimpleData>
    <SimpleData name="Area">18991.845394600000873</SimpleData>
    <SimpleData name="Began">2017/01/02</SimpleData>
    <SimpleData name="Ended">2017/01/05</SimpleData>
    <SimpleData name="Validation">News</SimpleData>

```

```

        <SimpleData name="Dead">0</SimpleData>
        <SimpleData name="Displaced">0</SimpleData>
        <SimpleData name="MainCause">Winter Storm Axel</SimpleData>
        <SimpleData name="Severity">1.5000000000000000</SimpleData>
    </SchemaData></ExtendedData>
    <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13.532608,54.306792      13.33466,53.662268
10.414927,53.415802      9.425187,54.457616      9.227239,55.251098      9.524161,55.994746      9.920057,55.944961      9.623135,54.953381
10.019031,54.407731      10.761336,54.109839      11.058258,53.911344      12.493381,54.009796
13.532608,54.306792</coordinates></LinearRing></outerBoundaryIs></Polygon></MultiGeometry>
    </Placemark>
    <Placemark>
        <styleUrl>#transGreyPoly</styleUrl>
        <ExtendedData><SchemaData schemaUrl="#Flood_2017_KML">
            <SimpleData name="ID">4434</SimpleData>
            <SimpleData name="Country">USA</SimpleData>
            <SimpleData name="long">-122.457654000000005</SimpleData>
            <SimpleData name="lat">39.623615000000001</SimpleData>
            <SimpleData name="Area">107407.257465000002412</SimpleData>
            <SimpleData name="Began">2017/01/04</SimpleData>
            <SimpleData name="Ended">2017/01/05</SimpleData>
            <SimpleData name="Validation">News</SimpleData>
            <SimpleData name="Dead">0</SimpleData>
            <SimpleData name="Displaced">0</SimpleData>
            <SimpleData name="MainCause">Heavy Rain</SimpleData>
            <SimpleData name="Severity">1.5000000000000000</SimpleData>
        </SchemaData></ExtendedData>
        <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>-122.012272,37.317161      -122.358681,37.168575      -
123.150473,38.309462      -123.447395,39.003822      -123.744317,39.896527      -124.387647,40.194369      -124.090725,40.640498      -
124.387647,42.078655      -121.863811,41.829455      -120.527662,38.308144      -
122.012272,37.317161</coordinates></LinearRing></outerBoundaryIs></Polygon></MultiGeometry>
    </Placemark>
</Folder>
</Document></kml>

```

KML klassifiziert

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<Document>
<name>Flood_2017_KML</name>
<Style id="style1">
    <LineStyle>
        <color>40000000</color>
        <width>3</width>
    </LineStyle>
    <PolyStyle>
        <color>ffdb0c2e</color>
        <fill>1</fill>
        <outline>1</outline>
    </PolyStyle>
</Style>
<Style id="style2">
    <LineStyle>
        <color>40000000</color>
        <width>3</width>
    </LineStyle>
    <PolyStyle>
        <color>ffded9b4</color>
        <fill>1</fill>
        <outline>1</outline>
    </PolyStyle>
</Style>
<Style id="style3">
    <LineStyle>
        <color>40000000</color>
        <width>3</width>
    </LineStyle>
    <PolyStyle>
        <color>ffded9b4</color>
        <fill>1</fill>
        <outline>1</outline>
    </PolyStyle>
</Style>
<Style id="style4">

```

```

        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ffded9b4</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style5">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ffdb062a</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style6">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ffded9b4</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style7">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ffe7dcc3</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style8">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ffdbb13c</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style9">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ff02f7f3</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style10">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ff02f7f3</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style11">

```

```

        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ff02f7f3</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style12">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>ff02f7f3</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Style id="style13">
        <LineStyle>
            <color>40000000</color>
            <width>3</width>
        </LineStyle>
        <PolyStyle>
            <color>fff0d687</color>
            <fill>1</fill>
            <outline>1</outline>
        </PolyStyle>
    </Style>
    <Placemark>
        <name>NULL</name>
        <description><![CDATA[<p>Name: NULL
<br/>description: NULL
<br/>timestamp: NULL
<br/>begin: NULL
<br/>end: NULL
<br/>altitudeMode: NULL
<br/>tessellate: -1
<br/>extrude: 0
<br/>visibility: -1
<br/>drawOrder: NULL
<br/>icon: NULL
<br/>ID: 4433
<br/>GlideNumbe: NULL
<br/>Country: Germany
<br/>OtherCount: NULL
<br/>long: 9.583276
<br/>lat: 54.705274
<br/>Area: 18991.8453946
<br/>Began: 2017/01/02
<br/>Ended: 2017/01/05
<br/>Validation: News
<br/>Dead: 0
<br/>Displaced: 0
<br/>MainCause: Winter Storm Axel
<br/>Severity: 1.5</p>]]></description>
        <ExtendedData>
        </ExtendedData>
        <styleUrl>#style9</styleUrl>
        <MultiGeometry>
            <Polygon>
                <outerBoundaryIs>
                    <LinearRing>
                        <tessellate>1</tessellate>
                        <coordinates>
                            13.532608,54.306792,0.00000
                            13.33466,53.662268,0.00000
                            10.414927,53.415802,0.00000
                            9.425187,54.457616,0.00000
                            9.227239,55.251098,0.00000
                            9.524161,55.994746,0.00000
                            9.920057,55.944961,0.00000
                            9.623135,54.953381,0.00000
                            10.019031,54.407731,0.00000

