



universität
wien

Diplomarbeit

Titel der Diplomarbeit

**„Vektordaten am taktilen Hyperglobus – über
Verfahren zur Aufbereitung und Visualisierung
geographischer Vektordatensätze am
sphärischen Display“**

Verfasser

Johannes LIEM

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, im Oktober 2012

Studienkennzahl lt. Studienblatt:

A 455

Studienrichtung lt. Studienblatt:

Kartographie und Geoinformation

Betreuer:

Ass.-Prof. Mag. Dr. Andreas RIEDL

Den sphärischen Displays dieser Erde.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Abbildungsverzeichnis	iv
Vorwort	vi
1 Einleitung.....	1
1.1 Ein Herantasten an die Problemstellung	1
1.2 Wissenschaftliche Fragestellungen	3
1.3 Aufbau der Arbeit	3
1.4 Zielsetzung	4
1.5 Methoden.....	5
2 Sphärische Displays und taktile Hypergloben.....	6
2.1 Die Stellung taktiler Hypergloben innerhalb der Globenfamilie	6
2.2 Das System „Taktiler Hyperglobus“	8
2.2.1 Eingabe: Das Basismedium.....	10
2.2.2 Ausgabe: Das Sphärische Display.....	11
2.2.3 Verarbeitung: Die Globensoftware	13
2.2.3.1 ARC Science Simulations.....	13
2.2.3.2 NOAA Science On a Sphere.....	15
2.2.3.3 Global Imagination	15
2.2.3.4 Science On a Snow Globe	15
2.2.3.5 HRG/Globocess	17
2.2.3.6 Sphere (Microsoft).....	17
2.2.4 Anwendungsbereiche und Themenvielfalt.....	18
3 Geographische Vektordaten und deren kartographische Darstellung	20
3.1 Das Vektormodell und dessen Geometrien.....	20
3.1.1 Simple Feature Geometrien.....	22
3.2 Kartographische Darstellung der Geometrien.....	27
3.2.1 Kartographische Gestaltungsmittel	28
3.3 Globale Vektordatensätze	30
3.3.1 Dateiformate.....	30
3.3.1.1 Keyhole Markup Language (KML).....	30
3.3.1.2 ESRI Shapefile Format (SHP)	31
3.3.2 Quellen	32
3.3.2.1 Natural Earth Data	32
3.3.2.2 Vector Map (VMAP).....	32
3.3.2.3 Globale Sachdaten	33
4 Stand der Forschung.....	34
4.1 Taktile digitale Globen.....	34
4.1.1 Die Wiedergabe von Punktgeometrien und Namengut.....	34
4.1.1.1 Breitenabhängige horizontale Streckung	34
4.1.1.2 Geodätische Puffer (ArcGIS).....	36
4.1.1.3 Visualisierung auf der Tangentialebene	38
4.1.2 Die Wiedergabe von Linien und Flächen.....	39
4.2 Virtuelle digitale Globen/Earthbrowser	39
4.2.1 Punktgeometrien.....	40
4.2.2 Liniengeometrien	40
4.2.3 Flächengeometrien	42

5 Die Wiedergabe von Vektordaten am sphärischen Display	45
5.1 Kriterien zur Methodenklassifizierung der Vektordatenintegration	45
5.1.1 Implementierung der Visualisierung: intern vs. extern	45
5.1.2 Visualisierungszeitpunkt: synchron vs. asynchron	47
5.1.3 Objektraum der Visualisierung: 2D vs. 3D	47
5.1.4 Ausgabe- bzw. Speicherformat der Visualisierung	49
5.1.5 Beispiele	50
5.2 Dreieinhalb Ansätze unter der Lupe	51
5.2.1 Ansatz 1: OpenGL ES Shader und sphärische Trigonometrie (2D/3D)	53
5.2.2 Ansatz 2a: Ogre3D ManualObject (3D)	54
5.2.3 Ansatz 2b: Ogre3D Mesh (3D)	54
5.2.4 Ansatz 3: PovRay und CSG (3D)	55
5.3 Aufbereitung von Vektordaten für die Wiedergabe am sphärischen Display und die damit verbundenen Anforderungen an die Daten	55
5.3.1 Geometrien einlesen	56
5.3.2 Ebenen identifizieren/Selektion	57
5.3.3 Koordinatensysteme und Projektionen	57
5.3.4 Maßstab/Generalisierung	58
5.3.5 Verfeinerung/Segmentierung	59
5.3.6 Stil der Signaturen	61
5.3.7 Thematische und inhaltliche Anforderungen	61
5.4 Die Visualisierung von Vektordaten am sphärische Display	62
5.4.1 Wesentliche Unterschiede von 2D und 3D Ansätzen	62
5.4.1.1 Zeichenträger	62
5.4.1.2 Verzerrungen und Nähte	63
5.4.1.3 Mathematische Grundlagen	63
5.4.2 Visualisierungsgeometrie durch Constructive Solid Geometry	64
5.4.3 Visualisierungsgeometrie durch Meshgenerierung	66
6 Technische Implementierungen	68
6.1 Punktgeometrien	68
6.1.1 OpenGL ES / Sphärische Trigonometrie	68
6.1.2 Ogre3D / Vektorrechnung	70
6.1.3 PovRay / CSG	71
6.2 Liniengeometrien	74
6.2.1 OpenGL ES / Sphärische Trigonometrie	74
6.2.2 Ogre3D / Vektorgeometrie	76
6.2.3 PovRay / CSG	77
6.2.4 Verbindungen zwischen Liniensegmenten	79
6.2.5 Groß- und Kleinkreise (Meridiane und Parallelen)	81
6.3 Flächengeometrien	82
6.3.1 Beliebige (sphärische) Dreiecke (PovRay)	83
6.3.2 (Sphärische) Polygone und deren Triangulierung (PovRay)	83
6.4 Fallbeispiele	86
7 Diskussion	87
8 Zusammenfassung und Ausblick	92
Quellenverzeichnis	95
Anhang	98
Anhang A: Vulkanstandorte	98
Anhang B: Städte	100

Anhang C:	Gradnetz	102
Anhang D:	Flugroute über Pol	104
Anhang E:	Schiffstrack	105
Anhang F:	Staatsgrenzen.....	106
Anhang G:	Polygontriangulierung mit PovRay.....	108
Anhang H:	Kurzfassung	109
Anhang I:	Abstract	110
Anhang J:	Curriculum Vitae.....	111
Anhang K:	Eidesstattliche Erklärung	113

Abbildungsverzeichnis

Abbildung 1: Taktiler Hyperglobus mit Touchscreen-Interface	1
Abbildung 2: Punktsignaturen ohne Berücksichtigung der Verzerrungseigenschaften der Plattkarte	2
Abbildung 3: Punktsignaturen unter Berücksichtigung der Verzerrungseigenschaften der Plattkarte	2
Abbildung 4: Die Globenfamilie	7
Abbildung 5: Das System „Taktiler Hyperglobus“	9
Abbildung 6: Schemata unterschiedlicher Projektionsverfahren bei sphärischen Displays.....	12
Abbildung 7: Funktionsweise der Globensoftware von ARC Science	14
Abbildung 8: Die Funktionsweise der OmniSuite Render Engine	16
Abbildung 9: Funktionsweise des Vertex Shaders von Microsofts Sphere Projekt	18
Abbildung 10: Das Vektormodell am Beispiel von Liniengeometrien	21
Abbildung 11: Beispiele für LineStrings	23
Abbildung 12: Beispiele für Polygone.....	24
Abbildung 13: Objekte, die nicht als einzelnes Polygon dargestellt werden können	25
Abbildung 14: Liniengeometrien und entsprechende linienhafte Signaturen.....	27
Abbildung 15: Punktsignaturen und Beschriftung am Beispiel der Kontinentaldrift.....	35
Abbildung 16: Geodätisches Puffern von Punktsignaturen mit ArcGIS (Detailansicht)	36
Abbildung 17: Geodätisches Puffern mit ArcGIS zur Wiedergabe von kreisförmigen Signaturen	37
Abbildung 18: Punktsignaturen mit Geodätischen Puffern (ArcGIS)	38
Abbildung 19: KML Interpolationsmodi für die Elemente LineString und LinearRing	40
Abbildung 20: KML Interpolationsmodi für Polygone	42
Abbildung 21: Klassifizierungsschema zur Vektordatenintegration am sphärischen Display	45
Abbildung 22: Schema der internen und synchronen Vektordatenintegration	46
Abbildung 23: Schema der externen synchronen bzw. asynchronen Vektordatenintegration	47

Abbildung 24: Objektraum der Vektordatenintegration: 2D vs. 3D	48
Abbildung 25: Klassifizierungsbeispiele der Vektordatenintegration	50
Abbildung 26: Klassifizierung unterschiedlicher Ansätze der Vektordatenintegration.	51
Abbildung 27: Prozesspipeline der Vektordatenvisualisierung für das sphärische Display.....	52
Abbildung 28: Schritte der Datenverarbeitung und -aufbereitung	56
Abbildung 29: Schema der Verfeinerung/Segmentierung von Vektorgeometrien	60
Abbildung 30: Mengenoperationen für CSG in PovRay.....	65
Abbildung 31: Zeichenprimitive für die Wiedergabe durch den Grafikprozessor.....	67
Abbildung 32: Große quadratische Signatur am Außenrand der Globenabbildung	69
Abbildung 33: Hexagon als Signatur in Ogre3D	71
Abbildung 34: Generierung einer Darstellungsgeometrie in PovRay mit CSG.....	73
Abbildung 35: Grafische Variationsmöglichkeiten von Punktsignaturen (PovRay)	74
Abbildung 36: Darstellung von Liniensignaturen auf der Globusabbildung (Ausschnitt)	75
Abbildung 37: Liniengeometrien mit TRIANGLE_STRIPs visualisiert	76
Abbildung 38: Generierung von Liniensegmenten bzw. LineStrings mit PovRay und CSG	78
Abbildung 39: Grafische Variationsmöglichkeiten von Liniensignaturen (PovRay)	79
Abbildung 40: Unterschiedliche Verbindungsarten von Liniensegmenten	80
Abbildung 41: Beispiel für Verbindungstypen als Dreiecksnetz	80
Abbildung 42: Verbindung von Liniensegmenten bei Ansatz 3 (Ogre3D).....	81
Abbildung 43: Generierung eines sphärischen Dreiecks (PovRay)	83
Abbildung 44: Triangulierung eines sphärischen Polygons mittels Ear Clipping (PovRay).....	84
Abbildung 45: Graphische Variation von Flächensignaturen (mit Löchern) in PovRay	86
Abbildung 46: Bewertung der besprochenen Ansätze	88

Vorwort

Vor einigen Jahren bin ich in der Lehrveranstaltung Multimedia Technologie und Geoinformation das erste Mal mit dem Thema Taktile Hypergloben und Sphärische Displays in Berührung gekommen. Mein Interesse für diese magischen Kugeln gipfelt nun in dieser schönen Diplomarbeit.

Ganz gut sind auch: die Hyperglobe Research Group, Universität Wien und die Cartography and Geovisualization Group, Oregon State University. Yeah!

Das war's dann auch schon, viel Spaß beim Lesen!

DA GE

HTS L

ANG !!

Vielen Dank euch allen!

Ja, euch allen!

Johannes

L.

1 Einleitung

Das sphärische Display, die Kugeloberfläche als „Monitor“, dient als Globenkörper taktiler Hypergloben. Ein taktiler Hyperglobus (Abbildung 1) ist eine moderne, digitale Präsentationsform, mit der unterschiedlichste globale Zusammenhänge und Phänomene wiedergegeben werden können. Sie werden derzeit vor allem in Museen, Schulen und Forschungseinrichtungen eingesetzt.



Abbildung 1: Taktiler Hyperglobus mit Touchscreen-Interface
(Quelle: [GLO-12aa])

Seit 2005 forscht die Hyperglobe Research Group (HRG) des Instituts für Geographie und Regionalforschung der Universität Wien in dem Bereich „Digitale Globen“.

Neben der theoretischen Auseinandersetzung arbeitet die HRG auch anwendungsorientiert. So wurde bspw. das Softwarepaket OmniSuite, eine Autoren- und Abspielplattform für sphärische Displays, von der HRG entwickelt. Diese Diplomarbeit reiht sich in die Forschung der HRG mit ein.

1.1 Ein Herantasten an die Problemstellung

Gegenwärtige Systeme taktiler Hypergloben verwenden sehr oft quadratische Platkarten (Rasterdaten) als Basis für die Visualisierung globaler Thematiken am sphärischen Display (Kapitel 2.2.1).

Geographische Vektordatensätze bzw. deren Visualisierung können als thematische Overlays am taktilen Hyperglobus verwendet werden. Hierbei ist die Berücksichtigung des sehr kleinen und globalen Maßstabes wichtig. Punkthafte Signaturen repräsentieren bspw. GPS-Koordinaten oder größere Städte. Flugverbindungen, Staatsgrenzen oder ein Gradnetz werden anhand von Band- und Liniensignaturen dargestellt. Flächengeometrien wie Staatsflächen oder Abgrenzungen von Habitaten werden mittels flächenhafter Signaturen visualisiert.

Nun könnte jeder geokodierter Vektordatensatz selbstverständlich mit einem Geoinformationssystem (GIS) als Platkarte aufbereitet werden und somit als Inputraster für den taktilen Hyperglobus dienen. Solche Platkarten sind jedoch lediglich für eine zweidimensionale Darstellung und nicht für die Wiedergabe am digitalen Globus geeignet. Die quadratische Platkarte weist nämlich mit zunehmender Breite eine zunehmende Verzerrung (Kehrwert des Kosinus der Breite) auf, die bei der

Visualisierung mit gegenwärtigen GISen bei Bedarf nur in Ansätzen berücksichtigt wird. (Kapitel 4.1.1.2)

Die nachstehenden Abbildungen veranschaulichen diesen Sachverhalt anhand von Punktgeometrien, wobei die Plattkarte mittels Texture Mapping auf die Kugel projiziert wird.

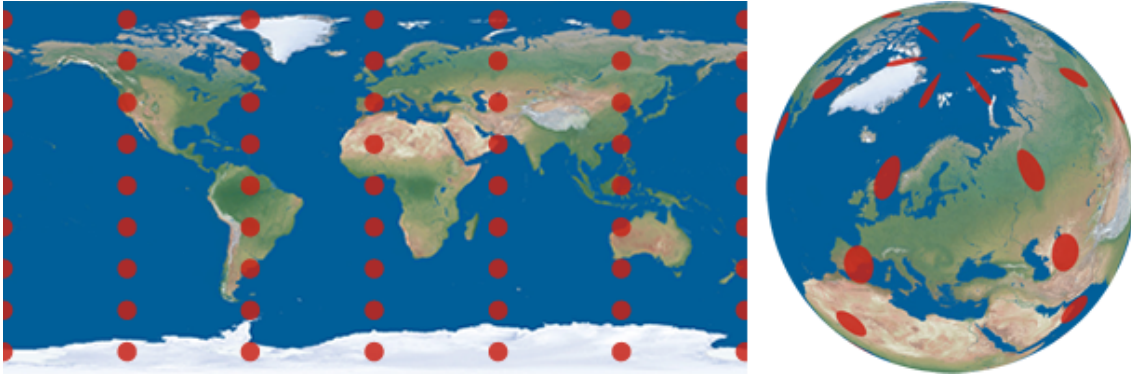


Abbildung 2: Punktsignaturen ohne Berücksichtigung der Verzerrungseigenschaften der Plattkarte

Punktsignaturen ohne Berücksichtigung der Verzerrungseigenschaften der Plattkarte (links); Globenansicht (die Plattkarte wurde mittels Texture Mapping auf die Kugel projiziert), bei der die Punktsignaturen deformiert dargestellt werden (rechts).

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

Bei Abbildung 2 werden die Punktsignaturen bei der Globendarstellung mit zunehmender Breite immer stärker deformiert. Folglich müssen die Kreise auf der Plattkarte in Abhängigkeit der Breite bereits verzerrt werden, damit sie auf der Kugel eine Kreisform erhalten. Die Abbildung 3 zeigt eine Tissot'sche Indikatrix, welche die Verzerrungseigenschaften einer Plattkarte wiedergibt.

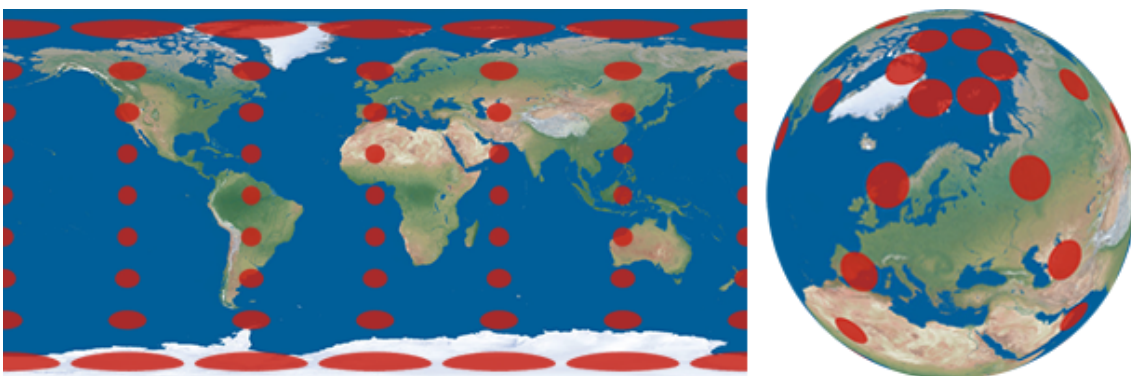


Abbildung 3: Punktsignaturen unter Berücksichtigung der Verzerrungseigenschaften der Plattkarte

Punktsignaturen unter Berücksichtigung der Verzerrungseigenschaften der Plattkarte (= Tissot'sche Indikatrix) (links); Globenansicht, bei der die Punktsignaturen formgetreu dargestellt werden (rechts).

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

Aus den obigen Abbildungen lassen sich weitere Punkte ableiten. Neben den Verzerrungseigenschaften des Basismediums müssen auch „Nähte“ berücksichtigt werden. Bei der Platkarte handelt es sich um eine „Ost-West-Naht“ (Datumsgrenze), bei der Signaturen, je nach deren Lage und Größe, auf „beiden Seiten“ der Platkarte dargestellt werden müssen. Folglich werden nicht nur Flächengeometrien, sondern auch Punkt- und Liniengeometrien flächenhaft visualisiert.

Aufgrund dieser Überlegungen ergeben sich zwei unterschiedliche Herangehensweisen für die Wiedergabe von Vektordaten am sphärischen Display:

- Die Einzeichnung der Punkt-, Linien- und Flächendaten in eine ebene Basisabbildung (Platkarte, CubeMap) unter Berücksichtigung der projektionsbedingten Verzerrungen und der „Nähte“ („Ost-West-Naht“ bei Platkarte; „Würfelkanten“ bei CubeMap).
- Die Visualisierung der Geometrien im virtuellen dreidimensionalen Raum entlang einer Kugeloberfläche, wobei Ansätze dieser Methode an ein bestehendes Wiedergabesystem (z.B. OmniSuite) zu koppeln sind bzw. die Möglichkeit bieten müssen, als Zwischenschritt das dementsprechende Inputformat (bspw. Platkarten) generieren zu können.

1.2 Wissenschaftliche Fragestellungen

Es ergeben sich nun folgende zentrale Forschungsfragen:

- Welche Methoden und Lösungsansätze sind für die Wiedergabe von Punkt-, Linien- bzw. Flächengeometrien am sphärischen Display geeignet? Welche Probleme und Hindernisse müssen bei der Visualisierung von Vektordatensätzen berücksichtigt werden?
- Welche Anforderungen sind an Geo-Vektordatensätze als Datenquelle für die Visualisierung am sphärischen Display zu stellen?

1.3 Aufbau der Arbeit

Zur Untersuchung und Beantwortung der Fragestellungen umfasst die Diplomarbeit folgende Inhalte:

Nach dem einleitenden Kapitel, das die Lesenden an die Problemstellung heranführt und den (methodischen) Rahmen der Diplomarbeit behandelt, folgt ein Kapitel, in dem das Forschungsobjekt Taktile Hyperglobus bzw. Sphärisches Display erörtert wird. Es wird auf die unterschiedlichen Aspekte (Software, Hardware) solcher Systeme

eingegangen (Kapitel 2). Anschließend werden (globale) geographische Vektordaten und deren zugrunde liegenden Geometrien, Formate und Quellen in der Geoinformation und Kartographie besprochen (Kapitel 3).

In weiterer Folge (Kapitel 4) werden bestehende Anwendungsbeispiele bzw. Methodenansätze für die Aufbereitung und Wiedergabe von Vektorinformationen am sphärischen Display diskutiert. Hierzu wird beleuchtet, wie Anbieter von taktilen digitalen Globen bzw. deren Software mit Vektordaten umgehen. Des Weiteren wird die Visualisierung von Vektordaten bei Earthbrowsern bzw. virtuellen digitalen Globen evaluiert.

Darauf aufbauend wird in Kapitel 5 eine Klassifizierungsmethode entwickelt, die es erlaubt die unterschiedlichen Verfahrensansätze zu systematisieren. Die Klassifizierung orientiert sich u.a. an den zwei bereits oben erwähnten Herangehensweisen. Es werden mehrere Verfahrensansätze anhand der eingeführten Klassifizierung vorgestellt. Der Prozess der Vektordatenintegration wird in die Datenverarbeitung bzw. -aufbereitung sowie die Datenvisualisierung unterteilt. In diesem Zusammenhang werden neben technischen auch thematische Voraussetzungen und Anforderungen erläutert, die an Vektordatenansätze für die Visualisierung am sphärischen Display zu stellen sind.

Unter der Berücksichtigung des Forschungssettings im Globus-Labor der HRG sowie der gegenwärtigen Soft- und Hardwarebedingungen werden die vorgestellten Methodenansätze aus technischer Sicht erörtert (Kapitel 6). Die Implementierungsmöglichkeiten der Datenvisualisierung werden für Punkt-, Linien- und Flächengeometrien mittels der verschiedenen Ansätze besprochen. Im Anhang befinden sich darüber hinaus konkrete Fallbeispiele zu den einzelnen Methoden.

Schließlich werden die unterschiedlichen Lösungsansätze und Ergebnisse vergleichend besprochen und die oben gestellten Forschungsfragen diskutiert (Kapitel 7).

Das abschließende Kapitel 8 liefert neben einer Zusammenfassung einen Ausblick auf mögliche Weiterentwicklungen, die auf den hier vorgestellten Ausführungen aufbauen können.

1.4 Zielsetzung

Das Ziel der Arbeit ist die Untersuchung verschiedener Möglichkeiten der Vektordatenintegration (KML-Dateien, ESRI Shapedateien) am sphärischen Display. Zur Beantwortung der wissenschaftlichen Fragestellungen werden unterschiedliche Methoden durchbesprochen und in Ansätzen implementiert. Wie können die geographischen Vektordaten visuell korrekt in bestehende Systeme integriert werden?

Für die Diplomarbeit werden geographische Vektordaten auf „Simple Features“ reduziert, und zwar auf die Elemente Punkt, Linie und Fläche. Auch deren Wiedergabe auf dem sphärischen Display beschränkt sich auf einfache, abstrakte Signaturen. Die Möglichkeiten der graphischen Variation der Signaturen sowie deren Kombinationsfähigkeit werden nicht ausgereizt.

1.5 Methoden

Für die technische Umsetzung der Datenvisualisierung wurden Methoden herangezogen, die in der Informatik und Mathematik beheimatet sind. Sphärische Trigonometrie, Vektorrechnung und analytische Geometrie, solide Festkörpergeometrie und die Generierung von Dreiecksnetzen sind Beispiele dafür.

Die unterschiedlichen vorgestellten Verfahren wurden ansatzweise implementiert. Dazu wurden u.a. die Programmier- bzw. Skriptsprachen JavaScript, OpenGL ES Shader Language, C++ und die PovRay eigene Scene Definition Language verwendet. Darüber hinaus wurden unterschiedliche Bibliotheken, wie OGR (Einlesen von Vektordaten) oder Ogre3D (Grafik-Engine), verwendet.

Als Quelle für Vektordaten werden für die unterschiedlichsten Beispiele vor allem Datensätze des Projekts Natural Earth Data (Kapitel 3.3.2.1) verwendet.

Viele der beschreibenden, schematischen Abbildungen wurden mit PovRay gerendert bzw. direkt von den Applikationen exportiert (Ogre3D) oder als Screenshot (OpenGL ES Shader) aufgenommen. Diese Abbildungen und andere Grafiken wurden mit Adobe Illustrator aufbereitet.

Aus kartographischer Sicht fließen vor allem Methoden der Vektordatenvisualisierung bzw. die Erzeugung und Wiedergabe von Signaturen inklusive deren graphischen Variationsmöglichkeiten in die Ausführungen mit ein. Darüber hinaus wird auf Methoden, die in Earthbrowsern Einsatz finden, zurückgegriffen.

2 Sphärische Displays und taktile Hypergloben

Ein sphärisches Display ist als ein Monitorsystem bzw. eine Projektionsfläche in Kugelform zu verstehen, bei dem eine digitale Bildinformation auf die Kugeloberfläche aufgebracht wird. Es liegt also nahe das sphärische Display als Globenkörper zu verwenden und erd- bzw. planetenumspannende Phänomene und Thematiken sowie Abbilder der Planeten selbst damit darzustellen. Das sphärische Display stellt somit die Hardwarekomponente von Globensystemen dar, die hier als taktile Hypergloben bezeichnet werden. Darüber hinaus setzt sich ein taktile Hyperglobus durch eine Globensoftware und vor allem durch die unzähligen dafür aufbereiteten Themen und Inhalte zusammen.

2.1 Die Stellung taktile Hypergloben innerhalb der Globenfamilie

Das Globenverständnis der HRG, das auch hier Anwendung findet, bezieht sich auf ein bereits 2000 von Riedl ausgearbeitetes Konzept, wobei sowohl für analoge als auch für digitale Globen gilt [RIE-00]:

„Ein Globus präsentiert ein maßgebundenes und strukturiertes Modell eines Himmelskörpers (bzw. der scheinbaren Himmelskugel) in seiner unverzerrten dreidimensionalen Ganzheit“.

Für die Klassifizierung der Globenfamilie zieht er als Unterscheidungsmerkmal folgende drei Aspekte der Beschaffenheit eines Globus heran [RIE-11a]:

- Die Art des **Globenbildes**: **analog** oder **digital**
- Die Beschaffenheit des **Globenkörpers**: **materiell** oder **virtuell**
- Die Ausprägung des **Raumes**, in welchem der Globus wiedergegeben wird: **real** oder **virtuell**

Riedl gelangt somit zu einer Terminologie (Abbildung 4), die neben analogen Globen digitale Globen kategorisiert [RIE-00]:

- **Virtuelle Hypergloben (VHG)**: Ein digitales Abbild (der Erde, eines Planeten oder einer sonstigen Darstellung) wird auf einem virtuellen Globenkörper im virtuellen Raum wiedergegeben.
- **Taktile Hypergloben (THG)**: Ein digitales Abbild wird auf einem materiellen Globenkörper im realen Raum wiedergegeben.

- **Hologloben (HG):** Ein digitales Abbild wird auf einem virtuellen Kugelkörper im realen Raum dargestellt.

Mit der Entwicklung digitaler Globen wuchs die Globenfamilie um ein Mitglied, das den Ansprüchen des vorherrschenden schnelllebigen Informationszeitalters gerecht wird. Mit digitalen Globen ist es möglich unterschiedlichste, auch zeitlich animierte Themen mit einem einzigen Produkt zu visualisieren, das im Vergleich zu analogen Globen erweiterte Interaktionsmöglichkeiten mit sich bringt. [RIE-00], [KRI-12]

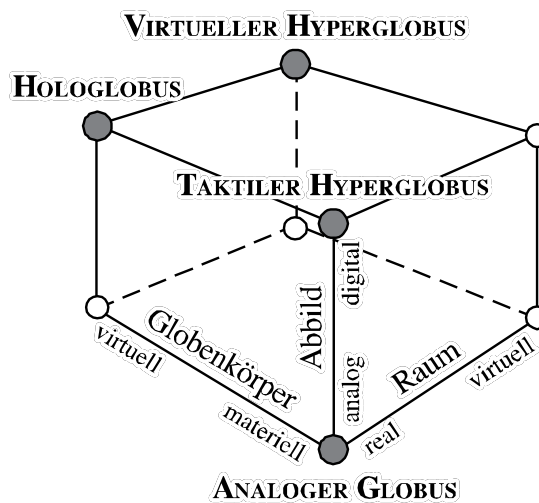


Abbildung 4: Die Globenfamilie

Kategorisierung nach Art des Globenbildes, nach Beschaffenheit des Globenkörpers und nach dem Raum der Visualisierung.

(Quelle: [RIE-00], verändert)

Das Präfix „Hyper“, das den interaktiven und multimedialen Charakter von solchen Globensystemen unterstreicht, verwendet Riedl in Anlehnung an Hypermedia [RIE-00].

In Bezug auf die Verwendung des Wortes „taktile“ weisen Riedl und Kristen [RIE-10] darauf hin, dass bei taktilen Hypergloben – im Gegensatz zu virtuellen Hypergloben – die Möglichkeit besteht den Globenkörper physisch zu berühren. Aber: „*Es ist damit keine haptische Wahrnehmung, also nicht der „aktive Informationsgewinn“ durch den Tastsinn gemeint.*“ Gleichbedeutend wird auch die Bezeichnung materielle Hypergloben verwendet. [RIE-11b]

Nach einer Studie von Hruby et. al. [HRU-09] werden Produkte wie Google Earth und NASA World Wind in der Alltagssprache als „Digitale Globen“ oder „Virtuelle Globen“ bezeichnet. Die Autoren zeigen auf, dass es sich streng genommen – nach der oben angeführten Definition – nicht ausschließlich um digitale Globen handeln kann. Die Globusansicht als kleinmaßstäbigste Darstellung dient hier meist als eine

Schnittstelle zu großmaßstäbigen kartografischen Ausdrucksformen, die nicht mehr die dreidimensionale Ganzheit des Globenkörpers wiedergeben. Treffender sind die Begriffe Earth- oder Geo-Browser: Der Anwender kann durch die unterschiedlichen Darstellungsformen (2D und 3D Ansichten) in unterschiedlichen Maßstabsebenen „browsen“ – ähnlich einem Webbrowser für das Internet. Diese Begriffsauffassungen werden der oben angeführten Definition von Globen gerecht. [HRU-09], [RIE-08]

Auf die Entwicklungsgeschichte digitaler Globen sowie auf Hologloben wird an dieser Stelle nicht weiter eingegangen. Hier bieten u.a. die von Riedl [RIE-11b] zusammengefassten Forschungsergebnisse der HRG einen sehr guten Überblick.

Das System, das hinter taktilen Hypergloben steht, wird im nächsten Abschnitt besprochen. Die Funktionsweise von virtuellen Hypergloben bzw. von Earth-Browsern in Bezug auf geographische Vektordaten ist Inhalt von Kapitel 4.2.

2.2 Das System „Taktiler Hyperglobus“

Einen ersten allgemeinen Überblick über die Funktionsweise von taktilen Hypergloben liefert die Abbildung 5, wobei drei grobe Bereiche unterschieden werden: die Eingabe, die Verarbeitung und die Ausgabe.

Am Anfang des Systems steht als Eingabe (Kapitel 2.2.1) ein Raster-Basismedium als Informationsträger des zu visualisierenden globalen Themas. Der Bereich Verarbeitung (Kapitel 2.2.3) bespricht die Globensoftware, die primär das Basismedium für die Ausgabe am sphärischen Display (Kapitel 2.2.2) aufbereitet.

Das Beispiel in Abbildung 5 (a) verwendet als Basismedium eine quadratische Platkarte. Die Globensoftware berechnet in Abhängigkeit des Ausgabesystems (Abbildung 5 (d)) aus dem Basismedium die benötigte Darstellung (Abbildung 5 (c)), die mit einem Projektor auf den Globenkörper aufgebracht wird. In diesem Beispiel (Abbildung 5 (d)) wird eine spiegelbasierte Innenprojektion mit nur einem Projektor (Single-Beamer) eingesetzt. Weist das sphärische Display diese Spezifikation auf, muss im Verarbeitungsschritt eine vermittelnde Azimutalabbildung von der Globensoftware generiert werden. In dem Beispiel (Abbildung 5 (b)) wird das Basismedium im virtuellen 3D Raum mittels Texture-Mapping auf einen Kugelkörper aufgebracht und in weitere Folge die benötigte Abbildung berechnet. Dieser Vorgang unterscheidet sich je nach Globensoftware bzw. Anbieter von taktilen Hypergloben.

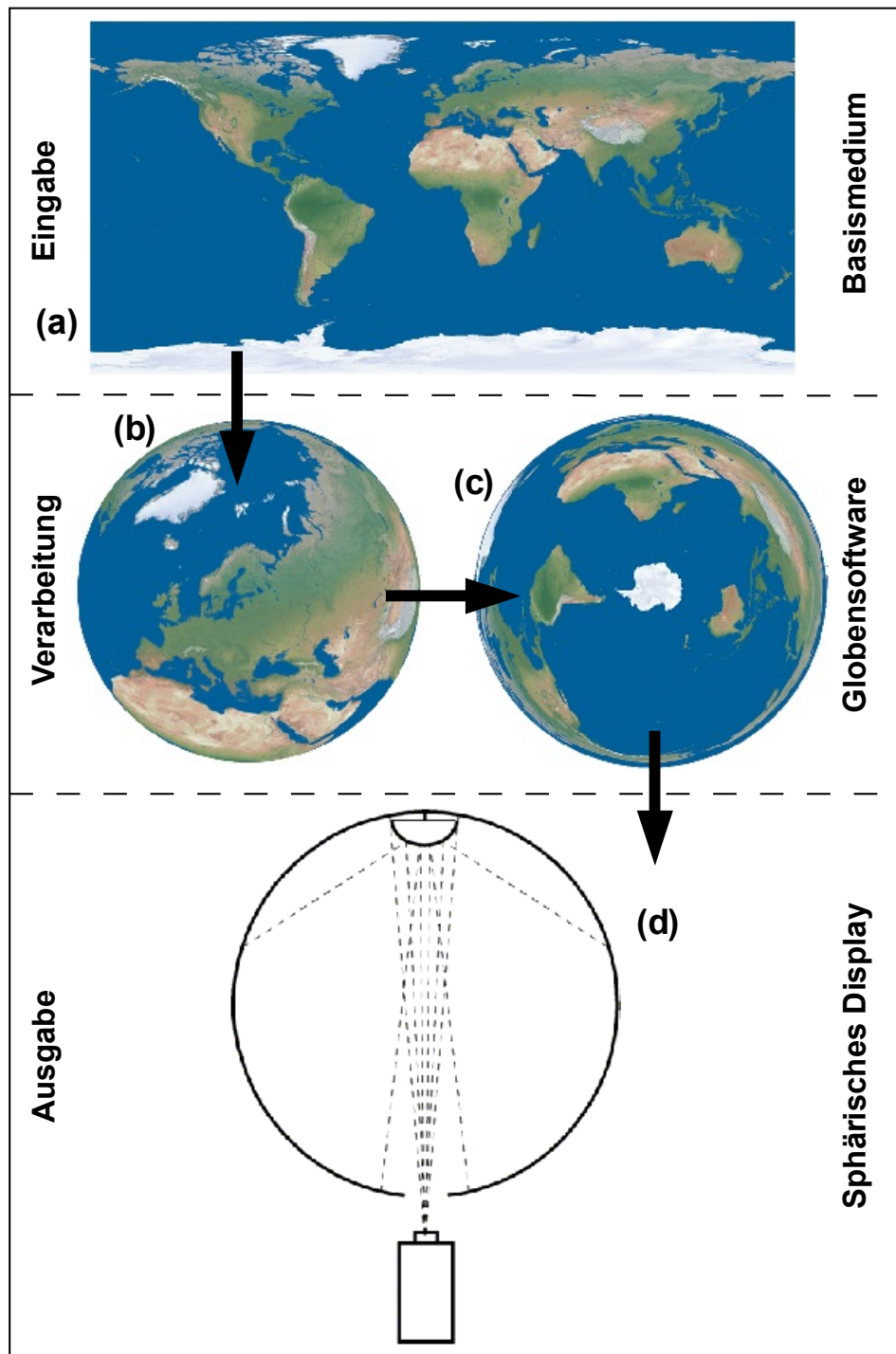


Abbildung 5: Das System „Taktile Hyperglobus“

Unterteilung des Systems in Eingabe, Verarbeitung und Ausgabe – wobei das Basismedium (a) mittels eines globensoftwareabhängigen Verfahrens (b) in jene Projektion (c) umgerechnet wird, die das sphärische Display (d) für die korrekte Wiedergabe der Eingangsdaten benötigt.

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12], schematische Darstellung des sphärischen Displays: [RIE-08])

Eine Liste von Unternehmen, die taktile Hypergloben produzieren bzw. vertreiben, ist bei [RIE-08] zu finden. Exemplarisch seien hier ARC Science Simulations [ARC-12a],

Global Imagination (GI) [GIM-12], NOAA Science On a Sphere (SOS) [SOS-12a], sowie der Kooperationspartner der HRG Globoccess AG [GLO-12a] genannt. Neben einer Globensoftware und sphärischen Displays bieten diese Unternehmen unterschiedlichste bereits kartographisch aufbereitete Themen von globaler Relevanz an (vgl. Themen Bibliothek von Globoccess, [GLO-12b]). Die Basismedien (quadratische Platkarten) werden in Form von Rasterbildern, bei animierten Inhalten in Form von Bildsequenzen oder Videos angeboten.

2.2.1 Eingabe: Das Basismedium

Basismedien müssen für die Darstellung an einem sphärischen Display – abhängig von der Globensoftware – in einer bestimmten (geographischen) Projektion vorliegen. Riedl und Kristen [RIE-10] nennen mehrere Anforderungen für die Wahl einer Projektion zur effizienten Darstellung auf einem digitalen Globus:

- einfache Datenaufbereitung (auch für fachfremde Personen)
- möglichst geringe (Pixel)Verzerrung
- geringe Dateigröße (erlaubt mehrere animierte Kartenebenen, wichtig für den Download von Echtzeitthemen)
- möglichst gleichmäßige Pixeldichte
- effiziente Darstellung mittels moderner Grafikkarten

Die gängigen Globensysteme der zuvor genannten Firmen verwenden primär die quadratische Platkarte als Basisabbildung [FOO-08], [RIE-10], [ARC-12b], [SOS-12b], aus der in weiterer Folge die benötigte Darstellung generiert wird, die für das eingesetzte sphärische Display projiziert werden muss.

Die abstandstreue Zylinderprojektion in normaler Lage mit einem Berührungskreis am Äquator dürfte derzeit den besten Kompromiss in Bezug der oben genannten Anforderungen darstellen. Die quadratische Platkarte wird in vielen Anwendungen, die globale Daten verarbeiten und visualisieren, als Basis verwendet. Demzufolge liegen viele globale Datensätze zu den unterschiedlichsten Themen bereits in dieser Projektion vor und können ohne weitere Bearbeitung in das Globensystem eingespielt werden. [RIE-10]

Einen weiteren wesentlichen Vorteil der quadratischen Platkarte stellt die einfache computergestützte Verarbeitung dar. Die geographischen Koordinaten (Breite und Länge) können sehr einfach in kartesische Koordinaten (x und y) überführt werden, wodurch ein sehr schneller und simpler Zugriff auf die Daten ermöglicht wird. Auch 3D

Graphikbibliotheken können „Latitude-Longitude Maps“ ohne großen Programmieraufwand für das Texture Mapping einsetzen. [RIE-10]

Mit den bereits in der Einleitung erwähnten Verzerrungseigenschaften der quadratischen Plattkarte gehen weitere Probleme einher. So führen Riedl und Kristen [RIE-10] an, dass die zur Verfügung stehende Pixelanzahl für jede geographische Breite konstant ist, obwohl die Länge der Breitenkreise mit wachsender Breite abnimmt. Folglich werden die Pixel ineffizient verwendet (dem Pol und dem Äquator stehen jeweils die selbe Anzahl an Pixel zur Verfügung). Daraus ergibt sich nicht nur eine zu große Datenmenge, sondern auch Artefakte im Polbereich, da aus vielen unterschiedlichen Farbwerten einige wenige interpoliert werden müssen. Die unterschiedliche Pixeldichte muss also auch bei der Erstellung von Vektordaten-Overlays aufgrund der sonst auftretenden Verzerrungen berücksichtigt werden.

Aufgrund dieser Problembereiche der Plattkarte (Verzerrungen, unterschiedliche Pixeldichte, ineffiziente Nutzung der Pixel) diskutieren Riedl und Kristen [RIE-10] alternative Basisabbildungen als Grundlage für taktile Hypergloben. Als Alternativen vergleichen sie die CubeMap, das Sphere Mapping bzw. Dual Sphere Mapping und sphärische Zweiecke mit der Plattkarte. Darüber hinaus präsentiert Kristen [KRI-12] mit einer Neuordnung der Sinusoidal Projektion eine weitere Alternative. Ihre derzeitige Conclusio [RIE-10]: *„Generell kann aufgrund der einfachsten Erstellung und allgemeinen Bekanntheit die quadratische Plattkarte als Standardabbildung für das Mapping von globalen Daten auf einem digitalen Globus empfohlen werden.“*

2.2.2 Ausgabe: Das Sphärische Display

Da der Schritt der Verarbeitung stark von der Ausgabe abhängt, werden zuerst die unterschiedlichen Typen sphärischer Displays, die bei taktilen Hypergloben verwendet werden können, besprochen.

So wie es unterschiedliche Kategorien von Globen gibt, können auch mehrere Typen sphärischer Displays unterschieden werden. Ein Aspekt ist die Art der Aufbringung des digitalen Bildes auf den Globenkörper, also das verwendete Projektionsverfahren. Ein anderes Merkmal ist die Beschaffenheit des Globenkörpers selbst.

Projektionsverfahren, die im Bereich taktiler Hypergloben zum Einsatz kommen [RIE-08], [RIE-10]:

- Bei Systemen mit **Außenprojektion (AP)** wird mittels mehrerer Projektoren das Bild – anhand einer allgemeinen perspektivischen Azimutalabbildung – auf einen lichtundurchlässigen Globenkörper projiziert. Je nach der gewünschten

Abdeckung bzw. Qualität der Darstellung auf der Kugeloberfläche werden vier bis sechs Projektoren um den Globenkörper angeordnet. Ein mögliches Schema der Anordnung von Projektoren und Globenkörper zeigt Abbildung 6 AP.

- Bei dem Typ **Innenprojektion (IP)** wird die Bildinformation – in Form einer vermittelnden Azimutalabbildung – durch eine Öffnung an einem der beiden Pole auf die Innenseite eines lichtdurchlässigen Globenkörpers projiziert. Diese Umsetzung basiert entweder auf einem Fischaugenobjektiv (Abbildung 6 IP_f) oder einem konvexen Spiegel (Abbildung 5 (d)). Bei letzterer Variante kann durch die Verwendung zweier Projektoren eine höhere Auflösung erzielt werden (Abbildung 6 IP_m). Eine weitere Variante sind Systeme, bei denen die Projektoren innerhalb des Globenkörpers angebracht werden. Ein Prototyp wurde im Jahr 2009 von der HRG entworfen (Abbildung 6 IP_d).
- Bei der **Direkt Projektion (DP)** stellt der Globenkörper keine Projektionsfläche wie bei den zwei vorangegangenen Systemen dar, sondern den Projektor selbst (Abbildung 6 DP). Genauer: es handelt sich um einen tatsächlichen „sphärischen Monitor“ ohne Zwischenschaltung optischer Systeme. Gegenwärtig bestehen sphärische Displays mit DP im Großteil aus in Kugelform angeordneten LED-Panels.

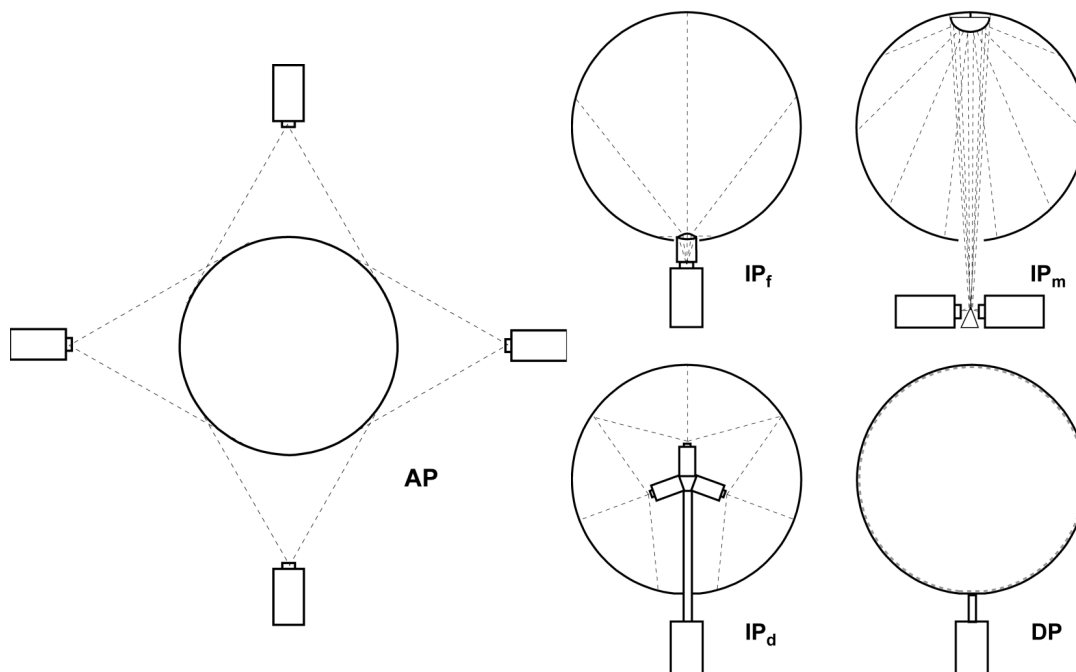


Abbildung 6: Schemata unterschiedlicher Projektionsverfahren bei sphärischen Displays
 AP: Außenprojektion, IP_f: fischaugenbasierte Innenprojektion, IP_m: spiegelbasierte Innenprojektion mit zwei Projektoren („Dual-Beamer“), IP_d: Innenprojektion mit mehreren Projektoren innerhalb des Globenkörpers, DP: Direktprojektion („sphärischer Monitor“)
 (Quelle: [RIE-10])

Die oben erwähnten Unternehmen, ausgenommen NOAA SOS, arbeiten vorwiegend mit Innenprojektionen (Fischaugen-, Single- oder Dualprojektoren). SOS projiziert das Globenbild hingegen von außen auf den Globenkörper.

In der Regel werden **starre Globenkörper** (z.B. beschichtetes Acrylglas) verwendet. Bei der fischaugenbasierten IP und der AP gibt es auch **aufblasbare Globenkörper**. Je nach Einsatzdauer und -ort weisen die einzelnen Varianten Vor- und Nachteile auf. [RIE-11a]

Wie bei analogen Globen bestimmt auch bei taktilen Hypergloben der Globenkörper den Maßstab der kartographischen Darstellung. Je nach dem Durchmesser des eingesetzten sphärischen Displays errechnet sich der Maßstab bzw. die Maßstabszahl bspw. wie folgt:

$$\text{Maßstabszahl} = \text{mittlerer Erddurchmesser} / \text{Globendurchmesser}$$

Das ergibt für gängige Globendurchmesser von 40 cm, 80 cm bzw. 200 cm Maßstäbe von rund 1 : 30 Mio., 1 : 15 Mio. bzw. 1 : 6 Mio. Je nach der Größe des sphärischen Displays sollte die Auflösung der Projektoren gewählt werden.

Eine detailliertere Auseinandersetzung mit den unterschiedlichen Typen sphärischer Displays, deren Spezifikationen (Bildqualität, Auflösungsvermögen, etc.) sowie Vor- und Nachteile ist bei Riedl [RIE-08], [RIE-09] oder Riedl und Kristen [RIE-10] zu finden.

2.2.3 Verarbeitung: Die Globensoftware

Die Globensoftware ist in Abhängigkeit des verwendeten sphärischen Displays primär dafür verantwortlich das Globenbild für die benötigte Projektion (= Globenabbildung) aufzubereiten. Mit dieser Projektion wird die (kartographische) Darstellung korrekt am Kugeldisplay angezeigt.

Für das Verständnis unterschiedlicher Methoden der Vektordatenintegration in das gesamte Globensystem spielt die Funktionsweise von unterschiedlichen Globensoftwareangeboten eine große Rolle. Für diese Diplomarbeit ist vor allem die Autoren- und Präsentationssoftware OmniSuite von der HRG entscheidend.

2.2.3.1 ARC Science Simulations

ARC Science bietet für ihre OmniGloben (spiegelbasierte Innenprojektion) die Softwarepakete Geometer und OmniExplorer an. Beide Pakete können anhand einer LookUp-Tabelle (LUT) die Plattkarte in das „Omni Disk Image“ (eine vermittelnde Azimutalabbildung) konvertieren. Der OmniExplorer, die Präsentationssoftware,

erledigt diesen Schritt in Echtzeit und projiziert die Abbildung auf den Globus. Der Geometer bereitet die Plattkarten auf dieselbe Weise für den OmniExplorer auf, kann aber auch Bildsequenzen und Videos verarbeiten. Die LUT beinhaltet Transformationswerte, die von der Hardwarekonfiguration des sphärischen Displays abhängen (Form und Position des Spiegels, Abstand zwischen Projektor und Spiegel) und muss somit für jede einzelne Globeninstallation extra konfiguriert werden bzw. bei Veränderung der Installation (z.B. Austausch des Projektors) von Hand nachjustiert werden. Aus der Plattkarte wird zunächst eine südpolständige Azimutalabbildung generiert (Abbildung 7 (a)), aus der mit Hilfe der LUT die geographischen Breiten radial verzerrt werden (Abbildung 7 (b)), damit das „Omni Disk Image“ (Globenabbildung) entsteht. Diese Abbildung weist eine nördliche Ausdehnung bis ca. 70° - 75° auf. [KRI-12], [ARC-12b]

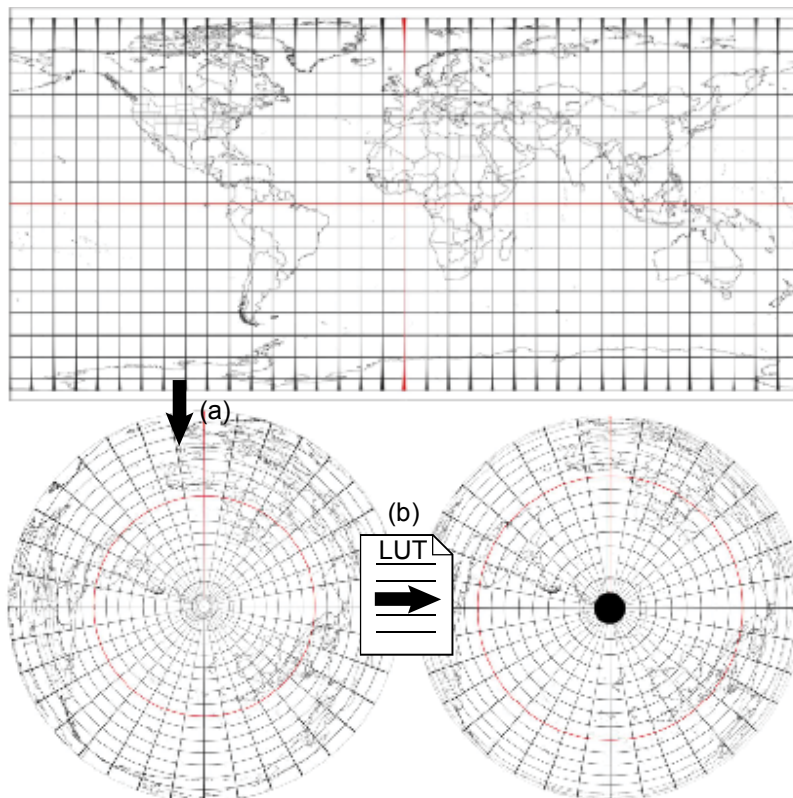


Abbildung 7: Funktionsweise der Globensoftware von ARC Science

Das Basismedium wird in eine südpolständige Azimutalabbildung transformiert (a). Anhand einer LookUp-Table wird die für die Projektion benötigte Darstellung berechnet (b).

(Quelle: einzelne Abbildungen bei [KRI-12], [ARC-12b])

Außer der Rotation um die vertikale Erdachse erlaubt das System keine Rotation um eine beliebige Achse. Ein solcher Effekt muss bereits mit Hilfe einer Bildsequenz bzw. eines Videos in Form des Basismediums umgesetzt werden.

2.2.3.2 NOAA Science On a Sphere

NOAA Science On a Sphere bieten eine auf Linux Betriebssystemen laufende Globensoftware an, die das Globenbild für Außenprojektionen aufbereitet. Hierbei werden von der Platkarte ausgehend vier transversale Azimutalabbildungen errechnet, wobei sich die Zentren der Abbildungen entlang der Äquatorebene um jeweils 90° unterscheiden [SOS-12a], [SOS-12b]. Dieser Ansatz ergibt sich durch die Anordnung der vier Projektoren um die Kugel (Abbildung 6 AP).

2.2.3.3 Global Imagination

Global Imagination hat ein großes Softwareangebot [GIM-12] für die unterschiedlichsten Verwendungszwecke und Anwendungen eines sphärischen Displays. Das Unternehmen produziert und vertreibt sphärische Displays mit einer fischaugenbasierten Innenprojektion (MagicPlanet). So wie die ersten beiden Angebote verwendet Global Imagination den zweidimensionalen Raum um die benötigte Projektion zu generieren, diese wird bei GI „Disk“ genannt. Auch diese Abbildung ist eine modifizierte polare Projektion, bei deren Erstellung die Beschaffenheit des eingesetzten Fischaugenobjektives und die Größe des sphärischen Displays entscheidende Faktoren darstellen. Ein Beispiel ist das AfterEffects Plug-In namens FullGlobe, das aus unterschiedlichen Quellen und Projektionen die „Disk“ (Globenabbildung) für den Magic Planet erstellt. [FOO-08]

2.2.3.4 Science On a Snow Globe

Einen weiteren, eher experimentellen Ansatz liefert Nirav Patel [SSG-12]. Im Vergleich zu den zuvor genannten wird dieses System nicht kommerziell vertrieben, der Quellcode steht frei zur Verfügung. In Anlehnung an SOS nennt sich das Projekt Science On a Snow Globe (SOSG). Das Projekt entspricht mehr einem „Proof of Concept“ als einem verkaufsfähigen Produkt. Ein erster, wenig performanter Ansatz generiert die Ausgabe mittels Python und dem Modul PyGame. Ein zweiter Ansatz, der auch Videodateien und Rotationen um die Erdachse integriert, wurde mit SDL, OpenGL, libvcl und GLSL (Programmiersprache für Shader unter OpenGL) umgesetzt. Wird bei dem ersten Ansatz die Transformation der Platkarte zur Fischaugendarstellung auf der CPU berechnet, verwendet der zweite Ansatz einen Fragment Shader um die pixelweise Konvertierung auf der GPU zu implementieren.

Das selbst gebaute, kleine sphärische Display (ca. 20 cm Durchmesser) verwendet einen Mikroprojektor, dem ein Fischaugenobjektiv vorgeschaltet ist. Anhand der Abbildungsfunktion des verwendeten Objektivs und gewissen variierbaren Parametern wird pixelweise der Farbwert durch den Fragment Shader für die Azimutalabbildung

(Globenabbildung) berechnet. Mehr zu dem Projekt und dem selbst gebauten „Mini sphärischen Display“ ist unter [SSG-12] zu finden.