```

```

10.761336,54.109839,0.00000
11.058258,53.911344,0.00000
12.493381,54.009796,0.00000
13.532608,54.306792,0.00000
    </coordinates>
  </LinearRing>
</outerBoundaryIs>
</Polygon>
</MultiGeometry>
</Placemark>

<Placemark>
<name>NULL</name>
<description><![CDATA[<p>Name: NULL
<br/>description: NULL
<br/>timestamp: NULL
<br/>begin: NULL
<br/>end: NULL
<br/>altitudeMode: NULL
<br/>tessellate: -1
<br/>extrude: 0
<br/>visibility: -1
<br/>drawOrder: NULL
<br/>icon: NULL
<br/>ID: 4434
<br/>GlideNumbe: NULL
<br/>Country: USA
<br/>OtherCount: NULL
<br/>long: -122.457654
<br/>lat: 39.623615
<br/>Area: 107407.257465
<br/>Began: 2017/01/04
<br/>Ended: 2017/01/05
<br/>Validation: News
<br/>Dead: 0
<br/>Displaced: 0
<br/>MainCause: Heavy Rain
<br/>Severity: 1.5</p>]]></description>
<ExtendedData>
</ExtendedData>
<styleUrl>#style2</styleUrl>
<MultiGeometry>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>
        <tessellate>1</tessellate>
        <coordinates>
          -122.012272,37.317161,0.00000
          -122.358681,37.168575,0.00000
          -123.150473,38.309462,0.00000
          -123.447395,39.003822,0.00000
          -123.744317,39.896527,0.00000
          -124.387647,40.194369,0.00000
          -124.090725,40.640498,0.00000
          -124.387647,42.078655,0.00000
          -121.863811,41.829455,0.00000
          -120.527662,38.308144,0.00000
          -122.012272,37.317161,0.00000
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</MultiGeometry>
</Placemark>

<Placemark>
<name>NULL</name>
<description><![CDATA[<p>Name: NULL
<br/>description: NULL
<br/>timestamp: NULL
<br/>begin: NULL
<br/>end: NULL
<br/>altitudeMode: NULL
<br/>tessellate: -1
<br/>extrude: 0
<br/>visibility: -1
<br/>drawOrder: NULL

```

```

<br/>icon: NULL
<br/>ID: 4435
<br/>GlideNumbe: NULL
<br/>Country: Brazil
<br/>OtherCount: Argentina
<br/>long: -55.551237
<br/>lat: -29.409362
<br/>Area: 386846.737648
<br/>Began: 2017/01/05
<br/>Ended: 2017/01/21
<br/>Validation: News
<br/>Dead: 0
<br/>Displaced: 0
<br/>MainCause: Heavy Rain
<br/>Severity: 1.0<p>]]></description>
<ExtendedData>
</ExtendedData>
<styleUrl>#style2</styleUrl>
<MultiGeometry>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>
        <tessellate>1</tessellate>
        <coordinates>
          -50.157155,-29.908609,0.00000
          -52.829453,-32.783281,0.00000
          -57.184308,-30.500116,0.00000
          -60.747372,-27.423964,0.00000
          -60.94532,-26.035444,0.00000
          -54.808933,-26.831911,0.00000
          -50.157155,-29.908609,0.00000
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</MultiGeometry>
</Placemark>
</Document>
</kml>

```

KML externe Stilisierung

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:gx="http://www.google.com/kml/ext/2.2"
  xmlns:kml="http://www.opengis.net/kml/2.2" >
<Document id="root_doc">
<Schema name="Flood_2017_KML" id="Flood_2017_KML">
  <SimpleField name="ID" type="int"></SimpleField>
  <SimpleField name="GlideNumbe" type="string"></SimpleField>
  <SimpleField name="Country" type="string"></SimpleField>
  <SimpleField name="OtherCount" type="string"></SimpleField>
  <SimpleField name="long" type="float"></SimpleField>
  <SimpleField name="lat" type="float"></SimpleField>
  <SimpleField name="Area" type="float"></SimpleField>
  <SimpleField name="Began" type="string"></SimpleField>
  <SimpleField name="Ended" type="string"></SimpleField>
  <SimpleField name="Validation" type="string"></SimpleField>
  <SimpleField name="Dead" type="int"></SimpleField>
  <SimpleField name="Displaced" type="int"></SimpleField>
  <SimpleField name="MainCause" type="string"></SimpleField>
  <SimpleField name="Severity" type="float"></SimpleField>
</Schema>
<Folder><name>Flood_2017_KML</name>
<styleUrl>Flood_2017_KML_StyleUrl_Stylesheet.kml#transGreyPoly</styleUrl>
<Placemark>
  <styleUrl>#transGreyPoly</styleUrl>
  <ExtendedData><SchemaData schemaUrl="#Flood_2017_KML">
    <SimpleData name="ID">4433</SimpleData>
    <SimpleData name="Country">Germany</SimpleData>
    <SimpleData name="long">9.583276000000000</SimpleData>
    <SimpleData name="lat">54.705274000000003</SimpleData>
    <SimpleData name="Area">18991.845394600000873</SimpleData>

```