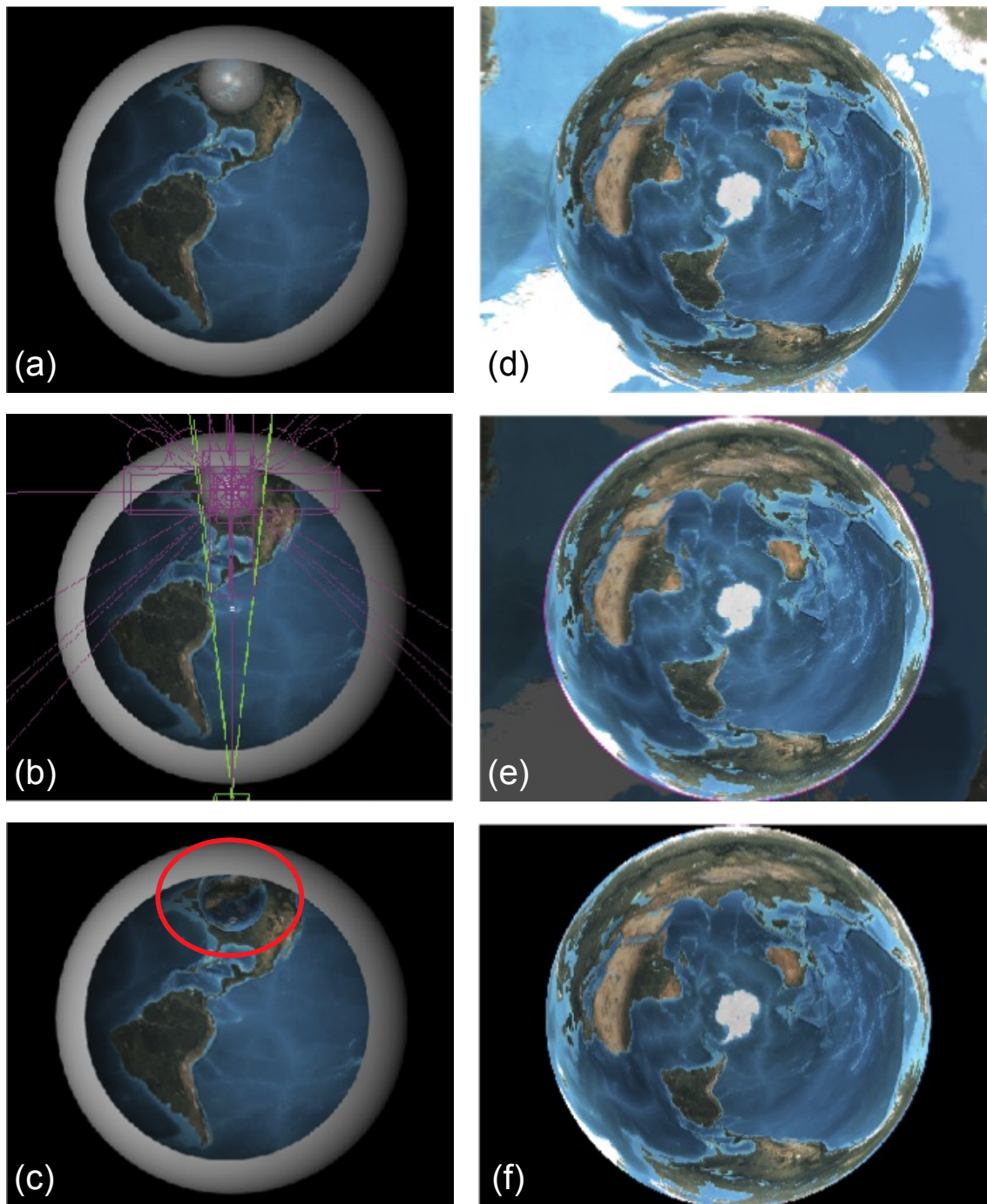


Abbildung 8: Die Funktionsweise der OmniSuite Render Engine

Bild (a) ist die Innenansicht des texturierten virtuellen Globus mit einem Modell des Spiegels des sphärischen Displays. In (b) wird mittels mehrerer Kameras (violett) eine Cube Map erstellt, die als Textur für das Spiegelmodell dient (c). Die Ansicht (d) ist aus dem Blickwinkel der Szene-Kamera (grün in (b)) und nimmt den Spiegel auf. Durch Ausblenden der übrigen Szene (e) entsteht die Omniglobe-Abbildung (f).

(Quelle: [KRI-12], erweitert)

2.2.3.5 HRG/Globoccess

OmniSuite ist ein Programmpaket, das von Jürgen Kristen (Mitglied der HRG) entwickelt wird und vor allem bei Installationen von Globoccess eingesetzt wird. In seiner Diplomarbeit [KRI-12] beschreibt Kristen das gesamte Programm, das in unterschiedliche Module untergliedert ist, sehr ausführlich. Grob lassen sich 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 „Omniglobe-Abbildung“.

Das Programm wird in C++ entwickelt und verwendet die freie 3D Grafik-Engine Ogre3D. Kristen baut das sphärische Display virtuell nach. Abbildung 8 stellt die Funktionsweise der Render Engine dar. Er setzt in die Ogre-Szene eine Kugel (Globenkörper), die mit dem Inputmedium (Plattkarte) texturiert wird. Im Nordpolbereich der Kugel befindet sich ein Modell eines Spiegels, das den verbauten Spiegel im sphärischen Display imitiert. Abbildung 8 (a) zeigt die Innenansicht dieser Szene. Da Spiegel-Effekte, wie sie mit Raytracing-Verfahren, aber nicht mit Grafik-Engines, einfach möglich sind, verwendet Kristen Environment Mapping zur Realisierung des Spiegel-Effekts. Dabei wird im Bereich des Spiegelmodells eine Cube Map der Szene erstellt (Abbildung 8 (b)), die als Textur für das Spiegelmodell dient (Abbildung 8 (c)). Mit einer Kamera im Südpolbereich der Kugel (entspricht dem Projektor des sphärischen Displays) wird der Spiegel abgefilmt (Abbildung 8 (d)). Damit die Omniglobe-Abbildung (Abbildung 8 (f)) entsteht, muss schließlich der Rest der Szene (virtueller Globenkörper) ausgeblendet werden (Abbildung 8 (e)).

Dieser Vorgang wird in Echtzeit umgesetzt, wodurch Interaktionen (z.B. Rotation des Globenkörpers) ermöglicht werden und sofort am sphärischen Display dargestellt werden. Darüber hinaus können mehrere kartographische Ebenen gleichzeitig angezeigt bzw. zur Laufzeit ein- und ausgeblendet werden. Mit dem Graphischen User Interface (GUI) von OmniSuite befindet sich ein höchst benutzerfreundliches und interaktives System in stetiger Weiterentwicklung, das bereits bei vielen Globeninstallationen weltweit eingesetzt wird und auch sehr gut von Laien bedient werden kann. Weiterführende Informationen sind bei [KRI-12] nachzulesen.

2.2.3.6 Sphere (Microsoft)

Bei dem Projekt „Sphere“ von Microsoft wird ebenfalls ein 3D Ansatz verwendet. Benko u.a. [BEN-08] untersuchen in dieser Studie die Multi-Touch Fähigkeit von sphärischen Displays.

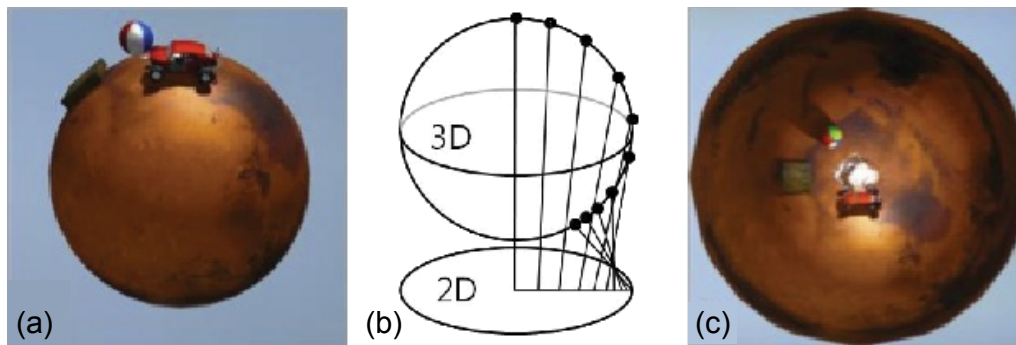


Abbildung 9: Funktionsweise des Vertex Shaders von Microsofts Sphere Projekt

Das darzustellende Thema wird in 3D Koordinaten erstellt (a). Der Vertex-Shader (schematische Darstellung (b)) berechnet für jeden 3D-Vertex den entsprechenden 2D-Vertex und erzeugt die 2D Projektion für das sphärische Display (c).

(Quelle: einzelne Abbildungen bei [BEN-08])

Der darzustellende Inhalt wird in 3D Koordinaten erstellt (Abbildung 9 (a)), und anschließend mittels eines eigenen Vertex Shaders (Abbildung 9 (b)) in die für die Projektion benötigte radiale zweidimensionale Darstellung umgerechnet (Abbildung 9 (c)). Auf die Implementierung der verwendeten Shader wird in der Arbeit nicht genauer eingegangen. Die Software wurde in C# und mit dem Microsoft XNA 2.0 Framework (3D Grafik Bibliothek, die auch bei der Programmierung der X-Box verwendet wird) umgesetzt. Das sphärische Display war ein fischaugenbasierter MagicPlanet von Global Imagination. [BEN-08]

Die vorgestellten Globensoftwarepakete bereiten auf unterschiedlichste Weise Eingangsdaten für die Darstellung am sphärischen Display auf. Neben Verfahren, die die notwendigen Transformationen in der Ebene durchführen, verwenden andere Programme den virtuellen dreidimensionalen Raum um zu einem Ergebnis zu gelangen. Da OmniSuite von der HRG den letzteren Ansatz verfolgt, werden auch in der Diplomarbeit die Möglichkeiten der Aufbereitung von geographischen Vektordaten im dreidimensionalen Raum genauer behandelt.

2.2.4 Anwendungsbereiche und Themenvielfalt

Damit das System „taktile Hyperglobus“ komplett beschrieben wird, muss an dieser Stelle noch auf die Anwendungsbereiche bzw. die Themenvielfalt eingegangen werden. Neben den technischen Aspekten des taktilen Hyperglobus bzw. des sphärischen Displays ist der Inhalt, die Themen, die am sphärischen Display präsentiert werden, ein sehr wichtiger Aspekt. Erst durch diese globalen geographischen Daten kann ein sphärisches Display zu einem digitalen Globus im realen Raum werden (taktile Hyperglobus oder Hologlobus).

Die bereits oben angesprochene Themenbibliothek von Globoccess [GLO-12b], die gemeinsam mit der HRG gepflegt wird, gibt einen guten Überblick über die verfügbaren erd- bzw. planetenbezogenen Inhalte für taktile Hypergloben. Mit dem interaktiven „Erzählen“ von „Globalen Geschichten“ beschäftigt sich derzeit Sebastian Wintner (HRG) in seiner Diplomarbeit [WIN-12].

Taktile Hypergloben – vergleicht man die Webseiten [GLO-12a], [GIM-12], [SOS-12a] der bisher genannten Anbieter taktiler Hypergloben – werden derzeit vor allem in Museen und Wissenschaftszentren, in Planetarien, in immer mehr Schulen, in Forschungseinrichtungen und auch bei Messen als „Wandergloben“ eingesetzt.

Eine sehr prägnante Zusammenfassung der Forschung in dem Bereich taktiler Hypergloben liefert der Artikel „Der Einsatz sphärischer Displays zur Visualisierung globaler Phänomene“ von Riedl und Kristen [RIE-10] sowie der Artikel „Der Globus ist tot, es lebe der Globus!“ von Riedl [RIE-11b].

3 Geographische Vektordaten und deren kartographische Darstellung

Nach dieser ausführlichen Einführung über sphärische Displays und taktile Hypergloben folgt nun ein weitaus bekannteres Thema der Geoinformatik bzw. Geovisualisierung: geographische Vektordaten und deren Visualisierung mittels kartographischer Signaturen. Wie nun bereits beschrieben wurde, verwenden gegenwärtige taktile Hypergloben Rasterdaten als Basisformat. Da die Diplomarbeit die Möglichkeiten der graphischen Integration von geographischen Vektordaten bzw. -geometrien (Punkt, Linie, Fläche) in solche Systeme untersucht, behandelt dieses Kapitel die Spezifika solcher Daten. Welche Anforderungen an Vektordaten für die Wiedergabe am sphärischen Display entscheidend sind, wird in Kapitel 5.3 erörtert.

3.1 Das Vektormodell und dessen Geometrien

In der Geoinformation bzw. der digitalen Kartographie werden Sachverhalte der realen Welt in digitalen Datenmodellen abgebildet. Neben räumlichen Attributen (geometrische Objektinformationen) werden auch nicht-räumliche Attribute (formale Objektbeschreibungen, Metadaten), nicht-räumliche Relationen (Beziehungen zwischen bestimmten Objektarten) sowie raumbezogene Relationen (Beziehungen zwischen raumbezogenen Objekten) geographischer Objekte in sog. digitalen Objektmodellen (DOM) verwaltet. [HAK-02]

Neben dem objektbasierten Vektormodell gibt es noch weitere Datenmodelle, die in der Geoinformation Anwendung finden, die aber hier nicht weiter erläutert werden. Ein Beispiel sind feldbasierte, mosaikartige Datenmodelle, die das Interessensgebiet (Feld) in thematisch homogenen Teilflächen untergliedert. Rasterdaten – das Feld wird in quadratische oder rechteckige Gitterzellen untergliedert – gehören diesem Konzept an. [BAR-05], [HAK-02]

Bei Vektormodellen stellt der Punkt, als Träger der geographischen Koordinaten, die geometrische Grundstruktur dar. Linien und auch Flächen werden aus einer Abfolge mehrerer Punkte konstruiert. Einzelne Punkte, Linien und Flächen werden in diesem Zusammenhang auch Features genannt. Thematische Aussagen werden anhand von Attributen, die mit der entsprechenden Geometrie verknüpft sind, verwaltet. Da Vektormodelle auf diskreten Objekten (Punkte, Linien, Flächen) aufbauen, wird das Interessensgebiet im Allgemeinen nicht unbedingt lückenlos überdeckt. [BAR-05]

Nach [HAK-02] beschreiben Punktdaten „[...] reale punktförmige Objekte (z.B. Lage- oder Höhenfestpunkte), punktförmig generalisierte, ursprünglich flächenförmige, diskrete Objekte (z.B. Brunnen) oder ausgewählte Punkte kontinuierlicher Objekte in Form von Wertefeldern durch Koordinaten [...]“. Liniendaten als Folge von Punkten repräsentieren „[...] reale linienförmige Objekte (z.B. Grenzen von Verwaltungsgebieten, Bruchkanten, Netzwerke) oder linienförmig generalisierte, ursprünglich flächenhafte diskrete Objekte (z.B. Straßenachsen).“ Schließlich beschreiben Flächendaten „diskrete Flächenobjekte durch geschlossene Randlinien.“

In Abbildung 10 wird das Vektormodell am Beispiel von zwei Liniengeometrien gezeigt. Die Datenstruktur des DOMs wird in Form einer Geometrie-Attribut-Tabelle dargestellt. Eine Liniengeometrie wird hier durch eine Folge von mindestens zwei oder mehreren geordneten x und y Koordinaten repräsentiert. Mit jeder Geometrie ist in diesem Beispiel ein Attribut verbunden. Die möglichen Ausprägungen des Attributwertes sind hier „Autobahn“ und „Landstraße“. Neben der Tabelle werden die Geometrien durch die linienhafte Verbindung der einzelnen Punkte graphisch veranschaulicht – es werden nur die Geometrien dargestellt, die Qualität der Geometrie (Attributwert) ist hier nicht erkennbar.

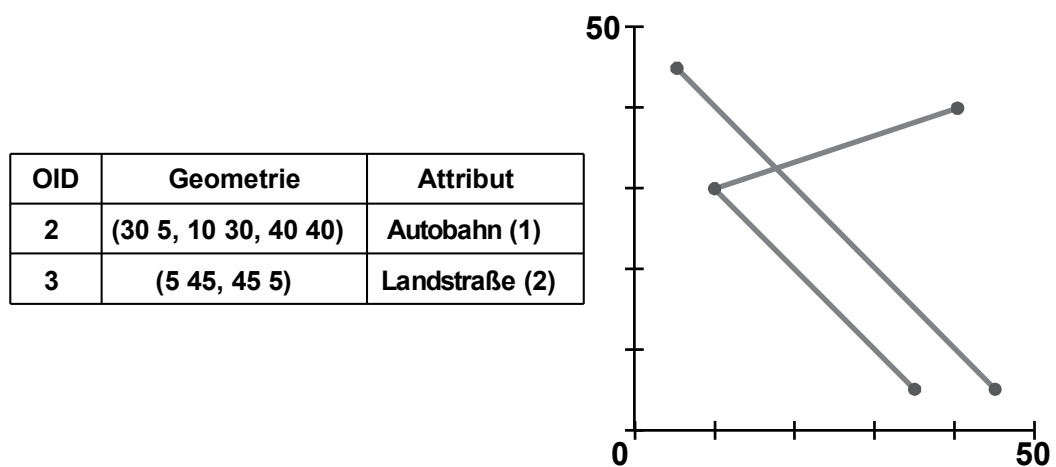


Abbildung 10: Das Vektormodell am Beispiel von Liniengeometrien

Organisations- bzw. Datenstruktur des DOMs von zwei Liniengeometrien mit unterschiedlichen Attributwerten dargestellt durch eine Tabelle (links). Einfache Darstellung der Geometrien des DOMs durch linienhafte Verbindung der Punkte (rechts).

(Quelle: eigene Darstellung)

Für die Diplomarbeit ist primär die geometrische Objektinformation – die Koordinaten der Punkte (bzw. Stützpunkte) – der Vektordaten entscheidend, da diese die Grundlage der Visualisierung darstellt. Die Visualisierung kann jedoch auch durch beschreibende Attribute beeinflusst werden (z.B. Bildung von Klassen für die Visualisierung anhand

eines bestimmten Attributs). Das Kapitel 3.2 bespricht die Visualisierung von Vektorgeometrien im Detail.

Innerhalb der Vektormodelle können mehrere Konzepte unterschieden werden. Das unstrukturierte Datenmodell oder auch „Spaghetti-Modell“ modelliert für jedes Feature eine eigene, in sich geschlossene Geometrie, so auch das Beispiel in Abbildung 10. Wenn ein Koordinate Teil von zwei unterschiedlichen Flächengeometrien ist, wird der Punkt in beiden Geometrien gespeichert. Nachbarschaftsbeziehungen zwischen Features sind in diesem Modell nicht per se modelliert, können aber aus den gespeicherten Geometrien mittels räumlicher Operationen errechnet werden. So eignet sich dieses Modell weniger für räumliche Analysen als für die Erzeugung kartographischer Ausdrucksformen. [HAK-02]

Ein Modell, das Koordinaten redundanzfrei verwaltet, ist das topologische Datenmodell, das Nachbarschaftsbeziehungen explizit modelliert. Als Grundlage dieses Modells dienen die mathematischen Konzepte der Topologie sowie der Graphentheorie. Hier wird dementsprechend die Terminologie „Knoten, Kanten und Maschen“ bzw. „0-Zelle, 1-Zelle und 2-Zelle“ für „Punkt, Linie und Fläche“ verwendet. Für (komplexe) topologische Strukturen und Relationen gelten gewisse Regeln, die dem topologischen Raum aus mathematischer Sicht Genüge leisten. Diesen Regeln folgend können geographische Objekte in diesem Konzept modelliert werden. Das Modell eignet sich wesentlich besser für raumbezogene Analysen als das zuvor besprochene „Spaghetti-Modell“. Eine Weiterentwicklung des topologischen Modells stellt das hierarchische Vektordatenmodell dar, das effizientere Analysen erlaubt. [HAK-02]

Den topologischen Aspekten des Vektorraumes wird hier keine weitere Aufmerksamkeit geschenkt, da diese primär für raumbezogene Analysen der Daten von Bedeutung sind. Für die kartographische Wiedergabe – in diesem Fall am sphärischen Display – sind unstrukturierte Modelle, die Geometriedaten (und deren Sachattribute) verwalten, hinreichend geeignet.

Ein Modell, das ein unstrukturiertes geometriebezogenes Datenmodell beschreibt, ist die Simple Feature Access Spezifikation des Open Geospatial Consortium [OGC-10].

3.1.1 Simple Feature Geometrien

Der Standard „OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture“ [OGC-10] beschreibt eine allgemeine Architektur für Geometrien einfacher räumlicher Objekte. Der Standard implementiert seinerseits ein sog. Profil des in der ISO 19107:2003 präsentierten

„Spatial Schema“-Konzepts und stellt dafür ein plattformunabhängiges Objektmodell in UML-Notation zur Verfügung.

Der Standard beschreibt und definiert mehrere Geometrietypen, die punkt-, linien- und flächenhafte Geometrien darstellen. Im Folgenden werden die englischen Bezeichnungen, so wie sie in der Spezifikation vorzufinden sind, verwendet.

Ein Punkt wird durch die Objektklasse Point repräsentiert. Ein Point, ein 0-dimensionales geometrisches Primitiv, hat eine x- sowie eine y-Koordinate, die die Position innerhalb des Koordinatensystems wiedergeben. Point Objekte können zusätzlich einen z- und/oder einen m-Wert haben. Der z-Wert gibt z.B. die Höhe in dreidimensionalen Systemen wieder, ein m-Wert kann berechnete und gemessene Werte an diesem Punkt enthalten. In der Geoinformation wird dieser bei der Modellierung von Netzwerken verwendet.

Eine Linie wird durch die Objektklasse Curve dargestellt. Dieses eindimensionale geometrische Primitiv verbindet einen Startpunkt mit einem Endpunkt. Umgesetzt wird eine Curve meist durch eine Abfolge mehrere Punkte, die Geometrie hat zwischen dem Start- und Endpunkt weitere Punkte („Stützpunkte“). Im Simple Feature Standard wird eine solche Abfolge durch die Unterklasse LineString implementiert. Die Verbindung zwischen den einzelnen Punkten (Liniensegmenten) wird hier linear interpoliert. Ist der Startpunkt ident mit dem Endpunkt (die Geometrie ist „geschlossen“ bzw. „closed“) und schneidet sich die Geometrie nicht selbst (die Geometrie ist „einfach“ bzw. „simple“), handelt es sich um einen sog. LinearRing. Abbildung 11 zeigt vier Beispiele dazu, wobei (a) einen einfachen LineString, (b) einen nicht einfachen LineString, (c) einen einfachen und geschlossenen LineString, also einen LinearRing und (d) einen nicht einfachen und geschlossenen LineString darstellt.

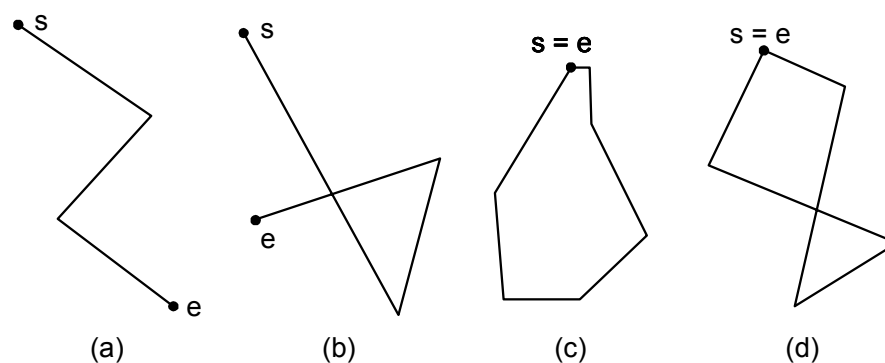


Abbildung 11: Beispiele für LineStrings

Beispiele für LineStrings mit Startpunkt *s* und Endpunkt *e*, wobei: (a) simple LineString, (b) non-simple LineString, (c) simple, closed LineString (=LinearRing) und (d) non-simple, closed LineString

(Quelle: [OGC-10], verändert)

Die Richtung und somit die Orientierung von LineStrings und LinearRings ergibt sich durch die Lage von Start- bzw. Endpunkt. Werden diese zwei Punkte vertauscht, so bleibt die Geometrie bzw. deren Darstellung erhalten, die Richtung bzw. Orientierung wird dadurch umgedreht.

Flächen werden durch Surface-Klasse repräsentiert, wobei Polygon die wichtigste Unterklasse von Surface ist. Daneben gibt es noch die Unterklasse PolyhedralSurface, auf die hier nicht weiter eingegangen wird. Ein Polygon ist ein zweidimensionales geometrisches Objekt, das eine äußere Grenze und keine oder mehrere innere(n) Grenze(n) hat. Grenzen sind Ringe, wie sie zuvor im Sinne von LinearRing besprochen wurden. Jeder innere Ring definiert eine sog. Insel, ein Loch im Polygon. Weist die äußere Grenze eine dem Uhrzeigersinn entgegengesetzte Orientierung auf, so haben alle inneren Grenzen eine dem Uhrzeigersinn entgegengesetzte entsprechende Orientierung. Das Dreieck (Triangle) ist ein Spezialfall des Polygons. Hier setzt sich die äußere Grenze aus drei voneinander unterschiedlichen Punkten zusammen, die darüber hinaus nicht auf einer Geraden liegen. Ein Triangle hat keine inneren Grenzen.

Abbildung 12 zeigt Beispiele für korrekte Polygone, wobei (a) einen äußeren Ring, (b) einen äußeren und eine innere Grenze und (c) neben einem äußeren zwei innere Ringe hat.

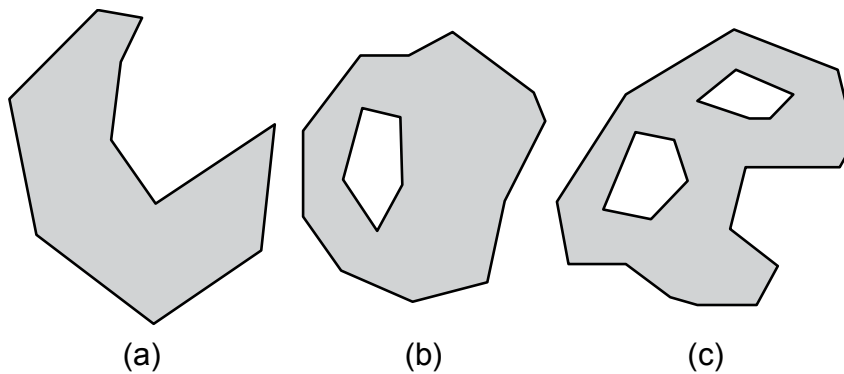


Abbildung 12: Beispiele für Polygone

Beispiele für Polygone mit jeweils einem äußeren Ring und (b) einem inneren Ring bzw. (c) zwei inneren Ringen.

(Quelle: [OGC-10], verändert)

Es gibt mehrere Regeln, die ein Polygon als ein simples geometrisches Objekt korrekt definieren [OGC-10]:

- Polygone sind (im topologischen Sinn) geschlossene Geometrien.
- Innere und äußere Grenzen werden durch Ringe dargestellt.
- Weder innere noch äußere Ringe dürfen einander schneiden.

- Ein Polygon darf keine Zacken/Dornen oder Punktionen aufweisen.
- Zwei beliebige Punkte, die im Polygon liegen, können über eine Linienfolge, die zur Gänze im Polygon liegt, verbunden werden (= verbundene Menge).
- Bei Polygonen mit mindestens einer Aussparung ist der Außenbereich keine verbundene Menge. Jede Insel definiert eine verbundene Menge als Bestandteil des Außenbereichs.

Abbildung 13 zeigt Beispiele für Objekte, die den genannten Regeln nicht Folge leisten und somit nicht mit einer einzelnen Polygongeometrie dargestellt werden können. Bei (a) und (b) gibt es jeweils Schnittbereiche zwischen dem äußeren und inneren Ring. (a) hat darüber hinaus keine verbundene Menge. Bei (c) tritt ein Dorn auf, der äußere Ring schneidet sich selbst. Bei (d) liegt schließlich keine verbundene Menge vor.

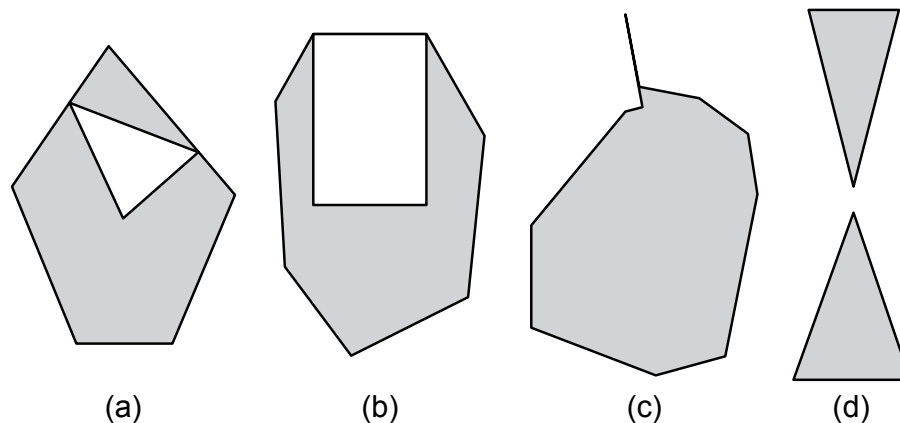


Abbildung 13: Objekte, die nicht als einzelnes Polygon dargestellt werden können

Objekte entsprechen nicht gewissen Regeln, die ein Polygon definieren.

(Quelle: [OGC-10], verändert)

Mit Hilfe von sog. Geometry Collections, die mehrere Geometrien eines Typs in einem Feature zusammenfassen, können die Beispiele (a) und (d) aus der Abbildung 13 jeweils als ein MultiPolygon umgesetzt werden. Darüber hinaus gibt es MultiPoint und MultiLineString Klassen.

Für die Diplomarbeit sind vor allem die Geometrietypen Point, LineString und Polygon von Bedeutung. Multi-Geometrien können für die Visualisierung in ihre einzelnen Geometrien zerlegt werden.