```

        <SimpleData name="Began">2017/01/02</SimpleData>
        <SimpleData name="Ended">2017/01/05</SimpleData>
        <SimpleData name="Validation">News</SimpleData>
        <SimpleData name="Dead">0</SimpleData>
        <SimpleData name="Displaced">0</SimpleData>
        <SimpleData name="MainCause">Winter Storm Axel</SimpleData>
        <SimpleData name="Severity">1.5000000000000000</SimpleData>
    </SchemaData></ExtendedData>
    <MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>13.532608,54.306792      13.33466,53.662268
10.414927,53.415802  9.425187,54.457616  9.227239,55.251098  9.524161,55.994746  9.920057,55.944961  9.623135,54.953381
10.019031,54.407731      10.761336,54.109839      11.058258,53.911344      12.493381,54.009796
13.532608,54.306792</coordinates></LinearRing></outerBoundaryIs></Polygon></MultiGeometry>
    </Placemark>
</Folder>
</kml>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:gx="http://www.google.com/kml/ext/2.2"
  xmlns:kml="http://www.opengis.net/kml/2.2" >
<Document>
  <name>Flood style</name>
  <Style id="transGreyPoly">
    <LineStyle>
      <color>40000000</color>
      <width>3</width>
    </LineStyle>
    <PolyStyle>
      <color>ffe7dcc3</color>
      <fill>1</fill>
      <outline>1</outline>
    </PolyStyle>
  </Style>
</Document>
</kml>

```

KML FadeIn

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2"
  xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">
  <Document>
    <name> fade on polygon</name>
    <description><![CDATA[
click here to reset start position
]]></description>
    <open>1</open>

    <Style id="style1">                                <!-- Definierter Polygon-Stil -->
      <LineStyle>
        <color>00ffffff</color>                                <!-- Begin Füllfarbe: Transparent -->
        <width>1</width>                                <!-- Begin Füllfarbe: Transparent -->
      </LineStyle>
      <PolyStyle>
        <color>00ffffff</color>                                <!-- Begin Füllfarbe: Transparent -->
      </PolyStyle>
    </Style>

    <gx:Tour>
      <name>Start the fade-in of the object</name>
      <gx:Playlist>

        <gx:Wait> <gx:duration>1</gx:duration></gx:Wait>

        <gx:AnimatedUpdate>
          <gx:duration>3</gx:duration>  <!-- Länge des Einblendens -->
          <Update>
            <targetHref></targetHref>
            <Change>
              <Style targetId="style1">                                <!-- Referenz Polygon-Stil -->
                <LineStyle>

```

```

        <color>64D2D2D2</color>      <!-- Finale Füllfarbe: Transparentes Grau -->
    </LineStyle>
    <PolyStyle>
        <color>64D2D2D2</color>      <!-- Finale Füllfarbe: Transparentes Grau
-->
        </PolyStyle>
    </Style>
    </Change>
    </Update>
</gx:AnimatedUpdate>

<gx:Wait> <gx:duration>3</gx:duration></gx:Wait>

</gx:Playlist>

</gx:Tour>

<Placemark id="polygon_name">      <!-- reference polygon name -->
    <name>NULL</name>
    <description><![CDATA[<p>Name: NULL
    <br/>description: NULL
    <br/>timestamp: NULL
    <br/>begin: NULL
    <br/>end: NULL
    <br/>altitudeMode: NULL
    <br/>tessellate: -1
    <br/>extrude: 0
    <br/>visibility: -1
    <br/>drawOrder: NULL
    <br/>icon: NULL
    <br/>ID: 4435
    <br/>GlideNumbe: NULL
    <br/>Country: Brazil
    <br/>OtherCount: Argentina
    <br/>long: -55.551237
    <br/>lat: -29.409362
    <br/>Area: 386846.737648
    <br/>Began: 2017/01/05
    <br/>Ended: 2017/01/21
    <br/>Validation: News
    <br/>Dead: 0
    <br/>Displaced: 0
    <br/>MainCause: Heavy Rain
    <br/>Severity: 1.0<p>]]>
    </description>
    <TimeSpan>
        <begin>2017-01-02</begin>
        <end>2017-01-05</end>
    </TimeSpan>
    <ExtendedData>
    </ExtendedData>
    <styleUrl>style1</styleUrl>      <!-- reference polygon style -->
    <MultiGeometry>
        <Polygon>
            <outerBoundaryIs>
                <LinearRing>
                    <tessellate>1</tessellate>
                    <coordinates>
                        -50.157155,-29.908609,0.00000
                        -52.829453,-32.783281,0.00000
                        -57.184308,-30.500116,0.00000
                        -60.747372,-27.423964,0.00000
                        -60.94532,-26.035444,0.00000
                        -54.808933,-26.831911,0.00000
                        -50.157155,-29.908609,0.00000
                    </coordinates>
                </LinearRing>
            </outerBoundaryIs>
        </Polygon>
    </MultiGeometry>
</Placemark>

<Placemark id="polygon_name">      <!-- reference polygon name -->
    <name>NULL</name>
    <description><![CDATA[<p>Name: NULL
    <br/>description: NULL