Für die einzelnen Geometrien spezifiziert der Simple Feature Standard Methoden für die Überprüfung räumlicher Beziehungen zwischen Objekten (equals, disjoint, intersects touches, crosses, within, contains, overlaps, etc.) sowie für räumliche Analysen (Distance, Buffer, Intersection, Union, Difference, etc.). Für eine Vertiefung

sei an dieser Stelle an die Spezifikation verwiesen, da diese raumbezogenen Analyse- und Datenmanipulationsfunktionen nicht unbedingt für die Visualisierung der Daten entscheidend sind.

Die Geometrien können anhand zweier Methoden ausgedrückt werden. Einerseits mit der „Well-known Text Representation“ (WKT), andererseits mit der „Well-known Binary Representation“ (WKB). WKT ist eine Markup Sprache zur Repräsentation geographischer Vektorgeometrien. Diese Form ist vom Menschen lesbar und eignet sich gut für die Ausgabe als Text. Im Folgenden einige Beispiele eines zweidimensionalen Koordinatensystems dazu [OGC-10]:

- *Point (10 10)* – ein Punkt
- *LineString (10 10, 20 20, 30 40)* – ein LineString mit drei Punkten
- *Polygon ((10 10, 10 20, 20 20, 20 15, 10 10))* – ein Polygon mit einem äußeren und keinem inneren Ring
- *Polygon ((35 10, 10 20, 15 40, 45 45, 35 10), (20 30, 35 35, 30 20, 20 30))* – ein Polygon mit einem äußeren und einem inneren Ring
- *MultiPoint ((10 10), (20 20))* – eine MultiPoint Geometrie mit zwei Punkten
- *MultiLineString ((10 10, 20 20), (15 15, 30 15))* – ein MultiLineString mit zwei LineStrings
- *MultiPolygon (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))* – ein Multipolygon mit zwei Polygonen

Für jeden Geometrietyp gibt es mit der WKB Repräsentation auch eine binäre Form. Eine Geometrie wird also durch eine Folge von Bytes ausgedrückt. Hierzu werden standardisierte Binärcodierungen verwendet. [OGC-10]

Der OGC-Standard sieht des Weiteren vor, dass jedes Geometrieobjekt mit einem räumlichen Bezugssystem, das das dahinterstehende Koordinatensystem definiert, verbunden ist.

Eine C++ Bibliothek, die die OpenGIS Simple Feature Spezifikation teilweise implementiert, ist die OGR Simple Feature Library [OGR-12]. Mehr über OGR ist im Kapitel 5.3 nachzulesen.

3.2 Kartographische Darstellung der Geometrien

Neben dem oben besprochenen Objektmodell kann auf einer weiteren Ebene, einer graphischen, das Darstellungsmodell beschrieben werden. Das sog. digitale kartographische Modell (DKM) basiert auf dem jeweiligen DOM und stellt dessen Visualisierung dar. Verwaltet ein DOM punkt-, linien- und flächenhafte Geometrien, so verwaltet ein DKM punkt-, linien- und flächenhafte Zeichen, die ebenfalls Geoobjekte der realen Welt repräsentieren – aber auf einer Darstellungsebene. Es geht also um die Modellierung und Visualisierung von (karto)graphischen Ausdrucksformen von Geodaten im Vektorformat – im Fall dieser Diplomarbeit für den taktilen Hyperglobus.

In Abbildung 14 wird das Beispiel aus Abbildung 10 aufgegriffen und aus dem DOM ein mögliches DKM gebildet. Die linienhaften Signaturen (rechts) werden in Abhängigkeit der Attributwerte visualisiert, wobei hier der Farbwert und die Strichstärke variieren. Der Farbwert stellt die unterschiedliche Qualität der Objekte dar („Autobahn“ und „Landstraße“), die Strichstärke kann z.B. auf die allgemeine Fahrbahnbreite von Autobahnen bzw. Landstraßen referenzieren und stellt somit einen quantitative Differenzierung dar.

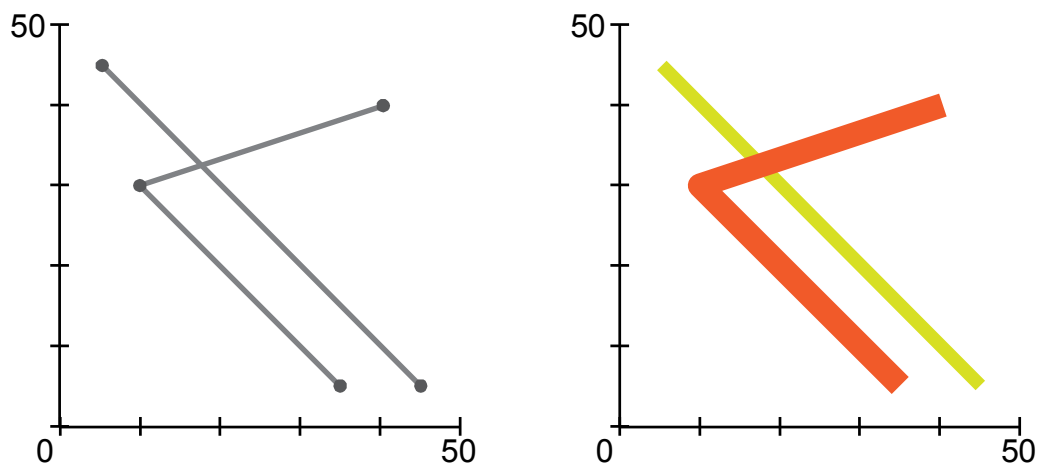


Abbildung 14: Liniengeometrien und entsprechende linienhafte Signaturen

Links werden zwei Liniengeometrien des DOMs dargestellt (vgl. Abbildung 10). Rechts wird eine mögliche Ausprägung des DKMs dargestellt. Die linienhaften Signaturen werden in Abhängigkeit der Attributwerte der Geometrien („Autobahn“, „Landstraße“, vgl. Abbildung 10) visualisiert.

(Quelle: eigene Darstellung)

Das Beispiel zeigt auf, dass Punkt- und Liniengeometrien, die jeweils per Definition keine räumliche Ausdehnung aufweisen, flächig dargestellt werden müssen. Dennoch wird aufgrund der Gestalt jeweils die Bezeichnung punkt- bzw. linienhafte Darstellung verwendet. In diesem Fall ist die Geometrie des DOMs die Mittellinie der linienhaften Darstellungsgeometrie.

3.2.1 Kartographische Gestaltungsmittel

Nach [HAK-02] bilden die Elemente Punkt, Linie, Fläche und zusammengesetzte Zeichen wie Signatur, Diagramm, Halbton und Schrift die Grundlage der kartographischen Gestaltung. Im Folgenden wird ein kurzer Überblick über Punkte, Linien und Flächen bzw. punkt- linien- und flächenhafte Signaturen gegeben. Die Elemente Diagramm, Halbton und Schrift werden nicht besprochen. Der Einfachheit halber werden lediglich geometrische (abstrakte) Signaturen besprochen. Auf bild- und symbolhafte Darstellungen wird nicht eingegangen.

Eine qualitative bzw. quantitative Differenzierung innerhalb der jeweiligen Zeichenklasse, das Verstärken bzw. Abschwächen einer Aussage sowie das Hervorrufen von Assoziationen kann durch ihre graphische Variation der Zeichen erreicht werden. Und weiter listet [HAK-02] die möglichen graphischen Variationen auf:

Die „**Größe (Breite)** zeigt unterschiedliche Quantitäten und eignet sich besonders gut zum Bewerten durch Betonen oder Abschwächen“. Die „**Form** lässt Qualitäten unterscheiden und erleichtert bei bild- und symbolhaften Zeichen Assoziationen“. Die „**Füllung** kann sowohl Qualitäten als auch Quantitäten gliedern“. Der „**Tonwert** als Helligkeitsstufe beschreibt Quantitäten“. Die „**Richtung (Orientierung)** eignet sich für weitere Aufgliederung von Merkmalen sowie zum Hinweis auf zeitliches Verhalten“. Schließlich beschreibt die **Farbe** „als Farbton in erster Linie verschiedene Qualitäten, als Farbsättigung und als Farbhelligkeit auch Quantitäten, daneben auch zeitliches Verhalten. Sie ist in hohem Maße für Assoziation geeignet“.

Die graphischen Variablen können je nach verwendetem Zeichen eingesetzt werden. Bei Punkten kann durch die Variation der Farbe eine Objektqualität unterschieden werden. Punkte sind tatsächliche graphische Punkte, die eine gewissen Minstdurchmesser aufweisen und die Lage eines Objekts angeben. Mit der Veränderung der Größe innerhalb einer Objektklasse können Quantitäten unterschieden werden. Hier ist der Übergang zu lokalen, punkthaften Signaturen fließend. [HAK-02]

Bei Linien (alle nicht unterbrochenen Striche, die eine Lage angeben) ist eine graphische Variation durch die Farbe sowie die Größe, die Strichstärke, möglich. Linien können einerseits Objekte abgrenzen, andererseits gleiche Werte in einem Kontinuum verbinden (Isolinien). Um weitere qualitative oder quantitative Aussagen treffen zu können, empfiehlt [HAK-02] die Ergänzung durch Signaturen oder Schrift bzw. die Überführung der Linie zu einer linienhaften Signatur.

Flächen sind nach [HAK-02] sog. Vollflächen mit konstanter Farbe und Helligkeit. Diese zwei Elemente können einer graphischen Variation unterzogen werden. So

können die Lage und Qualität eines flächenhaften Objekts wiedergegeben werden oder flächenbezogene Quantitäten (z.B. Choroplethenkarte) sowie Wertstufen eines Kontinuums.

Um eine höhere Aussagekraft zu erlangen ist der Übergang von diesen drei besprochenen Grundelementen zu kartographischen Signaturen notwendig. Geometrische oder auch abstrakte Signaturen umfassen einfache Figuren wie Kreis, Dreieck oder Quadrat für lokale, punkthafte Signaturen. Unterbrochene Linien, also gepunktete oder gestrichelte Linien bzw. Kombinationen davon sowie Bänder werden für lineare Signaturen verwendet. Abstrakte flächenhafte Signaturen sind geometrisch-rasterförmige Muster und Schraffuren.

Signaturen lassen sich allen zuvor angeführten graphischen Variationen unterziehen. Dadurch können unterschiedlichste qualitative und oder quantitative Aussagen mittels Signaturen getroffen werden. Vorteile von abstrakten Signaturen sind die einfache (programmiertechnische) Herstellung, die große Variationsvielfalt und die hohe Gruppen- und Kombinationsfähigkeit.

So wie für alle kartographischen Ausdrucksformen gelten auch für (digitale) Globen bestimmte Rahmenbedingungen für die graphische Wiedergabe, die Visualisierung der Daten.

In Abhängigkeit des Maßstabs müssen die Daten einer Generalisierung unterzogen werden, damit die entscheidenden Aussagen in den Vordergrund, die unwichtigen in den Hintergrund rücken. Davon hängt auch die Exaktheit der Ortsgebundenheit der Zeichen ab. Um die Lesbarkeit (Wahrnehmbarkeit der Gestalt) von einzelnen Signaturen aufrecht zu erhalten ist eine graphische Mindestgröße bzw. Minimaldimension bei der Visualisierung erforderlich. Um topologische Konflikte bei der Darstellung zu vermeiden und um die Objektrennung bzw. die Lesbarkeit zwischen den Zeichen zu gewährleisten sind die graphische Dichte und Differenzierung zu berücksichtigen. [HAK-02]

Wie und welche dieser Rahmenbedingungen, die in einem sehr engen Zusammenhang zueinander stehen, bei taktilen Hypergloben zu berücksichtigen sind, bzw. wie Simple Feature Geometrien am sphärischen Display visualisiert werden können, wird in Kapitel 5 besprochen.

Ob und wenn ja in welcher Art und Weise die graphische Ausprägung einer Geometrie gespeichert wird, ist vom Datenformat der Vektordaten abhängig.

3.3 Globale Vektordatensätze

Unter globalen Vektordatensätzen sind Vektordaten zu verstehen, die auf einer globalen Maßstabsebene vorliegen. Die Thematik muss dabei nicht zwangsläufig ein weltumspannendes Phänomen behandeln, sie soll jedoch auf einer globalen Maßstabsebene eine gewisse Relevanz aufweisen. So kann aus der Visualisierung der Flugroute von Wien nach Salzburg am sphärischen Display kaum eine nützliche Information gewonnen werden. Die Darstellung der Route von Wien nach New York kann hingegen durchaus mit einem Informationsgewinn verbunden sein. Ob eine Thematik für die Darstellung am sphärischen Display geeignet ist, kann u.a. mit Hilfe der Überprüfung der Globenwürdigkeit [RIE-00] (Kapitel 5.3) entschieden werden. In den nachstehenden Abschnitten wird auf spezifische Dateiformate und Quellen für geographische Vektordaten eingegangen.

3.3.1 Dateiformate

Bekannte Vektorformate in der Geoinformation sind u.a. ESRI Shape Format, KML, GPX für GPS Aufnahmen, GeoJSON als Austauschformat bzw. Anbindungen an Geo-Datenbanken wie PostGIS, in denen Vektorinformation gespeichert sind.

Nachfolgend werden mit dem KML bzw. SHP Format zwei wesentliche, sehr weit verbreitete Formate besprochen. Jedoch auch jedes andere geographische Vektorformat, das Simple Features spezifiziert, kann im Zusammenhang mit der Diplomarbeit herangezogen werden. Eine Liste von Vektorformaten, die durch die Simple Feature Schnittstelle angesprochen bzw. von der oben erwähnten C++ Bibliothek OGR verarbeitet werden können, ist u.a. bei der OGR Dokumentation [OGR-12] zu finden.

3.3.1.1 Keyhole Markup Language (KML)

Das KML Format, das als OGC-Standard in der Version 2.2.0 vorliegt [OGC-08], wurde für die Wiedergabe von geographischer Information in virtuellen digitalen Globen bzw. Earthbrowsern entwickelt. Die hierarchische Datenstruktur von KML folgt einer XML Notation, ist also von Mensch und Maschine lesbar. In dem sog. KML Schema sind die möglichen bzw. notwendigen Elemente und Attribute definiert, die eine KML Datei aufbauen bzw. strukturieren.

Die grundlegenden Geometrien, wie sie oben erläutert sind, werden im KML Standard beschrieben. Die Punkte bzw. Koordinaten werden als geographische Breite und Länge kodiert. Die XML Struktur erlaubt es die unterschiedlichen Geometrietypen in einer KML Datei zu mischen.

Die KML Spezifikation definiert nicht nur Vektorgeometrien, sondern auch Raster Overlays oder andere Elemente, die bspw. zur Kameranavigation gedacht sind. Diese werden hier nicht weiter behandelt. Mehr zu Punkt-, Linien- und Flächengeometrien, wie sie durch KML definiert sind bzw. in Earthbrowsern visualisiert werden, wird in Kapitel 4.2 erörtert.

Wie oben bereits erwähnt wurde, hängt es auch vom Datenformat ab, ob und wie die Symbologie für die Vektordaten gespeichert werden. In einer KML Datei können mit dem Style-Element *Style* definiert werden, die die Visualisierung der Geometrien bestimmen. Dazu gibt es abhängig von der Geometrie weitere Elemente wie: *width* und *outerWidth*, *color* und *outerColor* oder *fill* und *outline*.

Mit diesen und anderen Elementen und Attributen können die oben besprochenen graphischen Variationen für die kartographische Gestaltung umgesetzt werden. Bei anderen Formaten, wie dem ESRI Shapefile Format, werden hingegen keine Symbolstrukturen gespeichert.

3.3.1.2 ESRI Shapefile Format (SHP)

In ESRI Shapefile Datensätzen werden nicht-topologische Vektorgeometrien und Attribute von geographischen Objekten gespeichert [ESR-98]. Die Spezifikation umfasst die einfachen Geometrietypen Punkt, Linie und Fläche (Kapitel 3.1.1).

Im Unterschied zu KML Dateien, können in SHP Datensätzen nur Geometrien eines einheitlichen Geometrietyps gespeichert werden. Wird z.B. eine hydrographische Thematik (Quellen, Flüsse, Seen) bearbeitet, werden drei unterschiedliche Shape Datensätze benötigt. Einen Punktdatensatz für die Quellen, einen Liniendatensatz für die Flüsse (und Uferlinien) sowie einen Flächendatensatz für die Gewässerflächen.

Ein Shape Datensatz setzt sich aus mehreren Dateien zusammen. Die *.shp, *.shx und *.dbf Dateien sind obligatorisch. Die Geometrien (Koordinaten) der einzelnen Features werden in der *.shp Datei selbst gespeichert. Die *.shx Datei ist eine Indexdatei um die Datenverarbeitung (das Suchen im Datensatz) zu beschleunigen. Die Attributtabelle eines Shape Datensatzes wird im dBase Format (*.dbf) gespeichert. Darüber hinaus gibt es optionale Dateien. Dazu zählt bspw. eine Projektionsdatei (*.prj), die Informationen zur Projektion der Geodaten im „Well-known Text“ Format bereitstellt. Des Weiteren gibt es eine Datei für Metadaten (*.shp.xml) oder solche, die Verbindungen und Verknüpfungen zwischen Tabellen beinhalten.

Die *.shp Datei ist binär kodiert. Die genaue Datenstruktur (Byteabfolge) ist in der Spezifikation [ESR-98] zu finden. Die Bytes 32 bis 35 sind bspw. für den Geometrietyp

reserviert. Handelt es sich um einen Polygondatensatz, wird in diesem Bereich der Integer Wert 5 gespeichert.

Stildefinitionen von Features können von Shape Datensätzen nicht gespeichert werden. Hierzu können ESRI Layer Dateien (*.lyr) verwendet werden, die jedoch nur sinnvoll mit ArcGIS (wieder)eingesetzt werden können. Alternativen dazu, um kompatible Symbolgien für andere Programme bereitzustellen, sind die offenen OGC Standards „Symbology Encoding“ und „Styled Layer Descriptor“ im XML Format. [OGC-12]

3.3.2 Quellen

3.3.2.1 Natural Earth Data

Eine sehr gute Quelle für freie (Public Domain), globale geographische Vektordatensätze ist Natural Earth Data [NED-12]. Die Daten werden, je nach Thema, in bis zu drei unterschiedlichen Maßstabsebenen (1 : 10 Mio., 1 : 50 Mio. und 1 : 110 Mio.) im ESRI Shapefile Format angeboten. Alle Daten verwenden ein geographisches Koordinatensystem (WGS84).

Bei den Vektordaten – es werden auch Raster angeboten – werden die Kategorien „Physisch“ (Küstenlinien, Landflächen, Ozeanflächen, Flüsse und Seen, vergletscherte Gebiete, geographische Linien, Gradnetz) und „Kultur“ (Staatsgrenzen- und -flächen, nationale administrative Einheiten erster Ordnung, Dauersiedlungsraum, Parks und Schutzgebiete) unterschieden.

Ein Ziel von Natural Earth Data ist die zivile Bereitstellung von Basisdaten, die für die Erstellung kartographischer Produkte verwendet werden können. Daraus ergibt sich ein weiteres Ziel: um den diversen Ansprüchen gerecht zu werden, wird eine hohe Flexibilität der Daten angestrebt. Fast alle Elemente verfügen über ein Namenattribut, eine Information zur räumlichen Ausdehnung oder Empfehlungen für Zoomstufenbereiche (Web Mapping). Große Polygone sind für eine effizientere Datenverarbeitung unterteilt. Es wurde darauf geachtet, dass die Datumslinie überschreitende Features sich nahtlos aneinander fügen. Die Punktdichte wurde so gewählt, dass auch bei konischen Projektionen glatte Linien entstehen. [NED-12]

3.3.2.2 Vector Map (VMAP)

Vector Map (VMAP) ist eine Sammlung globaler Vektordatensätze (geographische Koordinaten und WGS84) mit mehreren Maßstabsebenen und somit Detailgraden. Level 0 (VMAP0), dessen Daten unter einer Public Domain Lizenz vertrieben werden, hat mit einem Maßstab von rund 1 : 1 Mio. die größte Generalisierung. Level 1 und 2 sind großmaßstäbiger und nur teilweise bzw. gar nicht öffentlich verfügbar. Die Daten

haben einen militärischen Hintergrund, sie werden von der National Geospatial Intelligence Agency (NGA) der USA herausgegeben. [VMA-12c]

Der VMAP0 Datensatz ist in vier Bereiche untergliedert (Nordamerika, Südamerika und Afrika, Europa und nördliches Asien, Südostasien und Australien), die als ZIP Dateien angeboten werden [VMA-12a]. Die Datenstruktur entspricht dem Vector Product Format (VPF), einem Militär Standard für geographische Vektordaten. Unter [VMA-12b] ist die VMAP0 im ESRI Shape Format verfügbar.

Folgende, vorwiegend topographische Themenbereiche werden durch die VMAP0 abgedeckt: politische Einheiten, Ortsnamen, Grenzen, Datenqualität, Höhenangaben, Hydrographie, Industrie, Physiographie, Bevölkerung, Transportwesen, Versorgungseinrichtungen und Vegetation [VMA-12c]

3.3.2.3 Globale Sachdaten

Bei den oben genannten Datenquellen handelt es sich jeweils um Geometriedaten. Eine Quelle für globale Sachdaten, die nicht per se mit einer Geometrie verknüpft sind, ist das „The World Factbook“ der CIA [CIA-12]. Es gibt u.a. die Themenbereiche Geographie, Gesellschaft, Wirtschaft, Kommunikation, Transport und Militär. Diese Daten liegen auf Länderebene vor und können somit anhand eines handelsüblichen GIS mit Geometriedaten verknüpft werden. Bspw. kann das Thema Geburtenrate mit wenigen Schritten mittels eines gemeinsamen Ländercodes in die Attributtabelle der Geometriedaten eingepflegt werden. Zum Erstellen einer Choroplethenkarte können abschließend, basierend auf den Werten der Geburtenrate, Klassen gebildet werden.

Darüber hinaus können alle beliebigen und geeigneten fachspezifische Sachdaten mit Geometriedaten verknüpft werden und als Quelle für (globale) geographische Vektordatensätze herangezogen werden. Zusätzlich bieten viele öffentliche Stellen und Forschungseinrichtungen Geodaten mit ihren Forschungsergebnissen im Internet (frei) an.

4 Stand der Forschung

Es ist nun angebracht zu besprechen, welche Methoden der geographischen Vektordatenintegration bereits bestehen und von Anbietern taktiler digitaler Globen verwendet werden. Darüber hinaus wird erläutert, wie Earthbrowser bzw. virtuelle digitale Globen Vektordaten visualisieren.

4.1 Taktile digitale Globen

Anhand von E-Mailanfragen wurde der aktuelle Stand ermittelt, wie Unternehmen, die taktile Hypergloben produzieren bzw. vertreiben, die Integration von Vektordaten in ihrem System umsetzen. Generell war es sehr schwer Informationen von anderen Globenanbietern (ausgenommen HRG/Globoccess) zu bekommen. Die Antworten blieben leider auch nach wiederholter Nachfrage oberflächlich und wenig genau – aus Sicht eines Unternehmens nachvollziehbar.

Wie oben bereits erläutert, dienen derzeit bei allen genannten Unternehmen globale Rasterdatensätze – primär in Form quadratischer Plattkarten – als Basismedium für das Globensystem. Die Globensoftware ist in der Regel nur für die passende, plattformabhängige Projektion der Rasterdaten für das sphärische Display verantwortlich. Vektordaten müssen bereits beim Aufbereiten des globalen Themas in das Basismedium integriert werden, wobei optimaler Weise die Verzerrungseigenschaften der Basisabbildung berücksichtigt werden. In der Praxis ist dies jedoch eher selten der Fall. In den folgenden Abschnitten sind Beispiele und Methoden angeführt, die diese Verzerrungseigenschaften beachten und bereits von Globenanbietern bzw. der HRG angewandt wurden.

Bei SOS und GI ist die Implementierung einer Teilmenge der KML Spezifikation geplant. Auch ESRI Shapefiles sollen bei GI in Zukunft nativ unterstützt werden. Hier werden auf Anfrage auch die bereits adressierten Probleme genannt: die starken Verzerrungen in den Polgebieten und die „Ost-West-Naht“ bei Plattkarten.

4.1.1 Die Wiedergabe von Punktgeometrien und Namengut

4.1.1.1 Breitenabhängige horizontale Streckung

Eine gängige Methode, die u.a. bei Globoccess/HRG und Global Imagination Anwendung findet, ist die breitenabhängige horizontale Streckung von Punktsignaturen und Namengut. Das Basismedium muss hierbei eine Plattkarte sein. Andreas Lampert (HRG) bespricht in seiner Diplomarbeit [LAM-11] ein Beispiel, das in Adobe Director

umgesetzt wurde. Die Breite der Punktsignaturen und der dazugehörigen Labels werden in Abhängigkeit von deren geographischer Lage verändert. Die neue Symbolbreite entspricht der ursprünglichen Symbolbreite durch den Kosinus der geographischen Breite. Hat die Punktsignatur am Äquator eine Breitenausdehnung von 10 Pixel, so hat die gleich große Signatur bei 80° eine Ausdehnung von 57 Pixel.

In Abbildung 15 hat Lampert eine Global Story über die Kontinentaldrift erweitert. (a) und (b) zeigen Ausschnitte aus einer Plattkarte, wobei in (a) die Signaturen (Punkt und Schrift), nicht jedoch in (b) schon in Abhängigkeit der geographischen Breite verzerrt werden. (c) und (d) zeigen jeweils die Ergebnisse der Visualisierung der Plattkarten auf einem Globenkörper.

Der Mehrwert dieser Methode kann am Beispiel „Buenos Aires“ nachvollzogen werden. Nicht nur die punkthafte Signatur, die in der Plattkarte (b) verzerrt wurde, wird bei der Globendarstellung (d) kreisförmig dargestellt, sondern auch der Schriftzug, der in (b) verzerrt wurde – sichtbar an der Länge der roten Linie – wird in (d) gut lesbar dargestellt.

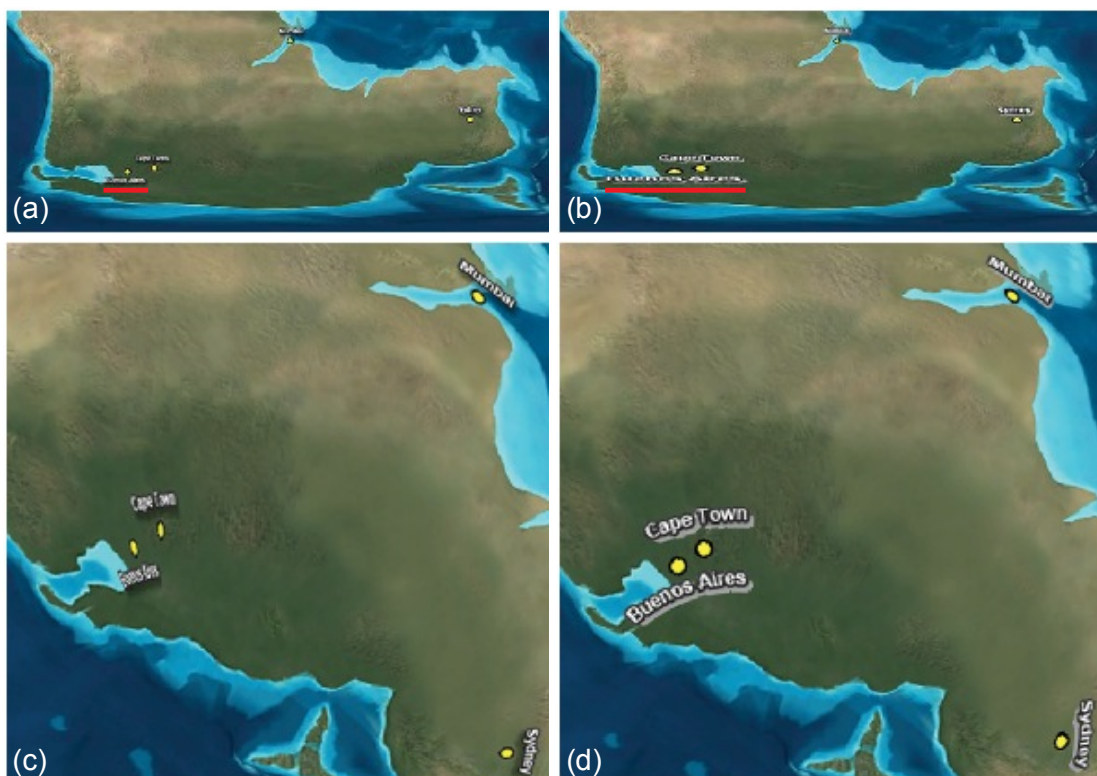


Abbildung 15: Punktsignaturen und Beschriftung am Beispiel der Kontinentaldrift

(a) und (b) zeigen Ausschnitte aus der Plattkarte; (c) und (d) sind Ausschnitte aus der Globenansicht; (a) und (c) ohne Berücksichtigung der Verzerrungseigenschaften der Plattkarte; (b) und (d) unter Berücksichtigung der Verzerrungseigenschaften der Plattkarte durch breitenabhängige horizontale Streckung von Punktsignaturen bzw. Ortsnamen.