```

```

<br/>timestamp: NULL
<br/>begin: NULL
<br/>end: NULL
<br/>altitudeMode: NULL
<br/>tessellate: -1
<br/>extrude: 0
<br/>visibility: -1
<br/>drawOrder: NULL
<br/>icon: NULL
<br/>ID: 4438
<br/>GlideNumbe: NULL
<br/>Country: USA
<br/>OtherCount: NULL
<br/>long: -119.043052
<br/>lat: 35.083904
<br/>Area: 74443.1106463
<br/>Began: 2017/01/20
<br/>Ended: 2017/01/21
<br/>Validation: News
<br/>Dead: 0
<br/>Displaced: 0
<br/>MainCause: Heavy Rain
<br/>Severity: 1.0</p>]]></description>
<TimeSpan>
  <begin>2017-01-07</begin>
  <end>2017-01-10</end>
</TimeSpan>
<ExtendedData>
</ExtendedData>
<styleUrl>style1</styleUrl>                                <!-- reference polygon style -->
<MultiGeometry>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>
        <tessellate>1</tessellate>
        <coordinates>
          -118.003825,33.546575,0.00000
          -119.191513,33.943864,0.00000
          -120.478175,34.341202,0.00000
          -121.170993,35.333279,0.00000
          -118.597669,36.621234,0.00000
          -116.915111,34.835276,0.00000
          -118.003825,33.546575,0.00000
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</MultiGeometry>
</Placemark>
</Document>
</kml>

```

KML Animation

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<Folder>
  <name>Flood_2017_KML</name>
  <Style id="style2">
    <LineStyle>
      <color>40000000</color>
      <width>3</width>
    </LineStyle>
    <PolyStyle>
      <color>ffded9b4</color>
      <fill>1</fill>
      <outline>1</outline>
    </PolyStyle>
  </Style>
  <Placemark>
    <name>NULL</name>
    <description></description>
    <Placemark>
      <TimeSpan>

```

```

    <begin>2017-01-02</begin>
    <end>2017-01-05</end>
</TimeSpan>
<name>NULL</name>
<ExtendedData>
  <Data name="Name"><displayName>Name</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="description"><displayName>description</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="timestamp"><displayName>timestamp</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="begin"><displayName>begin</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="end"><displayName>end</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="altitudeMode"><displayName>altitudeMode</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="tessellate"><displayName>tessellate</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="extrude"><displayName>extrude</displayName><value><![CDATA[0]]></value></Data>
  <Data name="visibility"><displayName>visibility</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="drawOrder"><displayName>drawOrder</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="icon"><displayName>icon</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="ID"><displayName>ID</displayName><value><![CDATA[4433]]></value></Data>
  <Data name="GlideNumbe"><displayName>GlideNumbe</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="Country"><displayName>Country</displayName><value><![CDATA[Germany]]></value></Data>
  <Data name="OtherCount"><displayName>OtherCount</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="long"><displayName>long</displayName><value><![CDATA[9.583276]]></value></Data>
  <Data name="lat"><displayName>lat</displayName><value><![CDATA[54.705274]]></value></Data>
  <Data name="Area"><displayName>Area</displayName><value><![CDATA[18991.8453946]]></value></Data>
  <Data name="Began"><displayName>Began</displayName><value><![CDATA[2017/01/02]]></value></Data>
  <Data name="Ended"><displayName>Ended</displayName><value><![CDATA[2017/01/05]]></value></Data>
  <Data name="Validation"><displayName>Validation</displayName><value><![CDATA[News]]></value></Data>
  <Data name="Dead"><displayName>Dead</displayName><value><![CDATA[0]]></value></Data>
  <Data name="Displaced"><displayName>Displaced</displayName><value><![CDATA[0]]></value></Data>
  <Data name="MainCause"><displayName>MainCause</displayName><value><![CDATA[Winter Storm
Axel]]></value></Data>
  <Data name="Severity"><displayName>Severity</displayName><value><![CDATA[1.5]]></value></Data>
</ExtendedData>
<styleUrl>#style9</styleUrl>
<MultiGeometry>
  <Polygon>
    <outerBoundaryIs>
      <LinearRing>
        <tessellate>1</tessellate>
        <coordinates>
          13.532608,54.306792,0.00000
          13.33466,53.662268,0.00000
          10.414927,53.415802,0.00000
          9.425187,54.457616,0.00000
          9.227239,55.251098,0.00000
          9.524161,55.994746,0.00000
          9.920057,55.944961,0.00000
          9.623135,54.953381,0.00000
          10.019031,54.407731,0.00000
          10.761336,54.109839,0.00000
          11.058258,53.911344,0.00000
          12.493381,54.009796,0.00000
          13.532608,54.306792,0.00000
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</MultiGeometry>
</Placemark>

<Placemark>
<TimeSpan>
  <begin>2017-01-04</begin>
  <end>2017-01-05</end>
</TimeSpan>
<name>NULL</name>
<description></description>
<ExtendedData>
  <Data name="Name"><displayName>Name</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="description"><displayName>description</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="timestamp"><displayName>timestamp</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="begin"><displayName>begin</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="end"><displayName>end</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="altitudeMode"><displayName>altitudeMode</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="tessellate"><displayName>tessellate</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="extrude"><displayName>extrude</displayName><value><![CDATA[0]]></value></Data>
  <Data name="visibility"><displayName>visibility</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="drawOrder"><displayName>drawOrder</displayName><value><![CDATA[NULL]]></value></Data>

```

```

<Data name="icon"><displayName>icon</displayName><value><![CDATA[NULL]]></value></Data>
<Data name="ID"><displayName>ID</displayName><value><![CDATA[4434]]></value></Data>
<Data name="GlideNumbe"><displayName>GlideNumbe</displayName><value><![CDATA[NULL]]></value></Data>
<Data name="Country"><displayName>Country</displayName><value><![CDATA[USA]]></value></Data>
<Data name="OtherCount"><displayName>OtherCount</displayName><value><![CDATA[NULL]]></value></Data>
<Data name="long"><displayName>long</displayName><value><![CDATA[-122.457654]]></value></Data>
<Data name="lat"><displayName>lat</displayName><value><![CDATA[39.623615]]></value></Data>
<Data name="Area"><displayName>Area</displayName><value><![CDATA[107407.257465]]></value></Data>
<Data name="Began"><displayName>Began</displayName><value><![CDATA[2017/01/04]]></value></Data>
<Data name="Ended"><displayName>Ended</displayName><value><![CDATA[2017/01/05]]></value></Data>
<Data name="Validation"><displayName>Validation</displayName><value><![CDATA[News]]></value></Data>
<Data name="Dead"><displayName>Dead</displayName><value><![CDATA[0]]></value></Data>
<Data name="Displaced"><displayName>Displaced</displayName><value><![CDATA[0]]></value></Data>
<Data name="MainCause"><displayName>MainCause</displayName><value><![CDATA[Heavy Rain]]></value></Data>
<Data name="Severity"><displayName>Severity</displayName><value><![CDATA[1.5]]></value></Data>
</ExtendedData>
<styleUrl>#style2</styleUrl>
<MultiGeometry>
    <Polygon>
        <outerBoundaryIs>
            <LinearRing>
                <tessellate>1</tessellate>
                <coordinates>
                    -122.012272,37.317161,0.00000
                    -122.358681,37.168575,0.00000
                    -123.150473,38.309462,0.00000
                    -123.447395,39.003822,0.00000
                    -123.744317,39.896527,0.00000
                    -124.387647,40.194369,0.00000
                    -124.090725,40.640498,0.00000
                    -124.387647,42.078655,0.00000
                    -121.863811,41.829455,0.00000
                    -120.527662,38.308144,0.00000
                    -122.012272,37.317161,0.00000
                </coordinates>
            </LinearRing>
        </outerBoundaryIs>
    </Polygon>
</MultiGeometry>
</Placemark>