(Quelle: [LAM-11], verändert)

Mit dieser Methode können vor allem für kleinere (kreisförmige) Signaturen bis rund $\pm 80^\circ$ Breite gute Ergebnisse erzielt werden. Liegen Signaturen auf einer Breite von $\pm 90^\circ$, kann die Methode nicht angewandt werden, da der Kosinus von 90 zu einer unzulässigen Division durch Null führt.

4.1.1.2 Geodätische Puffer (ArcGIS)

Im Vergleich zu dem oben besprochenen „Ellipsen Ansatz“ eignet sich die folgende Methode auch für kreisförmige Signaturen in Polregionen. Das Softwarepaket ArcGIS von ESRI bietet mit dem geodätischen Puffer (Geodesic Buffer) eine Methode, die den Verzerrungseigenschaften der Platkarte gerecht wird. Wird das gewöhnliche Puffer Tool (Proximity Toolbox) auf einen Punktdatensatz (Shapefile, Geodatabase) angewandt und der Datensatz hat ein geographisches Koordinatensystem (dem Datensatz ist also keine Projektion zugewiesen) und die Pufferdistanz wird in einer Längeneinheit wie Kilometer oder Meilen angegeben, wird ein Algorithmus für geodätisches Puffern verwendet. [ESR-12a]

Das Ergebnis ist ein Flächendatensatz mit Polygonen, die je nach Lage und Verzerrungseigenschaften eine entsprechende Form aufweisen. Je weiter vom Äquator entfernt, desto größer ist die Verzerrung (Platkarte). Durch den Einsatz sehr vieler Stützpunkte (Vertices) wird die entsprechende Deformation umgesetzt. Die Vertices des Puffers in Abbildung 16 sind alle gleichabständig zur gepufferten Punktkoordinate (300 km), die sich auf einer Breite von 80° befindet.

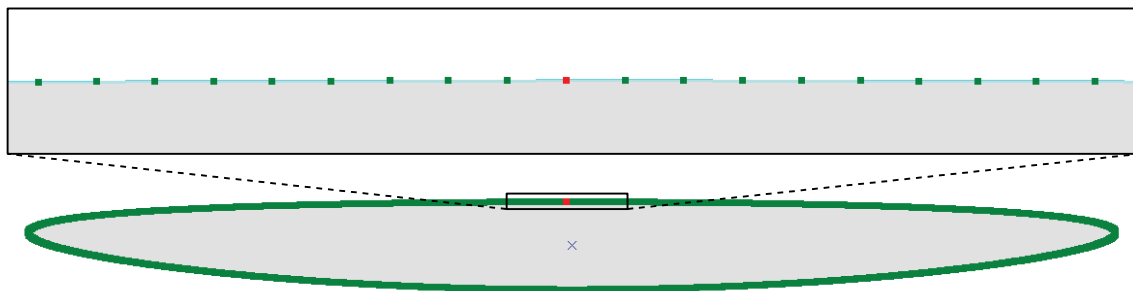


Abbildung 16: Geodätisches Puffern von Punktsignaturen mit ArcGIS (Detailansicht)

Die Punktsignatur wird durch eine hohe Anzahl an Vertices (grüne Punkte) umgesetzt.

(Quelle: eigene Darstellung)

Durch das Hinzufügen eines BUFF_DIST Feldes in der Attributtabelle des Punktdatensatzes kann für jedes Feature eine Pufferdistanz angegeben werden. Bspw. können auf Basis der Einwohner der größeren Städte der Erde Klassen gebildet werden und in das BUFF_DIST Feld ein entsprechender Wert geschrieben werden. Die unterschiedlichen Klassen werden aufgrund der verschiedenen Pufferdistanzen als Punktsignaturen (in Form eines Polygon-Features) visualisiert.

Überschreitet ein Polygon die Datumsgrenze (Ost-West-Naht) wird das berechnete Polygon als Multi-Part-Feature gespeichert (siehe Multi-Geometrien Kapitel 3.1.1). Damit die Punktsignaturen am sphärischen Display visuell ansprechend und „nahtlos“ dargestellt werden, muss bei der Darstellung (*Symbology*) auf eine Außenlinie verzichtet werden. Bei Signaturen, die über die Datumsgrenze gehen bzw. bei Signaturen im Polbereich würden bei der Wiedergabe am sphärischen Display irritierende Linien innerhalb der Signatur aufscheinen (Abbildung 17 (a) und (e)). Eine Lösung dafür ist den Punktdatensatz mit einer etwas größeren Distanz (gewünschte Outline) nochmals zu puffern und das Ergebnis unter den ersten Puffer zu legen. Soll lediglich die Außenlinie erscheinen, kann der Außenlinienpuffer mit dem Erase Tool der Flächenpuffer entfernt werden. (d)

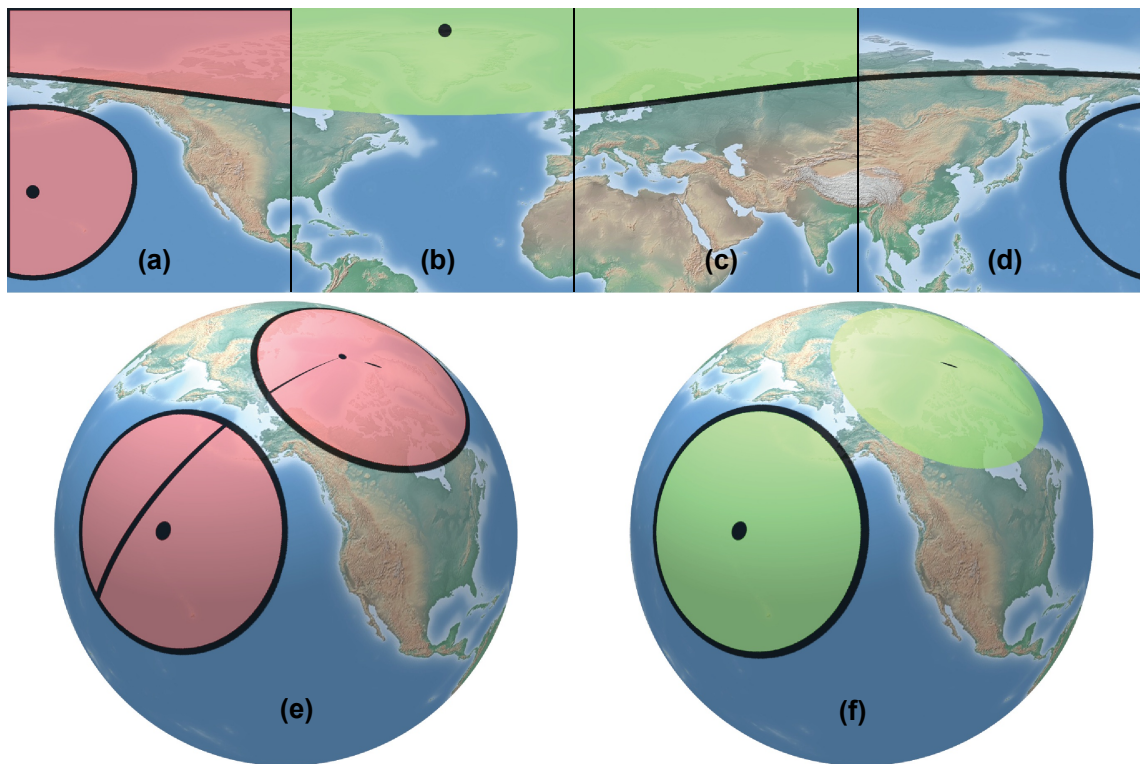


Abbildung 17: Geodätisches Puffern mit ArcGIS zur Wiedergabe von kreishaften Signaturen

3000 km Puffer in der Nähe der „Problemgebiete“ Pol bzw. Datumsgrenze: (a) Standard Außenlinien der ArcGIS Symbolisierung führen zu Artefakten in der Globendarstellung (e); (b) Darstellung ohne Außenlinie; (c) Kombination von zwei Puffern: der Puffer aus (b) wird mit einem zweiten Puffer (mit einer entsprechend größeren Pufferdistanz) hinterlegt; (d) Wiedergabe der Außenlinie (durch Substruktion der Puffer aus (c)); (e) korrekte Darstellung von (b) und (d) in Globenansicht (f); die schwarzen Punkte zeigen die geographische Lage der gepufferten Punktgeometrien an (bei Polbeispiel wird dieser bei der Globendarstellung „verschluckt“).

(Quelle: eigene Darstellung, topographische Grundlage: [NED-12])

In der HRG konnte Gilbert Kotzbek diesen Ansatz bereits erfolgreich anwenden. Er hat mit der Thematik „Atomkraftwerke und deren Sicherheitszonen“ gute Ergebnisse erzielt. Bei der Thematik handelt es sich um ein flächenhaftes Phänomen, das durch punkthafte Signaturen dargestellt wird. Drei verschieden große, konzentrische Puffer (50, 100 und 200 km) zeigen Sicherheitszonen rund um Atomkraftwerke zum Stand 2011 an [GLO-12a].



Abbildung 18: Punktsignaturen mit Geodätischen Puffern (ArcGIS)

vorne: Globenansicht mit formerhaltenden Signaturen; hinten: Plattkarte mit entsprechend aufbereiteten Signaturen um das Ergebnis der Globenansicht zu erlangen; die schwarze Raute bzw. das Sechseck dienen zur einfacheren Identifizierung der jeweiligen Signatur

(Quelle: eigene Darstellung, Thematik: Gilbert Kotzbek (HRG), topographische Grundlage: [NED-12])

Zu bedenken ist, dass die angesprochenen Probleme bei kleinen Signaturen in geringen und mittleren Breiten mit freiem Auge am taktilen Hyperglobus erkennbar sind. So ist je nach Art und Zweck der Visualisierung abzuwägen, welche Methode bzw. welcher Aufwand notwendig ist.

Eine Präsentation von Mike Fooley (GI) weist darauf hin, dass auch Tools von GI diesen Ansatz für kreisförmige Signaturen beherrschen [FOO-08].

4.1.1.3 Visualisierung auf der Tangentialebene

Bei dem oben vorgestellten Sphere Project von Microsoft [BEN-08] wird eine 3D Methode kurz erläutert, die unter der Annahme, dass die Referenzoberfläche lokal einer Ebene entspricht, funktioniert. Ein Symbol wird in der Tangentialebene in der zweiten Dimension visualisiert und anschließend auf die Oberfläche projiziert. [BEN-08] zeigen dies am Beispiel von Fotografien, die als Texturen für rechteckige Dreiecksnetze dienen. Die Eckpunkte eines Dreiecksnetzes, die ursprünglich in der Tangentialebene

der Punktkoordinate liegen, werden anschließend in Richtung des Ursprungs (Erdmittelpunkt) auf die Referenzoberfläche verschoben. Dieser Ansatz lässt sich auf Punktsignaturen übertragen.

4.1.2 Die Wiedergabe von Linien und Flächen

In Bezug auf Linien- und Flächen konnte der Autor nur von GI eine mehr oder weniger konkrete Information erhalten. Laut Mike Fooly ist das GI After Effects Plug-In namens FullGlobe im Stande Liniengeometrien direkt in die Globenabbildung (Disk), unter der Berücksichtigung der Abbildungsverzerrungen, einzuzichnen.

In der Online Hilfe für ArcGIS 10 [ESR-12a] wird explizit erläutert, dass geodätische Puffer nur auf Punktdatensätze anwendbar sind. In der Hilfe der Version 10.1 [ESR-12b] wird diese Beschränkung nicht mehr erwähnt. Es ist anzunehmen, dass nun auch Linien und Flächen geodätisch gepuffert werden können. Die ArcGIS 10.1 Hilfe verweist auf einen Artikel [MUR-99], der sich mit geodätischen Puffern für maritime Grenzlinien auseinandersetzt.

Im Vergleich zur Wiedergabe von Punktgeometrien setzen sich die befragten Globenanbieter weniger mit der korrekten Visualisierung von Linien- und Flächengeometrien auseinander.

4.2 Virtuelle digitale Globen/Earthbrowser

In Kapitel 2.1 wurde die Globenfamilie vorgestellt. Taktile Hypergloben sind eng mit virtuellen Hypergloben verwandt. Und auch die Beziehung zu Geo- oder Earthbrowsern wurde beschrieben. Es liegt also nahe zu besprechen, wie virtuelle digitale Globen bzw. Earthbrowser mit Vektorgeometrien umgehen, diese verarbeiten bzw. visualisieren. Da sich in diesem Fall die Techniken zwischen Earthbrowsern und tatsächlichen virtuellen digitalen Globen (virtuelle Hypergloben) nicht unterscheiden, werden bei der Nennung einer der Bezeichnungen in diesem Kapitel beide adressiert.

Generell werden moderne Earthbrowser mittels 3D Grafik Schnittstellen wie OpenGL oder Direct3D umgesetzt. Vertex-, Geometrie- und Fragment-Shader (kleine Programme für die Programmierung der GPU) werden eingesetzt um das gewünschte Ergebnis zu erreichen. Cozzi und Ring präsentieren mit ihrer Publikation „3D Engine Design for Virtual Globes“ [COZ-11] erstmals ein umfassendes Buch, das sich mit dieser Thematik auseinandersetzt. In mehreren Kapiteln wird auch die Verarbeitung und Visualisierung von geographischen Vektordaten besprochen.

Die Geometrien werden in einem dreidimensionalen kartesischen Koordinatensystem visualisiert. Deshalb müssen die Geometrien, genauer ihre Punkte bzw. Koordinaten (Länge, Breite, Höhe), auf Grundlage des verwendeten Globenkörpers (Kugel, Referenzellipsoid) in das Zielsystem transformiert werden. [COZ-11]

4.2.1 Punktgeometrien

Für die Wiedergabe von Punktinformationen verwenden Earthbrowser sog. Billboards. Hierbei handelt es sich um ein Konzept der Computergraphik. Billboards sind kleine Quadrate, die mit dem gewünschten Symbol bzw. Zeichen (in Rasterform) texturiert werden. Billboards werden darüber hinaus immer in Richtung des Betrachters orientiert. Wird der virtuelle Globus rotiert bzw. verändert die virtuelle Kamera ihren Standpunkt, werden die sichtbaren Billboards, damit ihre Vorderseite wieder orthogonal in Richtung der Kamera orientiert ist, neu ausgerichtet. Die notwendigen Berechnungen werden mit Vertex- bzw. Geometrie-Shadern auf der GPU durchgeführt. [COZ-11]

Mit Billboards wird auch Text ausgegeben, der ständig in Richtung des Betrachters orientiert sein soll. [COZ-11]

4.2.2 Liniengeometrien

In der bereits oben erwähnten KML Spezifikation [OGC-08] wird beschrieben, welche Möglichkeiten das KML Format bietet Linien zu visualisieren. Für die Elemente *LineString* und *LinearRing* werden diese Möglichkeiten durch die Elemente *altitudeMode* und *tessellate* festgelegt. Das erstgenannte Element hat die Ausprägungsmöglichkeiten *relativeToGround*, *absolute* und *clampToGround*, wobei die ersten beiden gleichbedeutend sind. Das boolsche Element *tessellate* kann die Werte *true* oder *false* annehmen.

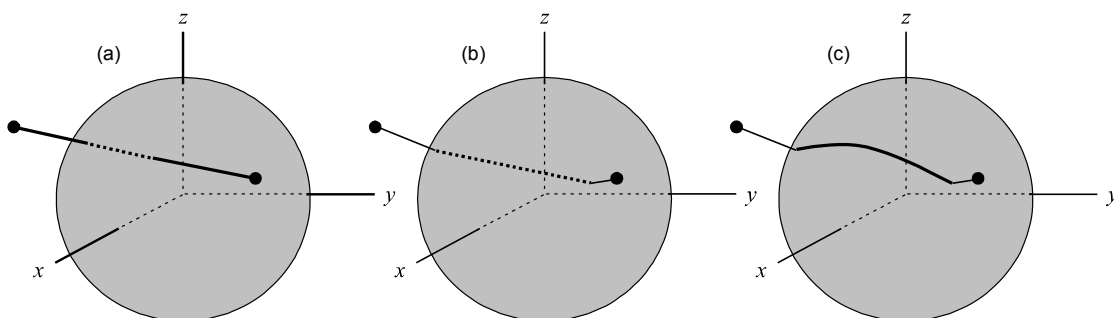


Abbildung 19: KML Interpolationsmodi für die Elemente *LineString* und *LinearRing*

(a) die Punkte werden linear verbunden; (b) die Punkte werden auf die Erdoberfläche projiziert und anschließend linear verbunden; (c) die Punkte werden auf die Erdoberfläche projiziert und anschließend entlang der Erdoberfläche verbunden.

(Quelle: [OGC-08], verändert)

Ist der *altitudeMode relativeToGround* oder *absolute* werden unabhängig von *tessellate* die Punkte durch eine gerade Linie verbunden (Abbildung 19 (a)). Ist der Höhenmodus *relativeToGround* und *tessellate* entspricht *false*, werden die Punkte zuerst entlang dem Vektor in Richtung des Erdmittelpunkts auf die Erdoberfläche projiziert und dann linear verbunden (Abbildung 19 (b)). Nimmt *tessellate* hingegen den Wert *true* an, werden die Verbindungen ebenfalls auf die Erdoberfläche projiziert, die Punkte werden entlang der Erdoberfläche verbunden (Abbildung 19 (c)).

Für die vorliegende Arbeit kommt aufgrund der Eigenschaften des sphärischen Displays nur der letzte Ansatz in Frage. Die Interpolation der Linien muss in Abhängigkeit des zugrunde liegenden virtuellen Globenkörpers (Kugel, Ellipsoid) vorgenommen werden. Handelt es sich um eine Kugel, werden die Punkte eines LineStrings entlang der Orthodrome verbunden. [COZ-11] präsentieren einen Algorithmus für Bildung einer Verbindung zwischen zwei Punkten entlang der geodätischen Linie auf dem WGS84 Ellipsoid. Durch Segmentierung, das Einfügen neuer Stützpunkte, wird die Verbindung an die Oberfläche des Globenkörpers angenähert. Dieser Segmentierungsalgorithmus berechnet die Stützpunkte durch Rotation des Ausgangspunktes um den Normalvektor der durch den Ausgangs-, End- und Erdmittelpunkt aufgestellten Ebene um einen gegebenen Winkel. Je nach dem Winkel, der auch die Genauigkeit der neuen Linie bestimmt, und dem Abstand der zwei Punkte zueinander, entsteht eine gewisse Anzahl an neuen Punkten bzw. Liniensegmenten. Abschließend müssen die neuen Stützpunkte noch auf die Referenzoberfläche projiziert werden. Dieser Schritt entfällt, wenn die Erde durch eine Kugel repräsentiert wird.

Liniengeometrien können nun mit einfachen Linien, die eine Strichbreite von eins haben, visualisiert werden [COZ-11]. Darüber hinaus präsentieren Cozzi und Ring einen Geometry Shader um Linien mit einer größeren Strichstärke zu rendern. Aus den zwei Punkten eines Liniensegments (Start- und Endpunkt) werden vier neue Punkte berechnet. Aus diesen werden zwei Dreiecke generiert, die gemeinsam die Linie visualisieren. Hierbei ist u.a. auch den Verbindungen einzelner Liniensegmente Beachtung zu schenken, insbesondere wenn Außenlinien (Halo Effekt) angezeigt werden sollen. Für die Wiedergabe von Außenlinien linienhafter Geometrien präsentieren [COZ-11] zwei Ansätze. Einerseits die Generierung weiterer Dreiecke (Geometry Shader) links und rechts der ursprünglichen Signatur, die mit der gewünschten Farbe dargestellt werden, andererseits die Texturierung der Dreiecksstruktur mit einer entsprechenden eindimensionalen Textur (Fragment Shader).

4.2.3 Flächengeometrien

Ähnlich wie oben beschrieben hängt laut KML Spezifikation [OGC-08] die Visualisierung von Polygonen vor allem von dem Element *altitudeMode* ab (weniger von dem Element *tessellate*). Der Höhenmodus kann wie bei Liniengeometrien den Wert *relativeToGround* (bzw. *absolute*) oder *clampToGround* annehmen.

Bei *relativeToGround* werden die Eckpunkte durch dreidimensionale Geraden verbunden (wie in Abbildung 19 (a)) und die Fläche anschließend linear gefüllt. Damit das gewünschte Ergebnis erzielt wird, müssen für diesen Modus alle Stützpunkte auf einer gemeinsamen Ebene liegen. In Abbildung 20 (a) ist eine Konsequenz sichtbar: Da diese Polygone nicht der Krümmung der Erdoberfläche folgen, schneiden sie den Globenkörper und liegen teilweise unterhalb der Oberfläche.

Bei *clampToGround* werden die Eckpunkte auf die quadratische Plattkarte projiziert und mit geraden Liniensegmenten verbunden (die Höheninformation wird nicht weiter berücksichtigt) und anschließend in dieser Ebene linear gefüllt. Abschließend wird der Referenzkörper (z.B. Kugel, Ellipsoid, Oberflächenmodell) mit der Plattkarte texturiert.

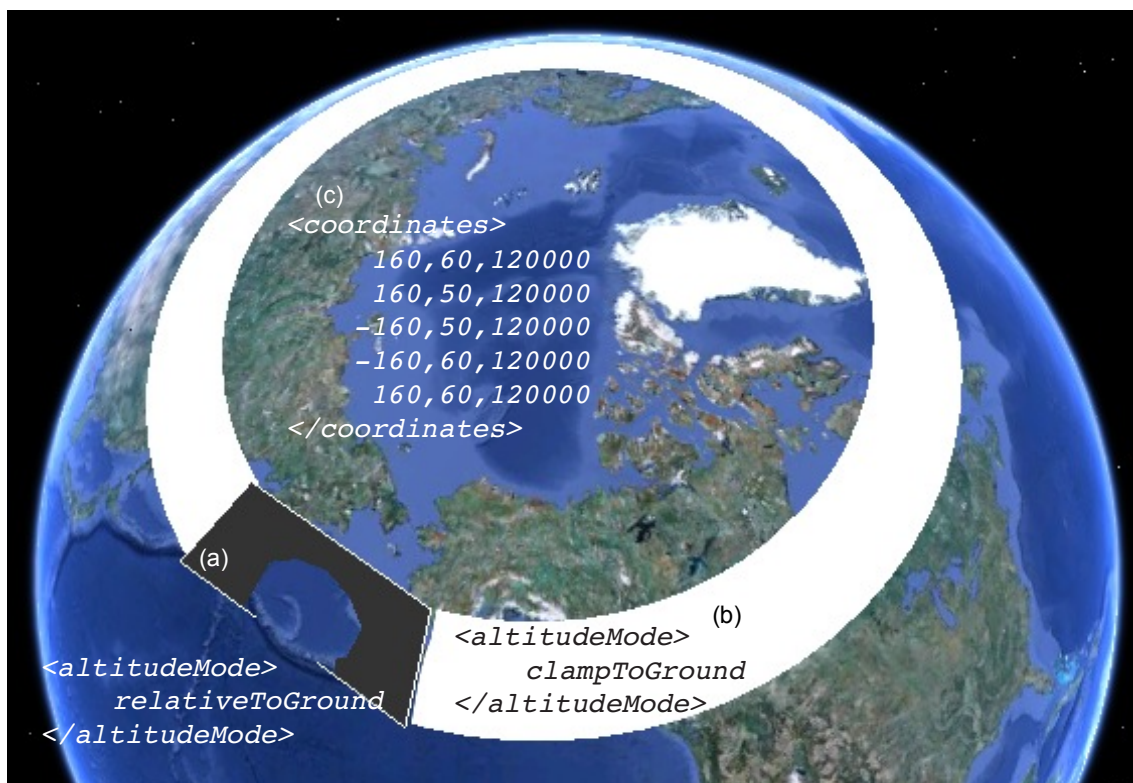


Abbildung 20: KML Interpolationsmodi für Polygone

(a) Polygon mit 4 Eckpunkten und *altitudeMode relativeToGround*; (b) Polygon mit 4 Eckpunkten und *altitudeMode clampToGround*; (c) Koordinaten (Länge, Breite in Grad) und Höhe über Grund (in Meter) beider Polygone
(Quelle: eigene Darstellung, Google Earth)

Bei dem clampToGround Modus können einige Besonderheiten, die unter Umständen unerwünscht sind, auftreten. Erstreckt sich ein Polygon über die Datumsgrenze wie in Abbildung 20, so wird das Polygon (b) nicht über die kürzest mögliche Distanz gezeichnet (über den ± 180 Grad Meridian), sondern über den gesamten Globus. Bei dem Erstellen der Textur werden die Eigenschaften der sphärischen bzw. annähernd sphärischen Oberfläche nicht berücksichtigt. Eine Lösung dafür ist das Trennen des Polygons in zwei Flächen entlang der Datumsgrenze, so wie es in Abbildung 17 für flächige Punktgeometrien gezeigt wird. Auch Polygone in Polbereichen, die z.B. über einen Pol gehen, werden mit diesem Modus nicht korrekt angezeigt, da auch hier bei der Erstellung der Textur die Verzerrungseigenschaften der Platkarte nicht berücksichtigt werden.

Wie bereits oben beschrieben und von Globenanbietern adressiert, müssen auch hier den Polbereichen (Verzerrungseigenschaften der Platkarte) und der „Ost-West-Naht“ bei der Implementierung eine besondere Beachtung geschenkt werden.

Diese letzte Methode wird bei [COZ-11] als „Render to Texture“ bezeichnet. Ein wesentlicher Vorteil ist, das die Polygone durch die Art und Weise der Texturierung mittels Fragment Shader automatisch an die jeweilige Oberfläche (auch bei der Verwendung eines Referenzkörpers mit Höheninformationen) angepasst dargestellt wird. Dadurch ist die Rechenleistung nicht abhängig von der Anzahl der Polygone, sondern von der Anzahl der eingesetzten Ebenen sowie der verwendet Auflösung der Texturen. Diesen Umstand beschreiben die Autoren in weiterer Folge als Nachteil für digitale virtuelle Globen. Je nach der Anzahl der möglichen Zoomstufen und Detailgrade der Globendarstellung werden unterschiedlich hoch aufgelöste Texturen für einen Datensatz benötigt. Das kann zu einem hohen Speicheraufwand führen, kann aber mit LOD (Level Of Detail) Algorithmen umgesetzt werden. Für taktile Hypergloben stellt dieser Umstand kein großes Problem dar, da durch das Wegfallen unterschiedlicher Zoomstufen eine Textur ausreicht (in Abhängigkeit der Projektoren Auflösung).

Einen weiteren Ansatz, den [COZ-11] ausführlich besprechen, ist die Tessellation von Polygonen zu Dreiecksnetzen. Hierbei wenden sie vier Prozesse an um von den geographischen Vektordaten zu Polygonen, die entlang der Oberfläche des virtuellen Globenkörpers visualisiert werden, zu gelangen. Es handelt sich um: die Bereinigung der Vektordaten, die Triangulierung der Polygone, eine weitere Untergliederung der Dreiecke und die Projektion der Eckpunkte auf die Oberfläche des Globus.

Da geographische Vektorformate in unterschiedlichsten Formaten vorliegen (Kapitel 3.3.1) und diese Vektordaten unterschiedlich speichern und verwalten, muss eine gewisse Vereinheitlichung vorgenommen werden, damit die Ausführung der Folgeschritte gewährleistet bleibt. Doppelte Punkte müssen aus der Liste der Stützpunkte gelöscht werden, da solche bei der Triangulierung zu degenerierten Dreiecken führen können. Bearbeitet der Triangulierungsalgorithmus „Simple Features“ (Kapitel 3.1.1), dann müssen sich selbst schneidende Polygone abgefangen werden. Des Weiteren muss die Drehrichtung (Uhrzeigersinn bzw. gegen den Uhrzeigersinn) ermittelt und ggf. invertiert werden.

Anschließend wird das bereinigte Polygon in Dreiecke unterteilt. Es entsteht ein Dreiecksnetz. [COZ-11] besprechen einen Triangulierungsalgorithmus, bei dem keine neuen Punkte hinzugefügt werden, denn jedes einfache Polygon mit n Eckpunkten kann immer in $n - 2$ Dreiecke untergliedert werden. Cozzi und Ring besprechen und erweitern dafür die sog. Ear Clipping Methode in der Ebene (2D) für das Ellipsoid (3D).

Ist das Dreiecksnetz nun generiert, wird dieses im nächsten Schritt verfeinert. Mit einem rekursiven Ansatz und einer Abbruchsbedingung wird getestet, ob ein Dreieck nochmals unterteilt werden muss oder den Kriterien genügt. Würde hier nicht die notwendige Feinheit des Gitternetzes erreicht werden, könnte es passieren, dass zu lange Kanten beim Rendern das Referenzellipsoid schneiden und somit Dreiecksteile unterhalb der Erdoberfläche liegen.

Im letzten Schritt der Polygonaufbereitung werden alle Punkte, die ursprünglichen Eckpunkte sowie die neu generierten, in Richtung des Erdmittelpunktes auf die Referenzoberfläche projiziert. Jetzt ist das Polygon fertig bearbeitet und kann gerendert werden.

[COZ-11] besprechen abschließend mögliche Modifizierungen und Optimierungen des besprochenen Ansatzes. In Kapitel 6.3 wird nochmals etwas genauer auf die Triangulierung von Polygonen auf der Kugel eingegangen.

5 Die Wiedergabe von Vektordaten am sphärischen Display

5.1 Kriterien zur Methodenklassifizierung der Vektordatenintegration

Anhand des Prozesses der Vektordatenintegration in das Globensystem werden Unterscheidungskriterien für die Klassifizierung möglicher Methoden besprochen.

Folgende Kriterien werden im Folgenden besprochen:

- Implementierung der Visualisierung: intern vs. extern
- Zeitpunkt der Bilderzeugung: synchron vs. asynchron
- Objektraum der Visualisierung: 2D vs. 3D
- Ausgabe- bzw. Speicherformat der Visualisierung

Abbildung 21 zeigt ein Klassifizierungsschema der Vektordatenintegration nach den genannten Kriterien, das herangezogen werden kann um die einzelnen Methoden zu kategorisieren. In Abschnitt 5.1.5 wird dieses Schema auf die bisher besprochenen Technologien angewandt.

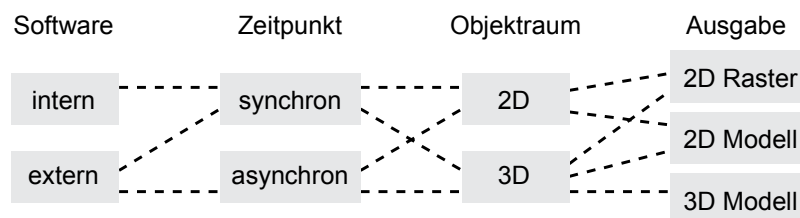


Abbildung 21: Klassifizierungsschema zur Vektordatenintegration am sphärischen Display

Schrittweise Kategorisierung von links nach rechts entlang den gestrichelten Linien
(Quelle: eigene Darstellung)

Als entscheidendes Kriterium für diese Arbeit wird der Objektraum betrachtet, da dieser den virtuellen Ort der Visualisierung und somit deren Methode bestimmt.

5.1.1 Implementierung der Visualisierung: intern vs. extern

Ist die Globensoftware im Stande Vektordaten zur Laufzeit zu verarbeiten und zu visualisieren, spricht der Autor von einer internen Implementierung der Vektordatenintegration. Werden jedoch die Vektordaten von bzw. mit Hilfe einer anderen Anwendung für das Globensystem aufbereitet, so passiert das extern – unabhängig von der Globensoftware.

In Abbildung 22 wird eine interne Vektordatenverarbeitung schematisch dargestellt. Als Eingabe gibt es auf der einen Seite einen Rasterdatensatz, auf der anderen einen Punktvektordatensatz, der als Overlay visualisiert werden soll. Die Globensoftware liest im ersten Schritt die Vektordaten ein, wendet Algorithmen auf die Vektordaten an, die notwendig für die anschließende Visualisierung sind. Die einzelnen Vektor- und Rasterebenen werden kombiniert und für die Ausgabe am Globus aufbereitet.

Das Beispiel zeigt ein Schema einer Globensoftware, die die Vektordaten als Platkarte aufbereitet. Genau so gut könnten die Vektordaten in die fertige azimutale Abbildung gezeichnet werden, wiederum unter Berücksichtigung der Verzerrungseigenschaften.

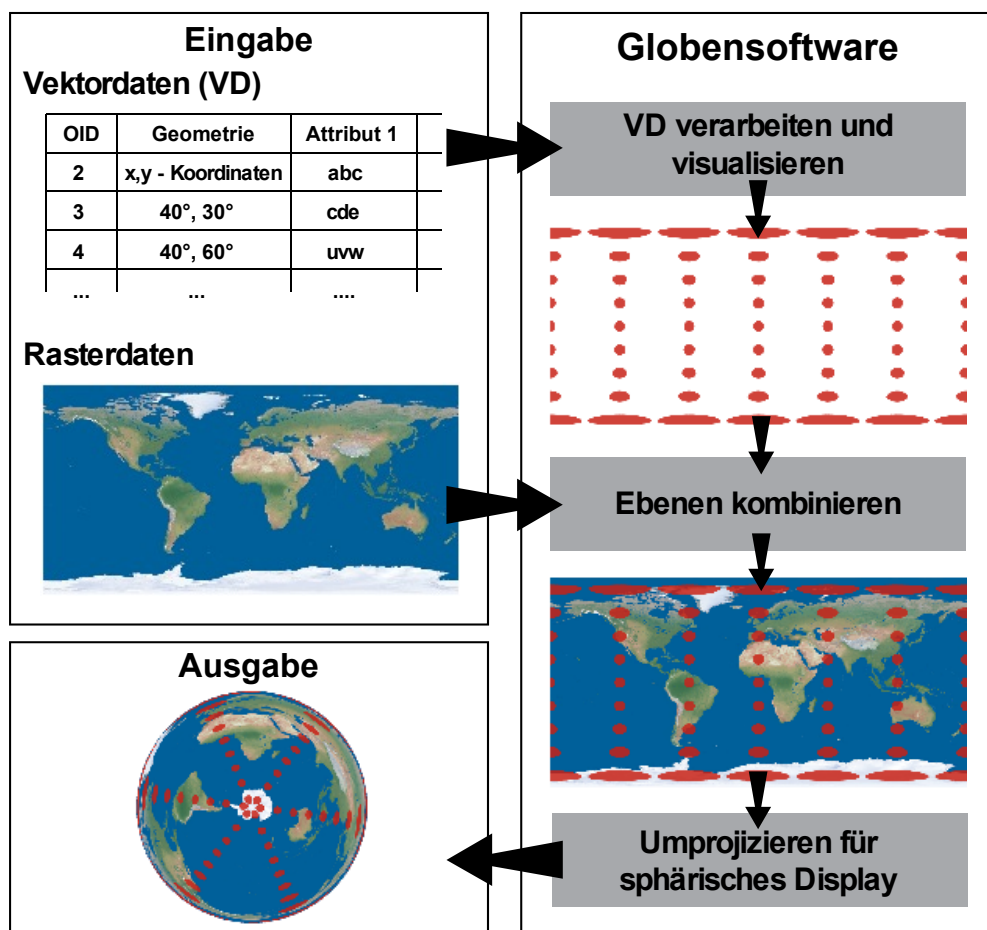


Abbildung 22: Schema der internen und synchronen Vektordatenintegration
Vektordatenintegration innerhalb der Globensoftware (exemplarisch mit Platkarte als Basismedium)

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

Bei externen Methoden der Vektordatenaufbereitung (Abbildung 23) werden die Daten nicht von der Globensoftware direkt verarbeitet. Die Daten werden automatisch oder manuell mittels einer anderen Software verarbeitet und visualisiert. Die unterschiedlichen Ebenen können in diesem Fall bereits am Ende der

Vektordatenaufbereitung, mittels der Globensoftware oder dazwischen (in einem weiteren unabhängigen Schritt) kombiniert werden. Die Globensoftware ist primär wiederum für die Generierung der Globenabbildung verantwortlich.

5.1.2 Visualisierungszeitpunkt: synchron vs. asynchron

Die Aufbereitung der Vektorgeometrien kann einerseits zur Laufzeit der Globensoftware (synchron), andererseits vor der Laufzeit (asynchron) statt finden. Ist der Prozess in die Globensoftware integriert (intern), ist der Zeitpunkt immer als synchron zu bezeichnen. Bei externer Visualisierung ist der Zeitpunkt synchron, wenn die Vektordaten von einem weiteren Programm während der Laufzeit der Globensoftware verarbeitet und visualisiert und in Echtzeit an die Globensoftware übergeben werden. Sind jedoch die Vektordaten vor der Laufzeit der Globensoftware für diese bereits aufbereitet, ist der Darstellungszeitpunkt asynchron. In Abbildung 23 wird dieser Zeitaspekt durch die Stoppuhr verdeutlicht.

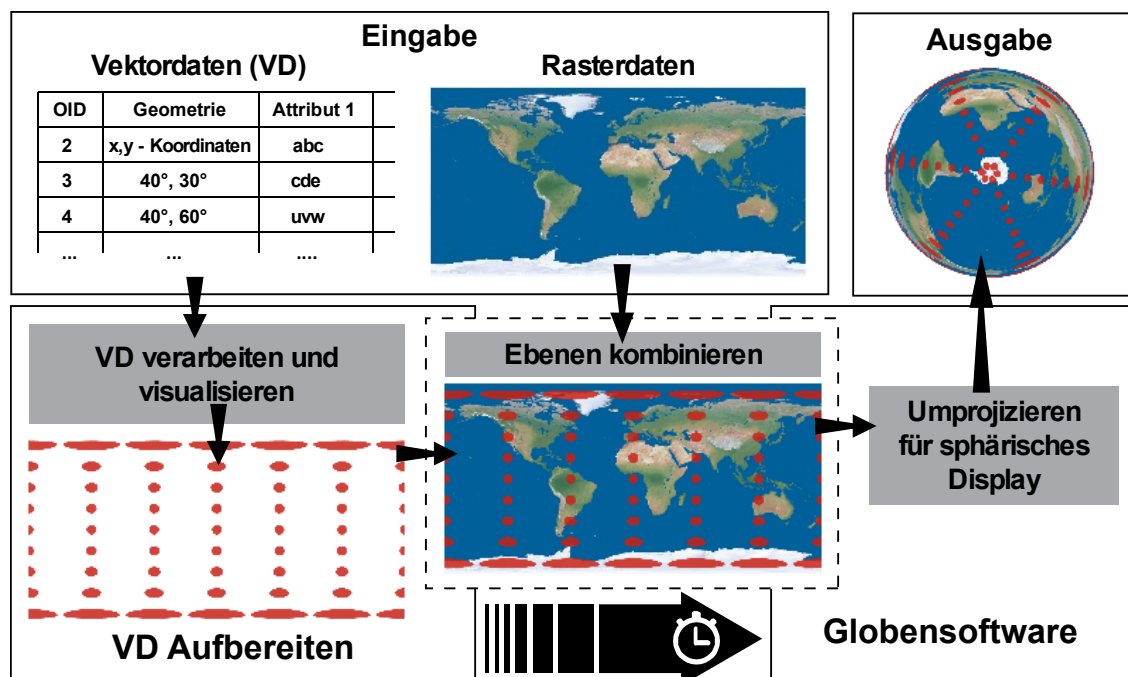


Abbildung 23: Schema der externen synchronen bzw. asynchronen Vektordatenintegration

Vektordatenaufbereitung außerhalb der Globensoftware (exemplarisch mit Platkarte als Basismedium) mit Unterscheidung des Zeitpunktes der Datenvisualisierung (vor oder zur Laufzeit der Globensoftware)

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

5.1.3 Objektraum der Visualisierung: 2D vs. 3D

Wie bereits beim Kapitel der Globensoftware erläutert wurde, gibt es Systeme, die zur Visualisierung der Daten ein ausschließlich zweidimensionales Koordinatensystem

verwenden und solche, die auch ein dreidimensionales Koordinatensystem einsetzen um die benötigte Projektion für die Wiedergabe der Daten am sphärischen Display zu erzeugen. Folglich ist auch die Vektordatenintegration in beiden Objekträumen möglich.

Bei der Integration der Vektordaten in der Ebene kann folgender Unterschied getroffen werden: Einerseits können die Vektordaten in einer kartographischen Projektion, üblicherweise die quadratische Platkarte, unter Berücksichtigung der Verzerrungseigenschaften aufbereitet werden (Abbildung 24 (b)). Andererseits können die Vektordaten direkt in die azimutale Globenabbildung integriert werden (Abbildung 24 (a)). Eine große Herausforderung hierbei ist die exakte mathematische Definition der azimutalen Projektion, da diese je nach Globushardware immer wieder unterschiedlich ist und somit die Verzerrungen auch hier ein (unbekanntes) Problem darstellen. In beiden Fällen kommt es auch auf den Verwendungszweck bzw. die Datenausbreitung an, ob die jeweilige Methode aufgrund der Verzerrungsprobleme bedenkenlos eingesetzt werden kann. Das trifft solange zu, als das Ergebnis am Globus nicht verfälscht dargestellt wird. [BEN-08]

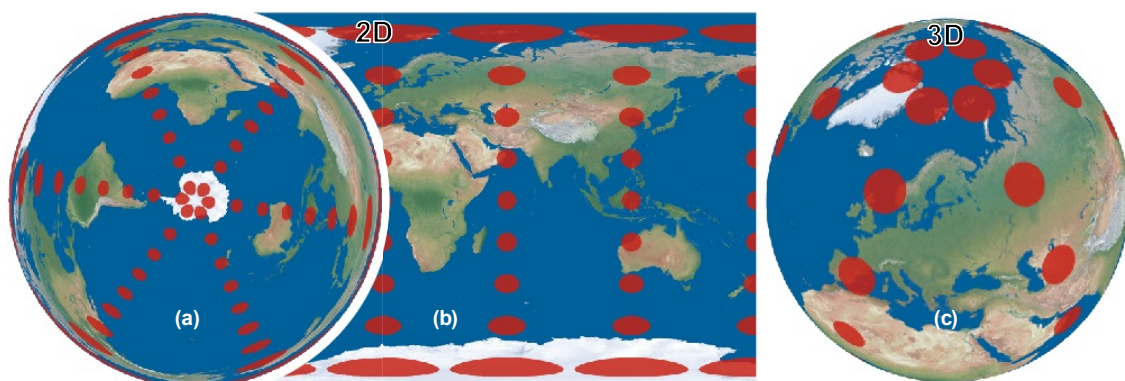


Abbildung 24: Objektraum der Vektordatenintegration: 2D vs. 3D

(a) Wiedergabe der Vektordaten im Objektraum 2D direkt auf der Globenprojektion; (b) Wiedergabe der Vektordaten im Objektraum 2D auf der Platkarte; (c) Wiedergabe der Vektordaten im Objektraum 3D entlang der Kugeloberfläche.

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

Die Verzerrungsprobleme der Platkarte fallen bei 3D Ansätzen weg und müssen nicht gesondert berücksichtigt werden. Die geographischen Koordinaten werden bei diesen Ansätzen in dreidimensionale kartesische Koordinaten überführt, so dass sie auf einer Einheitskugel bzw. einer Kugel mit einem gewissen Radius liegen (Abbildung 24 (c)). Auf welche Art und Weise die dreidimensionale Darstellung in die radiale 2D Abbildung umgerechnet wird, hängt von der Globensoftware ab.

In Kapitel 5.4 werden die Besonderheiten, die bei der Visualisierung von Vektordaten im Objektraum 2D bzw. 3D zu berücksichtigen sind, näher erläutert.

5.1.4 Ausgabe- bzw. Speicherformat der Visualisierung

In einem letzten Kriterium soll das Speicher-, Ausgabe- bzw. Transferformat, das die visualisierten Vektordaten annehmen, unterschieden werden, bzw. wird identifiziert, wann der Prozess der Rasterung durchgeführt wird.

Im Folgenden werden drei mögliche Formate, die bei der Vektordatenvisualisierung entstehen, unterschieden: 2D Basisraster, 2D Modelle sowie 3D Modelle.

Werden die Vektordaten in der Projektion des Basismediums (Kapitel 2.2.1) visualisiert (die Art und Weise ist unerheblich), steht am Ende des Prozesses ein 2D Basisraster, der von der Globensoftware verarbeitet werden kann.

2D Modelle sind zweidimensionale Polygonnetze bzw. Dreiecksnetze (Meshes), die durch Grafikprogrammierung generiert werden können. Punkte, Linien und Flächen werden durch aneinander gereihte Dreiecke mit Hilfe von Geometrie- und Vertex-Shadern in der Basisprojektion oder der Globenprojektion selbst wiedergegeben. Zu den 2D Modellen zählen aber auch Vektordarstellungen, die unter Berücksichtigung der Verzerrungseigenschaften der verwendeten Projektion in Vektorformaten wie bspw. dem SVG Format gespeichert sind. Die größten Hürden bei solchen Darstellungen sind die Randbereiche der Projektionen. Bei der Platkarte sind es die Datumsgrenze und die Pole, bei der azimutalen Globendarstellung der Bereich des maximalen Radius.

Unter 3D Modellen im Kontext der Diplomarbeit sind Dreiecksnetze zu verstehen, die entlang der Erd- bzw. virtuellen Kugeloberfläche im dreidimensionalen Raum die geographischen Punkt- Linien- und Flächendaten wiedergeben. Neben 3D Meshes können 3D Modelle u.a. auch mit konstruktiver Festkörpergeometrie (gebräuchlicher: Constructive Solid Geometry, CSG) erzeugt werden.

Alle genannten Ergebnisse können als Datei gespeichert werden. 2D Basisabbildungen im Rasterformat (png, jpg, dds), 2D oder 3D Modelle mit entsprechenden Formaten als Polygonnetze. Dabei ist zu beachten, dass die Globensoftware in weiterer Folge diese Daten verarbeiten kann. Die Generierung von Dateien wird vor allem bei externen und asynchronen Methoden verwendet, kann aber auch bei internen Ansätzen eingesetzt werden. Insbesondere bei internen bzw. auch externen synchronen Ansätzen können die erzeugten Darstellungen des Weiteren im Arbeits- bzw. Grafikspeicher verwaltet werden. Dabei wird keine physische Datei erzeugt. In diesem Zusammenhang ist auch Screen Capturing zu erwähnen. OmniSuite (HRG/Globoccess) und OnStage (Global Imagination) können Bereiche des Bedienungsmonitors aufnehmen und direkt auf dem Globus visualisieren. Vektordaten werden also extern, synchron oder asynchron, zwei- oder dreidimensional visualisiert und als 2D Basisdarstellung am Monitor angezeigt und

von der Globensoftware durch Screen Capturing aufgenommen und in die Globenprojektion übergeführt.

Von der Methode der Vektordatenintegration und der technischen Umsetzung abhängig ist der Zeitpunkt der Rasterung der Vektordaten. Ist ein 2D Basisraster das Ergebnis, so werden die Vektordaten beim Erzeugen beim Eintrag in bspw. eine Platkarte gerastert. Werden Polygonnetze erzeugt (intern oder extern) und von der Globensoftware verarbeitet, wird die Rasterung idR. im Zuge des Rasterprozesses der Grafikpipeline durchgeführt. Diese Rasterung wird von der GPU durchgeführt und dient zur Wiedergabe an einem Ausgabegerät (in diesem Fall ein Projektor eines Hyperglobus). Würde OmniSuite durch eine interne 3D Vektordatenintegration erweitert, so würde das 3D Modell bereits bei der Erstellung der CubeMap für das Environment Mapping (Kapitel 2.2.3.5) gerastert.

5.1.5 Beispiele

Die bisher besprochenen Technologien lassen sich ebenfalls in dieses Schema einordnen. Virtuelle Globen verarbeiten Vektordaten (z.B. KML-Dateien) intern und synchron. Der Objektraum kann je nach Geometrie und Definition (vgl. Kapitel 4.2) zwei- oder dreidimensional sein. „Render to Texture“ Algorithmen verwenden idR. den 2D Raum. Als Ergebnis entsteht ein 2D Basisraster, mit dem anschließend die virtuelle Kugel texturiert wird. Werden Punkte, Linien und Flächen als 3D Dreiecksnetze entlang der Kugeloberfläche modelliert, so passiert dies im virtuellen 3D Raum. (Abbildung 25 (a))

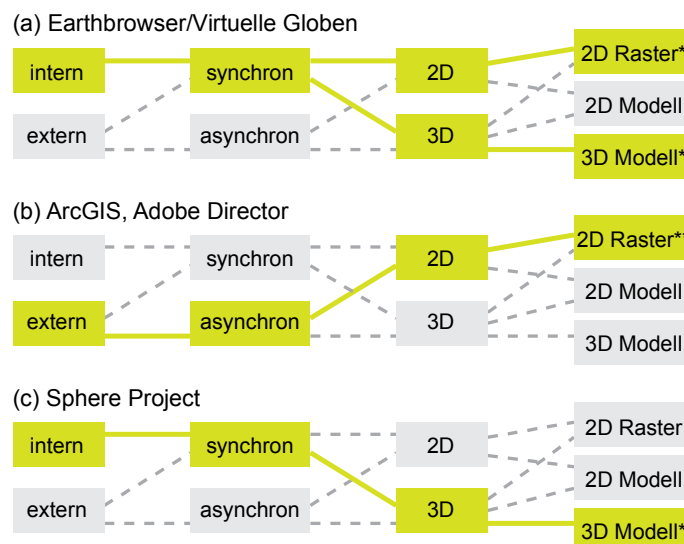


Abbildung 25: Klassifizierungsbeispiele der Vektordatenintegration

in: (a) Earthbrowser/Virtuelle Globen, (b) ArcGIS oder Adobe Director, (c) Sphere Project

* = virtuell im Speicher; ** = als physische Datei

(Quelle: eigene Darstellung)

Die zuvor besprochenen Beispiele in Adobe Illustrator (Kontinentaldrift) und in ArcGIS (Atomkraftwerke) sind externe, asynchrone Methoden, die zur Visualisierung den zweidimensionalen Raum verwenden. Das Ergebnis ist jeweils ein 2D Basisraster, der als physische Datei vorliegt und von der Globensoftware später verarbeitet werden kann. (Abbildung 25 (b))

Als dritte Methode wird hier der unter 4.1.1.3 besprochene Ansatz der Visualisierung von kleinen punkthaften Signaturen bzw. Meshes in der Tangentialebene klassifiziert. Die Herangehensweise bei dieser Technologie, so wie sie von [BEN-08] dokumentiert ist, ist intern und somit auch synchron, verwendet den dreidimensionalen Raum und generiert ein entsprechendes 3D Modell im Grafikspeicher. (Abbildung 25 (c))

5.2 Dreieinhalb Ansätze unter der Lupe

Im Folgenden werden vier unterschiedliche Verfahren zur Vektordatenintegration besprochen. Da zwei Methoden eng zusammenhängen, spricht der Autor von dreieinhalb Ansätzen.

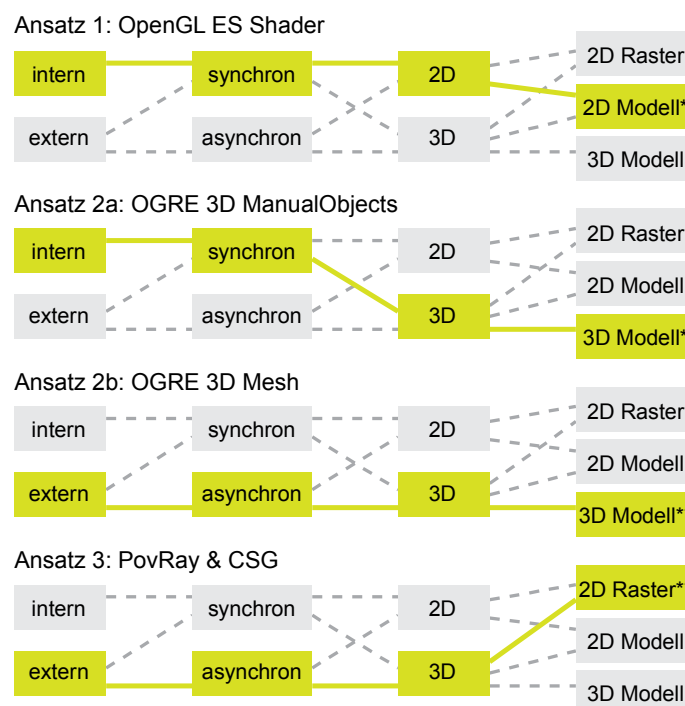


Abbildung 26: Klassifizierung unterschiedlicher Ansätze der Vektordatenintegration

Ansatz 1: OpenGL ES Shader und sphärische Trigonometrie; Ansatz 2a und 2b: die Integration mittels OGRE3D; Ansatz 3: PovRay und CSG; * = virtuell im Speicher; ** = als physische Datei (Quelle: eigene Darstellung)

In Abbildung 26 werden diese dreieinhalb Methoden nach der oben eingeführten Klassifizierung unterschieden. Ansatz 1 ist ein interner, folglich synchroner Ansatz, der

im zweidimensionalen Objektraum ein 2D Modell der Vektordaten generiert. Neben sphärischer Trigonometrie kommen bei diesem Verfahren OpenGL ES Shader zum Einsatz. Das resultierende 2D Modell wird im Grafikspeicher verwaltet.

Die Ansätze 2a und 2b adressieren die Vektordatenvisualisierung mit Ogre3D (OmniSuite verwendet diese Grafikengine). Bei beiden Methoden werden die Vektordaten entlang der Kugeloberfläche im dreidimensionalen Raum als 3D Modell (Mesh) gezeichnet. Bei 2a wird das Dreiecksnetz im Speicher verwaltet, bei 2b als eigene Datei (*.mesh). Die Methoden unterscheiden sich in der Implementierung (intern/extern) und im Zeitpunkt (synchron/asynchron) der Datenvisualisierung.

Der letzte Ansatz ist extern und asynchron. Im 3D Raum werden die Vektordaten mittels CSG generiert. Auch hier ist die Kugeloberfläche der Zeichenträger. Als Ergebnis entsteht ein zweidimensionaler Basisraster (z.B. Plattkarte) als physische Bilddatei.

Anhand dieser dreieinhalb Ansätze werden in den folgenden Kapiteln die Aufbereitung sowie die Visualisierung der Vektordaten diskutiert. Darauf basierend werden die Anforderungen, die an die Daten zu stellen sind, ermittelt.

Unabhängig von der Methode kann der Prozess der Vektordatenverarbeitung durch die Pipeline in Abbildung 27 beschrieben werden.

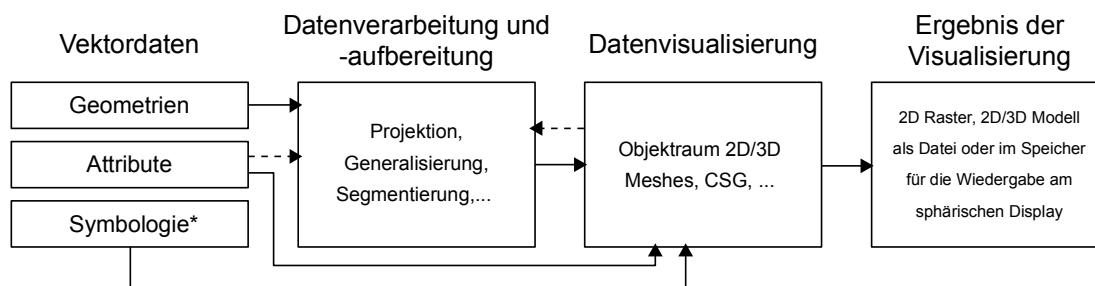


Abbildung 27: Prozesspipeline der Vektordatenvisualisierung für das sphärische Display
Vektordaten werden bearbeitet und im Schritt der Datenvisualisierung für die Wiedergabe am sphärischen Display aufbereitet.

* = Stildefinition innerhalb der Vektordaten gespeichert oder manuell anzugeben
(Quelle: eigene Darstellung)

Die darzustellenden Vektordaten, bestehend aus Geometrien, Attributen und Stildefinitionen, stehen am Anfang der Pipeline. Die Stildefinitionen können in Abhängigkeit des Vektorformats innerhalb der Vektordaten gespeichert sein oder müssen manuell definiert werden (Kapitel 3.3).

Die Vektorgeometrien werden in einem ersten Schritt verarbeitet und aufbereitet. Diese Verfahren sind abhängig von der eingesetzten Methode der Datenvisualisierung (Ansätze 1 – 3) sowie, wenn festgelegt, von Attributen der Vektordaten. Darüber hinaus spielt hier der Geometrietyp eine sehr wichtige Rolle. Punktdaten werden bspw. (auch je nach Ansatz) anders aufbereitet als Linien- oder Flächengeometrien. Zu den vorbereitenden Schritten (Kapitel 5.3) zählen unter anderem: Einlesen der Geometrien, Identifizierung der Ebenen und Selektion der darzustellenden Features, Definition der Projektion, maßstabsabhängige Generalisierung sowie Verfeinerung der Geometrien in Abhängigkeit des eingesetzten Visualisierungsverfahrens.