<Placemark>
<TimeSpan>
    <begin>2017-01-05</begin>
    <end>2017-01-21</end>
</TimeSpan>
<name>NULL</name>
<description></description>
<ExtendedData>
    <Data name="Name"><displayName>Name</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="description"><displayName>description</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="timestamp"><displayName>timestamp</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="begin"><displayName>begin</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="end"><displayName>end</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="altitudeMode"><displayName>altitudeMode</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="tessellate"><displayName>tessellate</displayName><value><![CDATA[-1]]></value></Data>
    <Data name="extrude"><displayName>extrude</displayName><value><![CDATA[0]]></value></Data>
    <Data name="visibility"><displayName>visibility</displayName><value><![CDATA[-1]]></value></Data>
    <Data name="drawOrder"><displayName>drawOrder</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="icon"><displayName>icon</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="ID"><displayName>ID</displayName><value><![CDATA[4435]]></value></Data>
    <Data name="GlideNumbe"><displayName>GlideNumbe</displayName><value><![CDATA[NULL]]></value></Data>
    <Data name="Country"><displayName>Country</displayName><value><![CDATA[Brazil]]></value></Data>
    <Data name="OtherCount"><displayName>OtherCount</displayName><value><![CDATA[Argentina]]></value></Data>
    <Data name="long"><displayName>long</displayName><value><![CDATA[-55.551237]]></value></Data>
    <Data name="lat"><displayName>lat</displayName><value><![CDATA[-29.409362]]></value></Data>
    <Data name="Area"><displayName>Area</displayName><value><![CDATA[386846.737648]]></value></Data>
    <Data name="Began"><displayName>Began</displayName><value><![CDATA[2017/01/05]]></value></Data>
    <Data name="Ended"><displayName>Ended</displayName><value><![CDATA[2017/01/21]]></value></Data>
    <Data name="Validation"><displayName>Validation</displayName><value><![CDATA[News]]></value></Data>
    <Data name="Dead"><displayName>Dead</displayName><value><![CDATA[0]]></value></Data>
    <Data name="Displaced"><displayName>Displaced</displayName><value><![CDATA[0]]></value></Data>
    <Data name="MainCause"><displayName>MainCause</displayName><value><![CDATA[Heavy Rain]]></value></Data>
    <Data name="Severity"><displayName>Severity</displayName><value><![CDATA[1.0]]></value></Data>
</ExtendedData>
<styleUrl>#style2</styleUrl>
<MultiGeometry>

```

```

<Polygon>
  <outerBoundaryIs>
    <LinearRing>
      <tessellate>1</tessellate>
      <coordinates>
        -50.157155,-29.908609,0.00000
        -52.829453,-32.783281,0.00000
        -57.184308,-30.500116,0.00000
        -60.747372,-27.423964,0.00000
        -60.94532,-26.035444,0.00000
        -54.808933,-26.831911,0.00000
        -50.157155,-29.908609,0.00000
      </coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
</MultiGeometry>
</Placemark>

<Placemark>
<TimeSpan>
  <begin>2017-01-16</begin>
  <end>2017-01-21</end>
</TimeSpan>
<name>NULL</name>
<description></description>
<ExtendedData>
  <Data name="Name"><displayName>Name</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="description"><displayName>description</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="timestamp"><displayName>timestamp</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="begin"><displayName>begin</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="end"><displayName>end</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="altitudeMode"><displayName>altitudeMode</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="tessellate"><displayName>tessellate</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="extrude"><displayName>extrude</displayName><value><![CDATA[0]]></value></Data>
  <Data name="visibility"><displayName>visibility</displayName><value><![CDATA[-1]]></value></Data>
  <Data name="drawOrder"><displayName>drawOrder</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="icon"><displayName>icon</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="ID"><displayName>ID</displayName><value><![CDATA[4436]]></value></Data>
  <Data name="GlideNumbe"><displayName>GlideNumbe</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="Country"><displayName>Country</displayName><value><![CDATA[Philippines]]></value></Data>
  <Data name="OtherCount"><displayName>OtherCount</displayName><value><![CDATA[NULL]]></value></Data>
  <Data name="long"><displayName>long</displayName><value><![CDATA[125.026807]]></value></Data>
  <Data name="lat"><displayName>lat</displayName><value><![CDATA[8.085715]]></value></Data>
  <Data name="Area"><displayName>Area</displayName><value><![CDATA[46236.6931556]]></value></Data>
  <Data name="Began"><displayName>Began</displayName><value><![CDATA[2017/01/16]]></value></Data>
  <Data name="Ended"><displayName>Ended</displayName><value><![CDATA[2017/01/21]]></value></Data>
  <Data name="Validation"><displayName>Validation</displayName><value><![CDATA[News]]></value></Data>
  <Data name="Dead"><displayName>Dead</displayName><value><![CDATA[4]]></value></Data>
  <Data name="Displaced"><displayName>Displaced</displayName><value><![CDATA[13559]]></value></Data>
  <Data name="MainCause"><displayName>MainCause</displayName><value><![CDATA[Heavy Rain]]></value></Data>
  <Data name="Severity"><displayName>Severity</displayName><value><![CDATA[1.0]]></value></Data>
</ExtendedData>
<styleUrl>#style2</styleUrl>
</MultiGeometry>
<Polygon>
  <outerBoundaryIs>
    <LinearRing>
      <tessellate>1</tessellate>
      <coordinates>
        126.610391,8.084969,0.00000
        124.235015,6.896087,0.00000
        123.443223,7.689869,0.00000
        124.432963,8.581928,0.00000
        126.016547,9.275343,0.00000
        126.610391,8.084969,0.00000
      </coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
</MultiGeometry>
</Placemark>
</Folder>
</kml>

```

12.10 Mapbox

Mapbox GL JS interne Stilisierung