Die Datenvisualisierung (Kapitel 5.4) ist neben der Bearbeitung der Daten der wichtigste Bestandteil in dem Prozessablauf. Unter Berücksichtigung der Symbologie und, falls festgelegt, bestimmter Attribute werden je nach Verfahren die zuvor aufbereiteten Geometrien visualisiert. Je nach Ansatz können einzelne Algorithmen Teil der Datenaufbereitung oder der Datenvisualisierung sein.

Am Schluss der Pipeline steht die Ausgabe der generierten Darstellungen. Die möglichen Ergebnisse der Visualisierung wurden bereits oben erläutert.

In Anlehnung an die Terminologie von Vektorgeometrien (Kapitel 3) werden je nach Prozess in der Pipeline unterschiedliche Bezeichnungen für die Geometrien verwendet. Die bereits selektierten Features aus dem Originaldatensatz werden als Ursprungsgeometrien bezeichnet. Objektgeometrien sind bereits generalisierte bzw. in Folge verfeinerte Geometrien. Schließlich gibt es noch die Visualisierungsgeometrien. Sie entsprechen den Geometrien der Signaturen, die im Schritt der Datenvisualisierung erzeugt werden (Festkörper oder Meshes).

Die unterschiedlichen Verfahren werden im Zuge der Diplomarbeit theoretisch erörtert und nur ansatzweise implementiert. Die folgenden vier Abschnitte beleuchten den technischen Hintergrund, auf dem die unterschiedlichen Methoden basieren.

5.2.1 Ansatz 1: OpenGL ES Shader und sphärische Trigonometrie (2D/3D)

Bei diesem internen und synchronen Ansatz wird ein 2D Modell der Vektordaten generiert, das direkt in die Globenabbildung integriert wird, also direkt in das Bild für den Projektor. Hierzu wird ein Vertex Shader verwendet (konkret: OpenGL ES Shader).

Für diese Methode muss die mathematische Projektionsformel der Globenabbildung bekannt sein (z.B. wie bei SOSG, Kapitel 2.2.3.4). Mit der inversen Projektion wird das Basisrasterbild mit Hilfe eines Fragment Shaders erstellt. Die Vorwärtsprojektion wird

verwendet um die zuvor aufbereiteten Vektordaten in die fertige Abbildung zu integrieren.

Um die Verzerrungen zu berücksichtigen, die die Globenabbildung aufweist, wird das 2D Modell mittels sphärischer Trigonometrie auf Basis der geographischen Koordinaten der Vektordaten erstellt. So kann bspw. eine Liniengeometrie mit veränderbarer Strichstärke modelliert werden.

5.2.2 Ansatz 2a: Ogre3D ManualObject (3D)

ManualObject ist eine Klasse von Ogre3D, die es erlaubt benutzerdefinierte zwei- oder dreidimensionale Objekte zur Laufzeit zu generieren (intern und synchron). In diesem Fall werden ausgehend von den Vektorgeometrien dreidimensionale Meshes generiert, die möglichst genau an die virtuelle Kugeloberfläche angenähert werden.

Beide Ansätze, 2a und 2b, sind als Schnittstelle zum Softwarepaket OmniSuite gedacht. Wird die Funktionsweise von OmniSuite (Kapitel 2.2.3.5) berücksichtigt, so ergibt es Sinn die Daten, räumlich gesehen, innerhalb der texturierten virtuellen Kugel zu visualisieren. Das virtuelle Kamera/Spiegel System generiert automatisch die Abbildung, die vom Projektor projiziert werden soll, wobei nicht nur die Basistextur, sondern auch die Vektorgeometrien enthalten sind.

Die ManualObject Klasse ist vergleichsweise ein einfaches Interface zur Erzeugung von Dreiecksnetzen. Sie stellt eine Möglichkeit dar, die direkte und oft aufwendige Programmierung von Vertex Shadern (so wie in Ansatz 1) zu umgehen.

5.2.3 Ansatz 2b: Ogre3D Mesh (3D)

Ogre Mesh ist eine weitere Ogre Klasse bzw. ein Ogre eigenes, binäres Dateiformat, das derartige Meshes (die Geometrie von dreidimensionalen Dreiecksnetzen) verwalten kann. Bei diesem externen und asynchronen Ansatz werden die Vektorgeometrien mit einem eigenen Programm aufbereitet und in Meshes überführt, die in einem von Ogre definierten XML Format gespeichert werden. Die Struktur ist in der Ogre MeshXML Document Type Definition (DTD) dokumentiert. Anschließend kann mit dem Kommandozeilenprogramm OgreXMLConverter die XML Datei in das binäre Format umgewandelt werden. Die dadurch generierte *.mesh Datei kann zur Laufzeit der Globensoftware in die virtuelle Szene geladen werden. Für ein vorzeitiges Betrachten können sog. Meshviewer verwendet werden.

Darüber hinaus gibt es eine Methode um Objekte, die mittels der ManualObject Klasse generiert wurden, in ein Ogre Mesh zu konvertieren. [O3D-12]

5.2.4 Ansatz 3: PovRay und CSG (3D)

PovRay (Persistence of Vision Raytracer) ist ein Programm um 3D Grafiken zu rendern. Dazu bietet der Raytracer eine eigene Skriptsprache an, mit der die zu erstellende Szene (Objekte, Licht, Kamera, ...) definiert wird (Scene Description Language, SDL). Darüber hinaus hat PovRay umfangreiche mathematische Funktionen (Analytische Geometrie, Vektorrechnung). Raytracing hat wenig gemein mit den Techniken, die Grafikprozessoren verwenden um 3D Szenen zu rendern. [POV-12]

Bei diesem Ansatz werden die Vektorgeometrien mit Hilfe von konstruktiver Festkörpergeometrie (Constructive Solid Geometry, CSG) ebenfalls entlang der Kugeloberfläche visualisiert.

Mit einer sphärischen Kamera (Öffnungswinkel von 180° bzw. 360°) im Kugelmittelpunkt (Koordinatenursprung) kann eine quadratische Platkarte mit beliebiger Auflösung gerendert werden. Da die Kamera im Inneren der Kugel sitzt, müssen die Vektordaten für ein korrektes Ergebnis spiegelverkehrt visualisiert werden. Die Platkarte kann in weiterer Folge als Overlay verwendet werden. Dieser externe und asynchrone Ansatz verwendet also den dreidimensionalen Raum zur Datenvisualisierung und den zweidimensionalen Raum zur Übergabe der Daten an die Globensoftware.

Verglichen mit den anderen Ansätzen, die oben vorgestellt wurden, hat sich der Autor mit diesem dritten Ansatz am intensivsten auseinander gesetzt.

5.3 Aufbereitung von Vektordaten für die Wiedergabe am sphärischen Display und die damit verbundenen Anforderungen an die Daten

Bevor die Vektordaten mit den zuvor vorgestellten Methoden visualisiert werden können, müssen die Geometrien eingelesen werden und unterschiedliche vorbereitende Algorithmen durchlaufen (Abbildung 28).

Nachdem die Geometrien eingelesen wurden, müssen einzelne (kartographische) Ebenen identifiziert werden und etwaige Selektionen (auf Attribute basierend oder manuell) vorgenommen werden. Da Vektordaten alle möglichen Projektionen haben können, muss im nächsten Schritt kontrolliert werden, ob die Daten in einer geographischen Projektion (Länge und Breite in Dezimalgrad) vorliegen. Wenn nötig werden die Geometrien um-projiziert. Nach einer nicht zwingenden, aber sinnergebenden maßstabsabhängigen Generalisierung muss je nach Darstellungsmethode eine Verfeinerung der Geometrien vorgenommen werden.

Schließlich können die Geometrien im Schritt der Datenvisualisierung weiterbearbeitet werden.

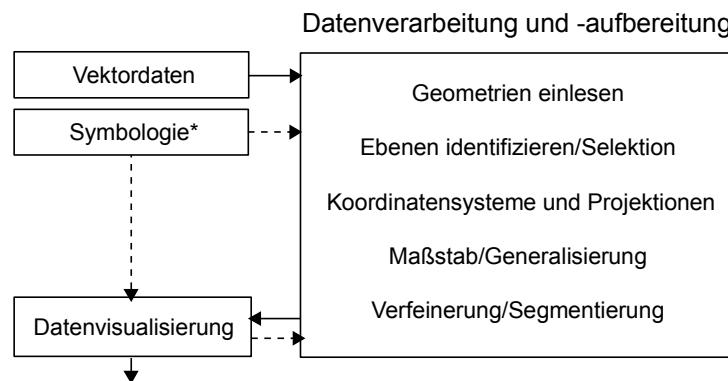


Abbildung 28: Schritte der Datenverarbeitung und -aufbereitung

Geometrien werden eingelesen, die Ebenen und Geometrietypen identifiziert und ggf. selektiert und um-projiziert. Die Daten können in Abhängigkeit des Maßstabes generalisiert werden und in Bezug auf die eingesetzte Visualisierungsmethode verfeinert werden.
(Quelle: eigene Darstellung)

All diese Voraussetzungen, Vorüberlegungen und Vorbearbeitungen, die u.a. vom eingesetzten Visualisierungsverfahren abhängen, definieren gemeinsam Anforderungen, die an Vektordatensätze für die Wiedergabe am sphärischen Display zu stellen sind. In diesem Zusammenhang wird abschließend auf die Globenwürdigkeit – in diesem Fall von Vektordatensätzen – eingegangen. Davor werden die gerade eingeführten Bearbeitungsschritte etwas genauer beleuchtet.

5.3.1 Geometrien einlesen

Eine Möglichkeit Vektordatensätze einzulesen bietet die offene C++ Bibliothek OGR Simple Features Library [OGR-12]. OGR ist ein Teil der Geospatial Data Abstraction Library (GDAL), die zum Einlesen und Verarbeiten von Rasterdaten gedacht ist.

OGR weist folgende, in der Geoinformation bekannte Architektur auf. Die Quelldatei (Data Source) besteht aus einem oder mehreren Layer Objekten. Ein Layer besteht wiederum aus Features, den einzelnen Geoobjekten (Punkte, Linien und Flächen verknüpft mit Attributen). Die Geometrien eines Features werden gesondert in der Geometry Klasse behandelt. Die wesentlichen Geometrietypen sind Point, LineString und Polygon (Kapitel 3.1). Ein Punktfeature besteht aus einer Koordinate, Linien- und Flächengeometrien aus einer Liste von Koordinaten.

Beruhend auf dieser Struktur handelt es sich beim Einlesen der Vektorgeometrien um einen iterativen Prozess, wobei die weiteren Schritte (Selektion und Bestimmen des Bezugssystems) ebenfalls beim Einlesen implementiert werden können.

5.3.2 Ebenen identifizieren/Selektion

In Anlehnung an die Verwendung von Kartenebenen in der Geoinformation sollen auch hier einzelne Darstellungsebenen identifiziert werden. Features einer Ebene weisen einen einheitlichen Geometrietyp, eine einheitliche Projektion und Ausdehnung, einen inhaltlichen Zusammenhang und eine graphische Einheit auf [KRI-12].

Bei Shape Dateien, die immer nur einen Geometrietyp verwalten, ist das Einlesen und Bestimmen der Ebenen sehr einfach. Bei KML Dateien, bei denen alle Geometrietypen in einem Datensatz vorkommen können, muss beim Einlesen und Identifizieren einzelner Ebenen mehr achtgegeben werden.

Durch die Unterteilung in unterschiedliche (thematische, graphische) Ebenen, können diese durch die Globensoftware als einzelne Overlays angesprochen werden und ein- bzw. ausgeblendet werden. Vor allem bei internen Ansätzen ist bei der Visualisierung auf die Reihenfolge der einzelnen Ebenen zu achten. Wie in der Geoinformationsverarbeitung üblich, sollen Punkte oberhalb von Linien und diese wiederum oberhalb von Flächen dargestellt werden.

Der Schritt der Ebenen-Identifizierung kann während des iterativen Einlesens der einzelnen Features durchgeführt werden. So kann bereits an dieser Stelle in Abhängigkeit von Attributwerten eine Selektion der Daten durchgeführt werden.

5.3.3 Koordinatensysteme und Projektionen

Die Breite und Länge sollen in geographischen Koordinaten vorliegen. Die Breite hat folglich einen Wert zwischen $\pm 90^\circ$, die Länge zwischen $\pm 180^\circ$. Liegen die Koordinaten in einer beliebigen anderen Projektion vor, müssen sie transformiert werden. Die Höhenangaben spielen für die Wiedergabe der Vektordaten am sphärischen Display keine wesentliche Rolle, da diese nicht visualisiert werden können (so wie es bei Earthbrowsern möglich ist).

Die Koordinatentransformation kann ebenfalls beim Einlesen mit OGR und Proj4 ("proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs") bzw. EPSG-Codes (4326) durchgeführt werden. Bei OGR wird diese Information mit der Geometrie gespeichert und somit wird jeder Punkt (bzw. Stützpunkt) separat transformiert.

Ursprungsgeometrien liegen in einem geographischen Koordinatensystem vor. Die Objekt- bzw. Visualisierungsgeometrien können je nach Visualisierungsansatz in unterschiedlichen Koordinatensystemen vorliegen.

Bei Ansatz 1 bleiben die Objektgeometrien geographische Koordinaten (Kugelkoordinaten). Die Geometrien werden mit sphärischer Trigonometrie bearbeitet

und so die Visualisierungsgeometrien generiert. Diese liegen anfangs noch als geographische Koordinaten vor, werden dann aber mit einem Vertex Shader in ein zweidimensionales kartesisches Koordinatensystem transformiert. Diese Transformation hängt von der Abbildungsfunktion der Globenabbildung ab.

Bei den Ansätzen 2a und 2b werden die Ursprungsgeometrien in dreidimensionale kartesische Koordinaten umgerechnet. Für diese Transformation muss ein virtueller Erdradius definiert werden. Das Zielkoordinatensystem ist jenes, das von Ogre3D definiert wird (die x und z Achsen spannen die horizontale Ebene auf; die y Achse dient als vertikale Achse). Folgende Formeln können verwendet werden um die Länge und Breite in XYZ umzurechnen:

$$x = r * \cos(\text{lat}) * \cos(\text{lon}) \quad y = r * \sin(\text{lat}) \quad z = -r * \cos(\text{lat}) * \sin(\text{lon})$$

Das negative Vorzeichen bei der Berechnung von z ist notwendig, da die Geometrien gespiegelt dargestellt werden sollen (da sich die Kamera im Inneren der virtuellen Kugel befindet).

Für Ansatz 3 können ebenfalls die oben angeführten Formeln für die Transformation verwendet werden. Auch PovRay verwendet ein dreidimensionales kartesisches Koordinatensystem.

5.3.4 Maßstab/Generalisierung

Das System taktile Hyperglobus ist per se kein System, das mit unterschiedlichen Auflösungen der Daten, so wie es etwa bei Earthbrowsern (Kapitel 4.2) ist, umzugehen hat, da die Größe eines Globenkörpers unveränderlich ist. Die Größe des Globenkörpers kann jedoch zwischen einzelnen Konfigurationen sphärischer Displays variieren (Kapitel 2.2.2) und folglich auch der Maßstab der kartographischen Ausdrucksform (von 1 : 30 Mio. bis 1 : 6 Mio.).

So könnten bspw. die Natural Earth Daten (1 : 110 Mio., 1 : 50 Mio. und 1 : 10 Mio.; Kapitel 3.3.2.1) je nach Größe des sphärischen Displays eingesetzt werden. Auf einem Display mit 2 m Durchmesser könnten die Daten im größten Maßstab eingesetzt werden, bei einer 40 cm Kugel der „mittlere“ Datensatz.

Wird bei der Globendarstellung jedoch keine dermaßen hohe geometrische Genauigkeit der Vektordaten benötigt, können auch kleinere Maßstäbe verwendet werden. Durch den geringeren Detailgrad kann unter Umständen eine graphisch ruhigere Darstellung erzeugt werden. Falls keine Daten in kleineren Maßstäben vorliegen, können an dieser Stelle Generalisierungsalgorithmen zur Vereinfachung, Glättung, Zusammenfassung und Selektion der Geometrien herangezogen werden.

In diesem Zusammenhang und später bei der Datenvisualisierung ist die graphische Mindestgröße der Zeichen zu berücksichtigen. Diese hängt auch beim sphärischen Display einerseits vom menschlichen Sehvermögen, andererseits – und das ist im Fall des sphärischen Displays der limitierende Faktor – von der Hardwarekonfiguration des sphärischen Displays ab.

Im Unterschied zu Earthbrowsern müssen Vektordaten für die Wiedergabe am sphärischen Display, da diese hardwarebedingt einen fixen Maßstab haben, nicht in unterschiedlichen geometrischen Auflösungen (Punkt- und Objektdichte) gerendert werden. Hingegen müssen bei taktilen Hypergloben immer alle Geometrien gerendert werden. Earthbrowser müssen sogar beim kleinsten Maßstab – die gesamte Erdgestalt ist am Monitor sichtbar – nur die der Kamera zugewandten Features rendern.

Im Zuge der Diplomarbeit wird auf Generalisierungsalgorithmen, die für die Wiedergabe von Vektordaten am sphärischen Display nicht zwingend notwendig sind, nicht näher eingegangen. Für den operativen Bereich ergibt der Einsatz solcher Methoden jedoch Sinn, da dadurch mit einer „aufgeräumten“ Geometrie flexibler und schneller umgegangen werden kann. Bspw. setzen sich [FAN-10] mit Methoden der maßstabsabhängigen Vereinfachung globaler Vektordaten auseinander.

5.3.5 Verfeinerung/Segmentierung

Segmentierung bedeutet, dass die Ursprungsgeometrie je nach Darstellungsansatz durch das Hinzufügen von Stützpunkten (Vertices) verfeinert wird und somit zur fertigen Objektgeometrie umgewandelt wird.

Bei Punktdaten entfällt dieser Schritt. Er ist nur bei Linien- und Flächengeometrien notwendig. Sind bei einem Liniensegment (= direkte Verbindung zweier Punkte ohne weitere Stützpunkte) der Start- und Endpunkt zu weit voneinander entfernt, muss ggf. eine regelmäßige Verfeinerung der Geometrie durchgeführt werden. Dabei ist zu beachten, dass neue Stützpunkte entlang der Orthodrome, der kürzesten Verbindung zwischen zwei Punkten auf der Kugeloberfläche, gesetzt werden.

Dieser Schritt steht nicht im Gegensatz zur vorangegangenen Generalisierung. Er dient als letzte Phase der Datenaufbereitung dazu, die Geometrien für die Visualisierung aufzubereiten und ist abhängig von der Visualisierungsmethode.

Bei Ansatz 1 wird die Verfeinerung benötigt um z.B. bei Liniengeometrien die sich mit verändernder Breite verändernde Strichstärke zu simulieren. Bei den Ansätzen 2a und 2b dient die Segmentierung zur Annäherung der Geometrien an die Kugeloberfläche.

Bei Ansatz 3 ist eine Verfeinerung nicht notwendig. In Abbildung 29 wird ein schematischer Vergleich vorgenommen.

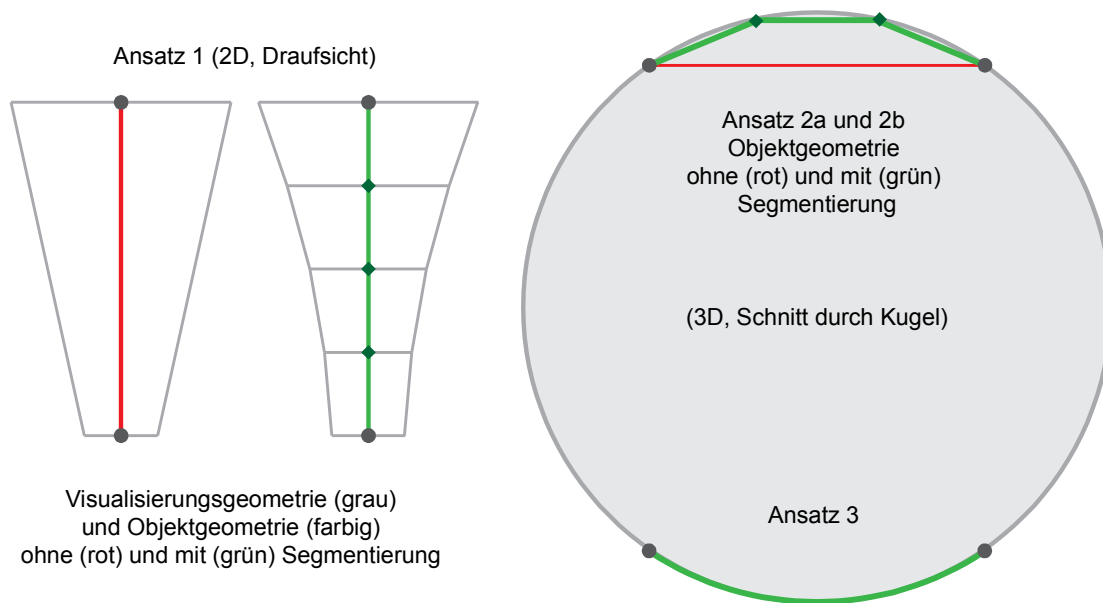


Abbildung 29: Schema der Verfeinerung/Segmentierung von Vektorgeometrien

links: Ansatz 1, Draufsicht im Objektraum 2D; rechts: Ansatz 2a und 2b sowie Ansatz 3, Querschnitt durch die virtuelle Kugel im Objektraum 3D

(Quelle: eigene Darstellung)

Dieser Schritt wird bei Ansatz 1 in geographischen Koordinaten durchgeführt. Bei den Ansätzen 2a und 2b im dreidimensionalen kartesischen Koordinatensystem der Visualisierung.

Es wurden dazu zwei mögliche Ansätze getestet. Bei einer rekursiven Methode wird, solange eine Abbruchsbedingung nicht erreicht ist (gewisser Minimalabstand benachbarter Vertices), der Mittelpunkt zwischen benachbarten Punkten in die Geometrie eingefügt.

Eine weitere Methode kann in dreidimensionalen kartesischen Koordinatensystemen gut umgesetzt werden. Dabei wird der Ortsvektor des Startpunktes um den Normalvektor der Ebene, die durch die Ortsvektoren von Start- und Endpunkt aufgespannt wird, um einen gegebenen Winkel rotiert. In einer iterativen Implementierung wird dieser Winkel jeweils mit dem Iterationsindex multipliziert. Somit werden entlang der Orthodrome in Abhängigkeit dieses Minimalwinkels neue Stützpunkte in die Geometrie integriert.

5.3.6 Stil der Signaturen

Wie bereits besprochen und in diversen Abbildungen angedeutet, können die Stildefinitionen einzelner Features Teil der Vektordaten sein und somit beim Einlesen der Daten ebenfalls importiert werden. Liegen keine Stildefinitionen vor, müssen diese manuell angegeben bzw. zuvor definierte Standardwerte für die Visualisierung verwendet werden.

Ein Stil kann auch in Abhängigkeit von Attributen generiert werden. Bspw. könnte die Größe von Punktsignaturen für Städte anhand der Einwohnerzahl berechnet werden. Ein weiteres Beispiel, bei dem die Signaturen (in diesem Fall die Farbe bzw. der Tonwert) anhand von Attributen ermittelt werden, ist die Erstellung einer Choroplethenkarte. Im Zuge der Umsetzungen wird auf die attributabhängige Visualisierung nicht näher eingegangen.

Wie anfangs bereits angedeutet, werden bei den untersuchten Ansätzen die unterschiedlichsten graphischen Variations- und Kombinationsmöglichkeiten von Signaturen nicht ausgeschöpft (Kapitel 3.2). Für Punktsignaturen werden Kreise, Quadrate und gleichseitige Dreiecke visualisiert, die in der Größe, der Form, der Füllung, der Richtung, dem Tonwert und der Farbe veränderbar sind, wobei zusätzlich oder ausschließlich eine Außenlinie gezeichnet werden kann. Linien werden als durchgezogene Geometrien visualisiert, die in der Strichbreite, der Farbe sowie dem Tonwert variieren können. Gepunktete oder unterbrochene Linien werden nicht behandelt. Bei flächenhaften Signaturen wird lediglich die Variation der Farbe und des Tonwertes berücksichtigt. Polygonkanten werden durch Liniensignaturen umgesetzt.

5.3.7 Thematische und inhaltliche Anforderungen

Neben den technischen Anforderungen an die Geometrie für die Wiedergabe globaler Vektordaten am sphärischen Display können auch thematische und inhaltliche Voraussetzungen an die Daten gestellt werden.

Einerseits können Vektordaten als topographische Grundlagen dienen und Objekte beinhalten, die die physische Welt beschreiben und vor allem für die Erzeugung von Grundkarten verwendet werden. Auf der anderen Seite können Vektordaten Träger thematischer Geoinformationen sein. Dabei handelt es sich vorwiegend um nicht topographische, jedoch auf der Erde lokalisierbare, reale oder imaginäre Sachverhalte [IMH-72], die für die Wiedergabe am taktilen Hyperglobus auf die Parameter der Globenwürdigkeit geprüft werden können. Diese Unterscheidung von Topographie und Thematik spiegelt sich u.a. in dem Natural Earth Datensatz, der zwischen „physical“ und „cultural“ unterscheidet, wider.

Die Parameter der Globenwürdigkeit werden ausführlich bei [RIE-00] diskutiert. Damit die Globenwürdigkeit der darzustellenden Thematik gegeben ist, muss „[...] eine enge Bindung an die Gestalt des Himmelskörpers bestehen [...]“, müssen die Daten auch vollständig (für die gesamte (globale) Ausdehnung des Phänomens) verfügbar sein und muss deren Aussage auch bei entsprechend kleinem Maßstab bestehen bleiben. In weiterer Folge kann ein Vektordatensatz auch durch die ergänzende Kombination mit anderen Sachverhalten der Globenwürdigkeit gerecht werden.

Die Visualisierung innereuropäischer Flugverbindungen am taktilen Hyperglobus würde der Globenwürdigkeit nicht gerecht werden. Aus der Wiedergabe interkontinentaler Flüge am sphärischen Display kann hingegen trotz des Maßstabs und aufgrund der Globalität ein Informationsgewinn für die Betrachtenden generiert werden.

Weiters ist für Vektordaten vor allem der Punkt Kombinationsfähigkeit bzw. Kombinationspflicht sehr wichtig. Oft dienen Vektordaten als Overlays, die zusätzliche Information für eine globale Thematik bieten. Bezugnehmend auf das in Kapitel 4.1.1.1 besprochene Beispiel der Kontinentaldrift, würden die Punktsignaturen und Labels alleine keinen Sinn ergeben. Erst durch die Kombination mit der Kontinentalverschiebung selbst erhalten die Signaturen ihre (sehr wichtige) Bedeutung.

Aus Sicht des Autors werden gegenwärtig, bei der Aufbereitung von Themen für den taktilen Hyperglobus, nicht immer die Parameter der Globenwürdigkeit berücksichtigt. Teilweise könnten gewisse Themen und Animationen, die weniger eine globale, als viel mehr eine kontinentale oder regionale Relevanz aufweisen, eben so gut verebnet auf einem Monitor präsentiert werden. Der Vorteil am taktilen Hyperglobus ist, neben der Einhaltung aller Trageeigenschaften, dass durch eine globale Visualisierung auf neue, zuvor nicht berücksichtigte Zusammenhänge geschlossen werden kann.

5.4 Die Visualisierung von Vektordaten am sphärische Display

5.4.1 Wesentliche Unterschiede von 2D und 3D Ansätzen

Je nach eingesetzter Methode können wesentliche Unterschiede, die den verwendeten Objektraum (2D oder 3D) betreffen, festgestellt werden. Im Folgenden werden der Zeichenträger und die damit verbundenen Verzerrungen und Nähte sowie die grundlegenden mathematischen Konzepte unterschieden.

5.4.1.1 Zeichenträger

Einen Unterschied stellt der Zeichenträger dar. Im Objektraum 3D ist eine virtuelle Kugeloberfläche der Zeichenträger. Die Geometrien werden direkt bzw. stark angenähert auf dem Referenzkörper dargestellt. Im Objektraum 2D handelt es sich

dabei um einen verebneten Raum, der eine bestimmte kartographische Projektion verwendet. Die Vektorgeometrien werden dabei direkt in bspw. die Platkarte oder in die Globenabbildung selbst eingezeichnet.