```

<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8' />
  <title>Add GeoJSON Flood Polygons</title>
  <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />
  <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.js'></script>
  <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.css' rel='stylesheet' />
  <style>
    body { margin:0; padding:0; }
    #map { position:absolute; top:0; bottom:0; width:100%; }
  </style>
</head>
<body>

<div id='map'></div>
<script>
mapboxgl.accessToken
'pk.eyJ1IjoiY2xrMTg0IiwiaSI6ImNqcWlkZDRzMzBkZmc0OW55eDR4cTM4YXQifQ.zPkBQsqBwXVfMZLlsRq7AQ';
var map = new mapboxgl.Map({
  container: 'map',
  style: 'mapbox://styles/mapbox/dark-v9',
  center: [7.5, 50.5],
  zoom: 5
});

map.on('load', function () {

  map.addLayer({
    'id': 'maine',
    'type': 'fill',
    'source': {
      'type': 'geojson',
      'data': {
        'type': 'FeatureCollection',
        'name': "Flood_2017_GEOJSON",
        'crs': { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::4030" } },
        'features': [
          {
            'type': 'Feature',
            'properties': {
              "ID": 4433,
              "GlideNumbe": null,
              "Country": "Germany",
              "OtherCount": null,
              "long": 9.583276,
              "lat": 54.705274,
              "Area": 18991.845394600001,
              "Began": "2017-01-02",
              "Ended": "2017-01-05",
              "Validation": "News",
              "Dead": 0,
              "Displaced": 0,
              "MainCause": "Winter Storm Axel",
              "Severity": 1.5
            },
            'geometry': {
              'type': 'MultiPolygon',
              'coordinates': [ [ [
                [ 13.532608, 54.306792 ],
                [ 13.33466, 53.662268 ],
                [ 10.414927, 53.415802 ],
                [ 9.425187, 54.457616 ],
                [ 9.227239, 55.251098 ],
                [ 9.524161, 55.994746 ],
                [ 9.920057, 55.944961 ],
                [ 9.623135, 54.953381 ],
                [ 10.019031, 54.407731 ],
                [ 10.761336, 54.109839 ],
                [ 11.058258, 53.911344 ],

```

```

        [ 12.493381, 54.009796 ],
        [ 13.532608, 54.306792 ]
    ] ]
    }
},
{
  "type": "Feature",
  "properties": {
    "ID": 4487,
    "GlideNumbe": null,
    "Country": "Chile",
    "OtherCount": null,
    "long": -70.99118,
    "lat": -33.418779,
    "Area": 71430.431284100006,
    "Began": "2017-06-14",
    "Ended": "2017-06-20",
    "Validation": "News",
    "Dead": 4,
    "Displaced": 50,
    "MainCause": "Heavy Rain",
    "Severity": 1.0
  },
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [ [ [
      [ -70.248875, -31.187761 ],
      [ -70.347849, -30.691847 ],
      [ -70.149901, -32.477059 ],
      [ -70.050927, -33.766357 ],
      [ -70.644771, -36.146209 ],
      [ -71.931433, -35.550525 ],
      [ -71.733485, -34.360549 ],
      [ -71.337589, -30.691349 ],
      [ -70.248875, -31.187761 ]
    ] ] ]
  }
},
{
  "type": "Feature",
  "properties": {
    "ID": 4491,
    "GlideNumbe": null,
    "Country": "France",
    "OtherCount": null,
    "long": 3.338286,
    "lat": 45.485499,
    "Area": 45514.498247099997,
    "Began": "2017-06-13",
    "Ended": "2017-06-15",
    "Validation": "News",
    "Dead": 1,
    "Displaced": 160,
    "MainCause": "Torrential Rain",
    "Severity": 2.0
  },
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [ [ [
      [ 5.070331, 45.137548 ],
      [ 2.497008, 44.246285 ],
      [ 1.606242, 46.230191 ],
      [ 4.278539, 46.724713 ],
      [ 5.070331, 45.137548 ]
    ] ] ]
  }
}
]
},
'layout': {},
'paint': {
  'fill-color': [
    'interpolate',
    ['linear'],
    ['number', ['get', 'Severity']],
    1.0, 'rgb(246, 249, 24)'
  ]
}

```

```

        1.5, 'rgb(216, 0, 0)',
        2.0, 'rgb(229, 0, 206)'
    ],
    'fill-opacity': 0.8
  }
});
});
</script>

</body>
</html>

```

Mapbox GL JS FadeIn

```

<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8' />
  <title>Add GeoJSON Flood Polygons</title>
  <meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />
  <script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.js'></script>
  <link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.52.0/mapbox-gl.css' rel='stylesheet' />
  <style>
    body { margin:0; padding:0; }
    #map { position:absolute; top:0; bottom:0; width:100%; }
  </style>
</head>
<body>

<div id='map'></div>
<script>
mapboxgl.accessToken
'pk.eyJ1IjoiY2xMTg0fiwiYSI6ImNqcWlkDRzMzBkZmc0OW55eDR4cTM4YXQifQ.zPkBQsqBwXVfMZLlRq7AQ';
var map = new mapboxgl.Map({
  container: 'map',
  style: 'mapbox://styles/mapbox/dark-v9',
  center: [7.5, 50.5],
  zoom: 5
});

map.on('load', function () {

  map.addLayer({
    'id': 'maine',
    'type': 'fill',
    'source': {
      'type': 'geojson',
      'data': {
        'type': 'FeatureCollection',
        'name': "Flood_2017_GEOJSON",
        'crs': { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::4030" } },
        'features': [
          {
            'type': 'Feature',
            "properties": {
              "ID": 4433,
              "GlideNumbe": null,
              "Country": "Germany",
              "OtherCount": null,
              "long": 9.583276,
              "lat": 54.705274,
              "Area": 18991.845394600001,
              "Began": "2017-01-02",
              "Ended": "2017-01-05",
              "Validation": "News",
              "Dead": 0,
              "Displaced": 0,
              "MainCause": "Winter Storm Axel",
              "Severity": 1.5
            },
            'geometry': {
              'type': 'MultiPolygon',
              'coordinates': [ [ [
                [ 13.532608, 54.306792 ],

```

```

        [ 13.33466, 53.662268 ],
        [ 10.414927, 53.415802 ],
        [ 9.425187, 54.457616 ],
        [ 9.227239, 55.251098 ],
        [ 9.524161, 55.994746 ],
        [ 9.920057, 55.944961 ],
        [ 9.623135, 54.953381 ],
        [ 10.019031, 54.407731 ],
        [ 10.761336, 54.109839 ],
        [ 11.058258, 53.911344 ],
        [ 12.493381, 54.009796 ],
        [ 13.532608, 54.306792 ]
    ] ]
    }
},
{
  "type": "Feature",
  "properties": {
    "ID": 4487,
    "GlideNumbe": null,
    "Country": "Chile",
    "OtherCount": null,
    "long": -70.99118,
    "lat": -33.418779,
    "Area": 71430.431284100006,
    "Began": "2017-06-14",
    "Ended": "2017-06-20",
    "Validation": "News",
    "Dead": 4,
    "Displaced": 50,
    "MainCause": "Heavy Rain",
    "Severity": 1.0
  },
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [ [ [
      [ -70.248875, -31.187761 ],
      [ -70.347849, -30.691847 ],
      [ -70.149901, -32.477059 ],
      [ -70.050927, -33.766357 ],
      [ -70.644771, -36.146209 ],
      [ -71.931433, -35.550525 ],
      [ -71.733485, -34.360549 ],
      [ -71.337589, -30.691349 ],
      [ -70.248875, -31.187761 ]
    ] ] ]
  }
},
{
  "type": "Feature",
  "properties": {
    "ID": 4491,
    "GlideNumbe": null,
    "Country": "France",
    "OtherCount": null,
    "long": 3.338286,
    "lat": 45.485499,
    "Area": 45514.498247099997,
    "Began": "2017-06-13",
    "Ended": "2017-06-15",
    "Validation": "News",
    "Dead": 1,
    "Displaced": 160,
    "MainCause": "Torrential Rain",
    "Severity": 2.0
  },
  "geometry": {
    "type": "MultiPolygon",
    "coordinates": [ [ [
      [ 5.070331, 45.137548 ],
      [ 2.497008, 44.246285 ],
      [ 1.606242, 46.230191 ],
      [ 4.278539, 46.724713 ],
      [ 5.070331, 45.137548 ]
    ] ] ]
  }
}
}

```

```
        }
    },
    'layout': {},
    'paint': {
        'fill-color': [
            'interpolate',
            ['linear'],
            ['number', ['get', 'Severity']],
            1.0, 'rgb(246, 249, 24)',
            1.5, 'rgb(216, 0, 0)',
            2.0, 'rgb(229, 0, 206)'
        ],
        'fill-opacity': 0,
        'fill-opacity-transition': {duration: 2000}
    }
});

// so after the map is initialized we call a function which rund as example after 3 seconds and set the opacity value to 0.8
// the circle-opacity-transition is set as 2000, what means that changes to this value takes 2 seconds
setTimeout(function() {
    map.setPaintProperty('maine', 'fill-opacity', 0.8);
},
    3000
);
</script>

</body>
</html>
```

Erklärung

Ich versichere:

- dass ich die Masterarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- dass alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Publikationen entnommen sind, als solche kenntlich gemacht sind.
- dass ich dieses Masterarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Datum

Unterschrift