5.4.1.2 Verzerrungen und Nähte

Dadurch ergibt sich ein weiteres Unterscheidungsmerkmal, das bei der Umsetzung der jeweiligen Methoden ein entscheidender Faktor ist. Jede kartographische Projektion hat gewisse Verzerrungen, die bei der Visualisierung im zweidimensionalen Raum berücksichtigt werden müssen. Diese Verzerrungen müssen bei der Wiedergabe der Daten im dreidimensionalen Raum nicht berücksichtigt werden. Die Verzerrungen werden hier automatisch durch die Globensoftware in der Ausgabeabbildung generiert (bei Ansatz 2a und 2b durch das Spiegel/Kamera System von OmniSuite; bei Ansatz 3 durch eine sphärische Kamera). Auch die oft angesprochene „Ost-West-Naht“ spielt bei korrekter Umsetzung im Objektraum 3D keine wesentliche Rolle, muss aber bei 2D Ansätzen mit großer Sorgfalt behandelt werden.

5.4.1.3 Mathematische Grundlagen

Ein weiterer Punkt ist der Einsatz unterschiedlicher mathematischer Grundlagen. Bei Ansatz 1 wird sphärische Trigonometrie verwendet um Polygonnetze für Punkt-, Linien- und Flächensignaturen zu generieren. Dazu wird ein Skript verwendet, das folgende Operationen ausführen kann [VEN-12]:

- Berechnung der Orthodrome (Distanz, Azimut am Start- und Endpunkt, Mittelpunkt)
- Berechnung eines Zielpunktes sowie dessen Azimut anhand eines Startpunktes, eines Kurswinkels und einer Distanz (entlang der Orthodrome)
- Abstandsberechnung (entlang der Orthodrome) zwischen einem gegebenen Punkt und einem gegebenen Großkreis
- Berechnung des polnahsten Punktes auf einer Orthodrome
- Berechnungen von Punkt 1 und 2 für die Loxodrome

Der Azimut oder Kurswinkel ist jener Winkel, der sich zwischen der Nordrichtung und einer Zielrichtung im Uhrzeigersinn aufspannt.

Bei den 3D Ansätzen 2a, 2b und 3 spielt die analytische Geometrie, im Speziellen die Vektorrechnung, eine wichtige Rolle. Dazu bieten die Ogre3D Klassen Math und Vector3 (für die Ansätze 2a und 2b) sowie PovRay (für Ansatz 3) die notwendigen Funktionen und Methoden an [O3D-12], [POV-12]:

- Addition und Subtraktion
- Produkt mit einem Skalar
- Skalarprodukt
- Kreuzprodukt
- Betrag eines Vektors
- Einheitsvektor/Normierung

Darüber hinaus werden zur Generierung der Visualisierungsgeometrien auch Rotationen eingesetzt. Einerseits sind es Rotationen von Ortsvektoren um den Koordinatenursprung, andererseits Rotationen von Normalvektoren um einen beliebigen Ortsvektor.

Wie diese mathematischen Grundlagen eingesetzt werden, wird in Kapitel 6 erörtert. Ein weiteres mathematisches Konzept, die konstruktive Festkörpergeometrie, wird im folgenden Kapitel besprochen.

5.4.2 Visualisierungsgeometrie durch Constructive Solid Geometry

Bei Ansatz 3 wird zur Wiedergabe der Vektorgeometrien in PovRay konstruktive Festkörpergeometrie verwendet. Gebräuchlicher ist die englische Bezeichnung Constructive Solid Geometry (CSG).

CSG-fähige Körper werden durch Mengenoperationen, die aus der bool'schen Algebra bekannt sind, so aufbereitet, dass dadurch die Vektorgeometrien repräsentiert werden können. Zu CSG-fähigen Körpern zählen bspw. der Quader (*box*), die Kugel (*sphere*), der Zylinder (*cylinder*), der Kegel bzw. der Kegelsumpf (*cone*), das Prisma (*prism*) oder der Torus [LOH-12]. Grundsätzlich können die unten angeführten bool'schen Operationen auf alle Körper angewendet werden, bei denen eine Innen- bzw. Außenseite identifiziert werden kann. Da der Normalvektor einer Ebene (*plane*) in PovRay nach Außen zeigt, gehört auch diese zu den CSG-fähigen Körpern.

Folgende Operationen sind von PovRay implementiert und werden im Zuge der Diplomarbeit eingesetzt (Abbildung 30) [O3D-12], [LOH-12]:

- Vereinigung (*union, merge*)
- Differenz (*difference*)
- Schnittmenge (*intersection*)
- Komplementmenge (*inverse*)

- Öffnender Schnitt (*clipped_by*)

Diese Operationen müssen verschachtelt auf die oben angeführten Körper angewandt werden um die Darstellungsgeometrie der Vektordaten zu erhalten.

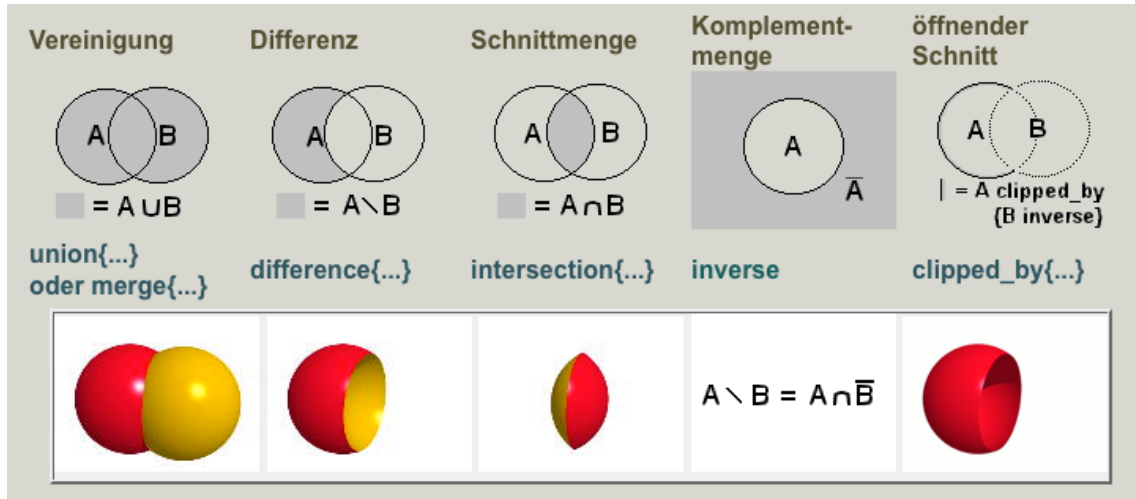


Abbildung 30: Mengenoperationen für CSG in PovRay

Von links nach rechts: Vereinigung, Differenz und Schnittmenge zweier Körper sowie die inverse Menge und ein öffnender Schnitt.

(Quelle: [LOH-12], unverändert)

Bei der Vereinigung kann zwischen *union* und *merge* unterschieden werden. Bei beiden Methoden werden zwei oder mehrere Objekte, die nicht notwendigerweise räumlich zusammenhängen, verbunden. So gebildete Objektgruppen können gemeinsam eingefärbt, skaliert, rotiert oder verschoben werden. Darüber hinaus können mit Hilfe der Vereinigung neue Objekte generiert werden, die sich aus den grundlegenden geometrischen Körpern zusammensetzen. Bei *union* bleiben die inneren Trennungsflächen der verbundenen Objekte bestehen, bei *merge* werden diese verschmolzen. [LOH-12]

Bei der Differenz wird der Schnittbereich zweier Körper aus der Menge des erst genannten Objektes entfernt, wobei die dadurch entstandene Schnittfläche dem zweiten Objekt entspricht. So können „[...]“ aus vorhandenen Körpern Löcher, Ausbuchtungen, Abschrägungen, Einkerbungen und dergleichen [...] [LOH-12] herausgefräst werden.

Die Schnittmenge, die Bereiche zweier Körper, die sich überlappen, wird mit dem *intersection* Operator berechnet. Bei Differenzen und Schnittmengen kann mit dem Zusatz *inverse* die Komplementmenge eines Objektes für die jeweilige Operation verwendet werden. So lassen sich verschachtelte Operationen vereinfachen.

Schließlich bietet die *clipped_by* Modifikation die Möglichkeit öffnende Schnitte zu generieren. Dabei wird ein ursprünglicher Körper durch einen zweiten aufgeschnitten. Über bleibt die Hülle des geöffneten Körpers (Abbildung 30).

Im Zuge der unterschiedlichen CSG-Operationen können Eindeutigkeitsprobleme und somit ungewollte Artefakte im Ergebnis entstehen. Diese sog. komplanaren Flächen ergeben sich aus ungenauen und nicht eindeutigen Angaben bzw. durch unvermeidliche Rundungsfehler des Computers.

So muss bspw. bei der Bildung der Differenz darauf geachtet werden, dass keine Flächen des ersten Körpers mit Flächen des zweiten (ausschneidenden) Körpers ident sind, sondern darüber hinausragen. Andernfalls kann der Raytracer keine eindeutige Entscheidung treffen, welche der beiden Flächen über bleiben soll [LOH-12].

Es gilt also komplanare Flächen durch eindeutige Angaben zu vermeiden. Dabei ist auch zu beachten, dass einzelne, sich überlagernde Signaturen nicht in exakt der selben Ebene schneiden, sondern einen geringen Abstand zueinander aufweisen.

5.4.3 Visualisierungsgeometrie durch Meshgenerierung

Grafikkarten bzw. Grafikprozessoren verarbeiten Polygonnetze bzw. Dreiecksnetze (Meshes). Dazu muss eine Liste von Punkten (Vertexpuffer) in den Grafikspeicher geladen werden. Damit die Grafikkarte weiß, wie sie die Punkte verarbeiten soll, muss darüber hinaus definiert werden, welches Zeichenprimitiv gerendert werden soll.

Die Zeichenprimitive in Abbildung 31 sind im Zusammenhang mit der Diplomarbeit bei Ansatz 1 und 2a entscheidend. Bei Ansatz 2b wird durch die extern definierte Polygonstruktur die Darstellungsgeometrie bestimmt.

Eine `LINE_LIST` ist eine Liste mehrerer Linien, wobei pro Linie zwei Punkte benötigt werden. Ein `LINE_STRIP` ist eine Folge von verbundenen Linien. Dabei wird ein Startpunkt benötigt und ein weiterer Punkt je Linie. Die `TRIANGLE_LIST` ist eine Liste mehrerer Dreiecke, wobei drei Punkte pro Objekt benötigt werden. Der `TRIANGLE_STRIP` ist ein Streifen zusammenhängender Dreiecke. Für das erste Dreieck werden drei Punkte benötigt, für jedes weitere Dreieck ein zusätzlicher Punkt. Aus dem zusätzlichen Punkt und den zwei vorausgegangenen Punkten wird das weitere Dreieck konstruiert. Ein `TRIANGLE_FAN` ist eine Art Fächer, bei dem für das Startdreieck ebenfalls drei Punkte und für jedes weitere Dreieck ein Punkt benötigt werden. Hierbei wird das neue Dreieck immer mit dem ersten Punkt in der Liste (Zentrum des Fächers), dem neuen Punkt und dem vorausgegangenen Punkt konstruiert.

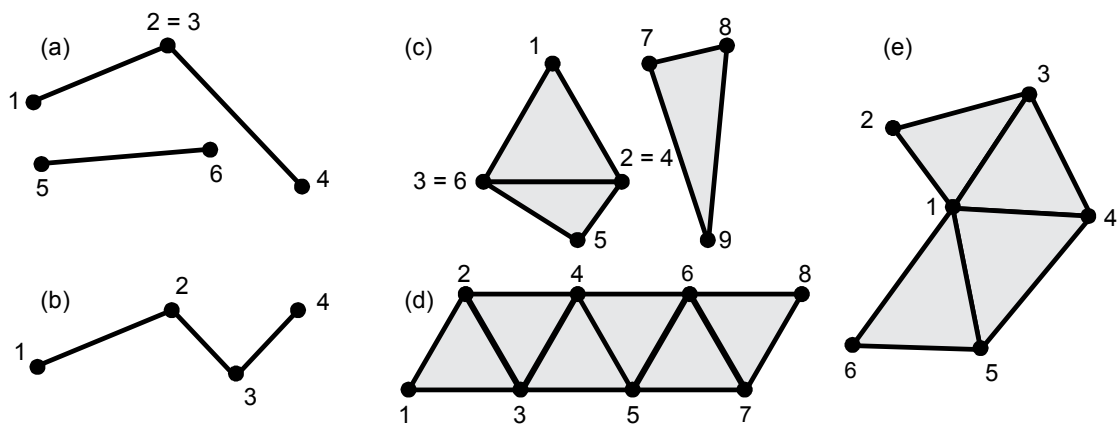


Abbildung 31: Zeichenprimitive für die Wiedergabe durch den Grafikprozessor

(a): *LINE_LIST*; (b): *LINE_STRIP*; (c): *TRIANGLE_LIST*; (d): *TRIANGLE_STRIP*;
(e): *TRIANGLE_FAN*

(Quelle: eigene Darstellung)

Bei der Erstellung der Dreiecke ist auf die Drehrichtung (*windingorder*; im oder gegen den Uhrzeigersinn), die die Vorder- bzw. Rückseite eines Dreiecks definiert, zu achten. Andernfalls kann es dazu kommen, dass bspw. nur die Vorderseite der Dreiecke angezeigt wird, wobei diese dann auch der Kamera abgewandt sein kann. Um dies zu umgehen kann mit dem sog. Culling festgelegt werden, welche Seite bzw. ob beide Seiten gerendert werden sollen.

Auch hier besteht das Problem komplanarer Polygone. Das sog. Z-Fighting beruht ebenfalls auf Präzisionsproblemen bei der Rundung. Z-Fighting kann durch die geschickte Verwendung des Z-Puffers vermindert werden. Ein Z-Puffer wird verwendet, um bei der Rasterung (als Teil der Grafikpipeline) zu berechnen, welche Objektteile gerendert werden sollen. Darüber hinaus ergibt es Sinn bereits bei der Erzeugung der Polygonnetze darauf zu achten, dass keine komplanaren Bereiche entstehen.

An dieser Stelle wird nochmals darauf hingewiesen, dass das bildgebende Konzept bei Raytracern nicht mit den Prozessen der Grafikpipeline verglichen werden kann.

6 Technische Implementierungen

In diesem Kapitel werden die technischen Aspekte der unterschiedlichen Ansätze besprochen. Ganz bewusst wird hier auf die Eingliederung von Quelltext (JavaScript, C++, PovRay SDL) verzichtet. Jedoch werden die dahinterstehenden Konzepte und Algorithmen zur Generierung der Visualisierungsgeometrie für Punkt- Linien- und Flächengeometrien besprochen.

6.1 Punktgeometrien

Punktgeometrien können durch einfache Geometrien, wie Kreis, Ellipse, Quadrat, Rechteck, Rhombus, regelmäßiges Vieleck udgl., repräsentiert werden. Wie diese Signaturen in den einzelnen, oben besprochenen, Ansätzen erzeugt werden können, ist in den folgenden Abschnitten beschrieben.

6.1.1 OpenGL ES / Sphärische Trigonometrie

Punktgeometrien können bei dieser Methode mit den oben erwähnten Methoden der sphärischen Trigonometrie als Dreiecksnetze in deren Darstellungsgeometrie überführt werden. Dazu wird vor allem die Berechnung eines Zielpunktes anhand eines Startpunktes (Zentralpunkt), eines Kurswinkels und einer Distanz (entlang der Orthodrome) benötigt.

Ausgehend von den Koordinaten des Punktes werden die benötigten Neupunkte berechnet. Die folgenden Schritte erläutern den dazu benötigten Algorithmus. Zu Beginn müssen mehrere Werte bestimmt werden: die Granularität, die Symbolbreite, der Mittelpunkts- oder Zentriwinkel und die Symbolrotation.

Bei der Granularität handelt es sich um die Anzahl der Ecken der zu erstellenden geometrischen Signatur. Ein ganzzahliger Wert gibt die Form (graphische Variation) an: 3 = Dreieck, 4 = Quadrat, 5 = Pentagon, 6 = Hexagon, usw. Ab einer Granularität von rund 20 (abhängig von der Symbolgröße) nähert sich die Form an die eines Kreises an.

Die Symbolbreite gibt die Größe (graphische Variation) bzw. den Durchmesser des Symbols in Kilometer an. Für die Berechnungen ist hierbei zu beachten, dass der Erdradius mit 6731 km angenommen wird.

Mittelpunkts- oder Zentriwinkel wird zur iterativen Konstruktion der jeweiligen Kurswinkel verwendet. Der Wert lässt sich mit der Division von 360° durch die Granularität bestimmen.

Die Symbolrotation ist eine Rotation des Symbols um dessen Zentrum (Zentralpunkt). Dadurch kann die graphische Variable der Richtung (Orientierung) beeinflusst werden.

Schließlich werden mit einer Schleife [Null; Granularität] die Neupunkte iterativ ermittelt. Der Schleifenzähler wird je Runde um eins inkrementiert. Bei einer Granularität von 4 werden die Anweisungen innerhalb der Schleife folglich 5 Mal ausgeführt (der Schleifenzähler nimmt die Werte 0 bis 4 an).

Je Iterationsdurchgang wird ein Kurswinkel berechnet. Dabei wird der Schleifenzähler mit dem Zentriwinkel multipliziert und mit der Symbolrotation addiert (Schleifenzähler * Zentriwinkel + Symbolrotation). Ein Kurswinkel von Null zeigt in Richtung Norden.

Der jeweilige Neupunkt wird mittels Startpunkt (Zentralpunkt), der halbierten Symbolbreite und dem jeweiligen Kurswinkel berechnet. Die benötigten Formeln sind bei [VEN-12] zu finden.

Abschließend ist es eventuell (je nach Symbolgröße und Granularität) notwendig zwischen benachbarten Neupunkten (entlang der Orthodrome) weitere Stützpunkte einzufügen. So können die Verzerrungseigenschaften der Globenabbildung besser berücksichtigt werden (Abbildung 32).

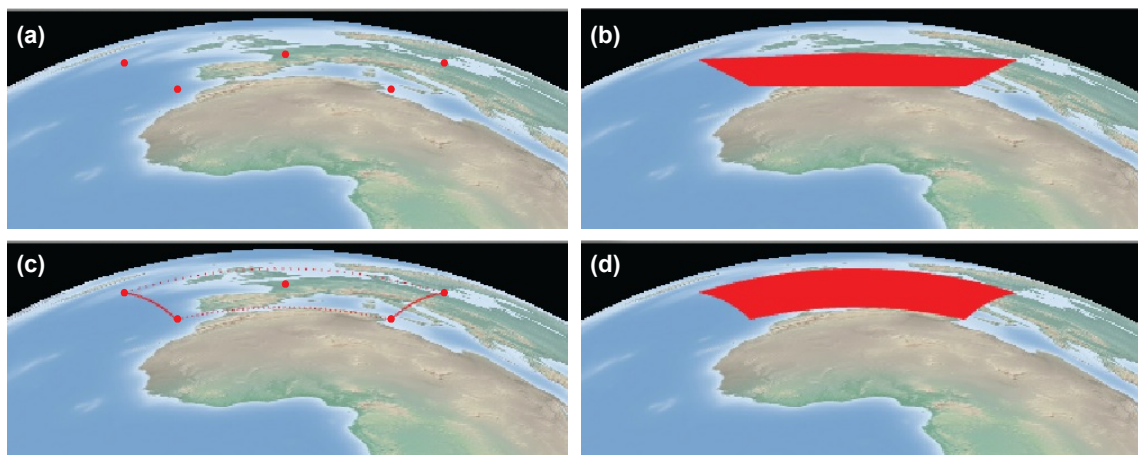


Abbildung 32: Große quadratische Signatur am Außenrand der Globenabbildung

Granularität = 4; Symbolbreite = 3000 km; Zentriwinkel = 90°; Symbolrotation = 45°; Kurswinkel 1 = 45°; Kurswinkel 2 = 135°; Kurswinkel 3 = 225°; Kurswinkel 4 = 315°; Kurswinkel 5 = 405°;

(a) Zentrum und Neupunkte; (b) TRIANGLE_FAN mit Zentrum und Neupunkten; Ergebnis ist nicht korrekt, da die Verzerrung der Globenabbildung nicht exakt berücksichtigt wird; (c) durch Interpolation weiterer Stützpunkte wird ein korrektes Ergebnis erzielt (d);

(Quelle: eigene Darstellung, topographische Grundlage: [NED-12])

Die anhand des oben besprochenen Algorithmus generierten Punkte werden nun als TRIANGLE_FAN gerendert, der ursprüngliche Punkt dient als Zentrum des Fächers.

Jeder Vertex wird von geographischen Koordinaten in das Koordinatensystem der Globenabbildung transformiert (Vertex Shader). Ein zweidimensionales Modell ist das Ergebnis des Algorithmus.

6.1.2 OGRE3D / Vektorrechnung

Grundlegend funktioniert die Berechnung der Neupunkte für die Darstellungsgeometrie ähnlich wie bei dem oben beschriebenen Algorithmus.

Die Granularität, die Symbolbreite und der Zentriwinkel sind analog zu dem vorhergegangenen Ansatz zu verstehen. Bei der Symbolrotation ist zu berücksichtigen, dass ein Kurswinkel von Null gegen Osten zeigt.

Bevor die Neupunkte iterativ generiert werden, wird ein Konstruktionspunkt generiert. Dazu wird der Normalvektor jener Ebene gebildet, die durch die positive y-Achse (a) (Nordpol) und den Ortsvektor des Zentrums (b) (ursprüngliche Objektgeometrie) aufgespannt wird. Der Normalvektor (n) wird durch das Kreuzprodukt von (a) und (b) gebildet. Anschließend wird dieser Vektor (n) normiert. Der dadurch entstandene Einheitsvektor wird mit der halbierten Symbolbreite multipliziert (v). Nun kann der Konstruktionspunkt, indem der Vektor (v) an den Zentralpunkt (b) durch Addition angehängt wird, berechnet werden. Abschließend muss der Konstruktionspunkt auf die Kugeloberfläche projiziert (verschoben) werden. Das wird durch die Normierung des Ortsvektors des Konstruktionspunktes mit anschließender Multiplikation des Erdradius erreicht.

Die Iteration wird ebenfalls über die oben angeführten Werte durchgeführt. Das Vorgehen innerhalb der Schleife beruht jedoch nicht auf sphärischer Trigonometrie, sondern auf dreidimensionaler Vektorgeometrie, wobei das Prinzip ähnlich ist.

Es wird nicht wie zuvor jeder Neupunkt ausgehend vom Zentrum und dem Kurswinkel berechnet. Die Neupunkte werden bei diesem Ansatz durch Rotation des Konstruktionspunktes um die Achse, die durch den Ortsvektor des Zentralpunktes gegeben ist, erzeugt. Der Kurswinkel bzw. der Rotationswinkel ergibt sich jeweils wieder durch den Schleifenzähler, den Zentriwinkel und die Symbolrotation. Informationen zur Rotation um eine beliebige Achse im dreidimensionalen Raum sind u.a. bei [MUR-11] zu finden. Der Konstruktionspunkt fließt nicht in die Visualisierungsgeometrie, die wiederum als TRIANGLE_FAN gerendert wird, mit ein. Ein Beispiel einer punkthaften Signatur in Ogre 3D zeigt Abbildung 33.

Darüber hinaus kann, wenn der Zentrumspunkt nicht berücksichtigt wird, mit den Neupunkten sehr einfach eine Außenlinie (rote Linie in Abbildung 33) mit der

Strichstärke der Einheit eins erzeugt werden. Hierzu werden die Neupunkte als LINE_STRIP gerendert.

Überschreiten die Abstände zwischen den Neupunkten aufgrund einer zu großen Symbolgröße einen gewissen Schwellenwert, ergibt es auch hier Sinn zusätzliche Stützpunkte zwischen den Neupunkten (entlang der roten Linie in Abbildung 33) einzufügen.

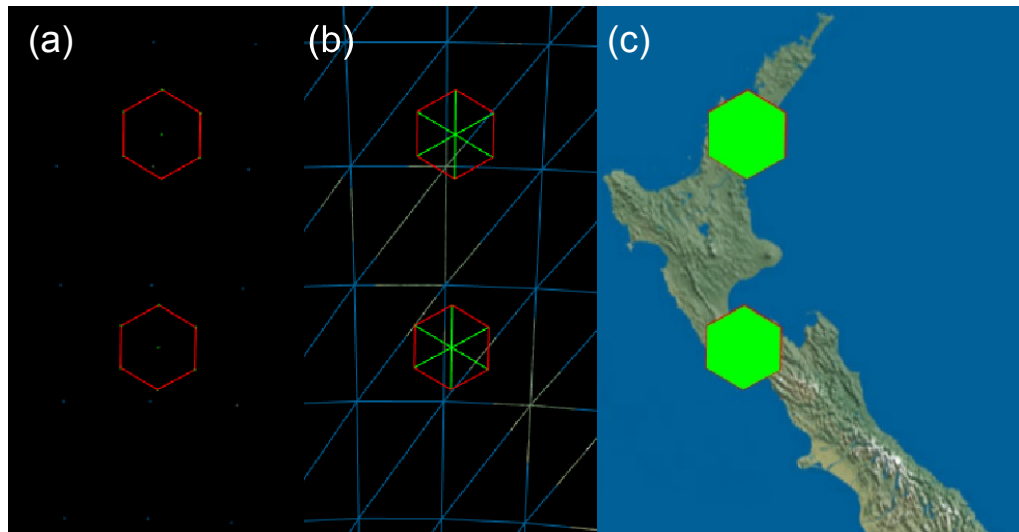


Abbildung 33: Hexagon als Signatur in Ogre3D

(a) Zentrums- und Neupunkte in grün, Außenlinie in rot; (b) Drahtgitterdarstellung der Szene zeigt Dreiecksstruktur; (c) Hexagon als Punktsignatur (Granularität = 6; Symbolbreite = 200 km; Zentriwinkel = 60°; Symbolrotation = 30°)

Anmerkung: Neuseeland wird hier gespiegelt dargestellt, weil das Kamerasystem im Inneren der virtuellen Kugel platziert ist.

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

Eine Alternative um Punktsignaturen in OmniSuite (Ogre3D) zu visualisieren wäre die Verwendung von Billboards (Kapitel 4.2.1). Solange jedoch OmniSuite mit der in Kapitel 2.2.3.5 vorgestellten Methode (Environment Mapping mit CubeMap) das auf das sphärische Display zu projizierende Bild erstellt, können klassische Billboards in OmniSuite nicht eingesetzt werden. Billboards, die im Bereich einer Würfelkante bzw. einer Ecke liegen, werden, da sie sich jeweils in Richtung der zwei bzw. drei Würfelseiten ausrichten, nicht korrekt wiedergegeben (die Verzerrungen der CubeMap werden nicht korrekt dargestellt).

6.1.3 PovRay / CSG

Dieser Ansatz unterscheidet sich sehr stark von den zwei zuvor erläuterten Methoden. Bei diesem Ansatz werden die in Kapitel 5.4.2 vorgestellten Verfahren der Constructive Solid Geometry verwendet, um die Visualisierungsgeometrie zu generieren.

Anhand des nachfolgenden Beispiels, der Visualisierung einer geometrischen Signatur (Raute als Form), wird das Konzept dazu erläutert.

Für eine Punktgeometrie (Latitude = 60° N, Longitude = 90° O) soll eine Raute als Darstellungsgeometrie generiert werden (Abbildung 34).

Der Grundkörper gibt die Form der späteren Signatur wieder. In diesem Fall wird ein Würfel mit der Seitenlänge 1 gebildet und anschließend entlang der positiven x-Achse um einen Wert skaliert, der etwas größer als der Erdradius ist (Abbildung 34 (a)).

Anschließend wird mit einer Skalierung des Quaders entlang der y- und z-Achse die Symbolgröße bestimmt (Abbildung 34 (b)). Sind die Skalierungen entlang der zwei Achsen unterschiedlich, kann auch auf die Form Einfluss genommen werden (Rechteck, Ellipse).

Die Symbolrotation wird um die x-Achse ausgeführt. In diesem Fall beträgt diese 45° (Abbildung 34 (c)). So kann die Orientierung einer Signatur variiert werden.

In den folgenden zwei Schritten wird die Position der Signatur bestimmt. Dazu wird der Quader um die z-Achse um einen Wert von 60° rotiert (Abbildung 34 (d)). Somit ist die geographische Breite bestimmt. Zur Bestimmung der geographischen Länge wird das Objekt um die y-Achse um einen Wert von 90° rotiert.

Schließlich muss der Schnittbereich zwischen dem positionierten Grundkörper und der virtuellen Erdkugel ermittelt werden.

Für den Grundkörper kann an Stelle des Quaders mit quadratischer Grundfläche eine vierseitige (quadratische) regelmäßige Pyramide mit der Spitze im Koordinatenursprung verwendet werden. Die minimalen Unterschiede sind nur bei größeren Signaturen sichtbar. Werden Pyramiden verwendet, so sind die Kanten der Schnittflächen entlang der Orthodrome, andernfalls verlaufen diese entlang der Loxodrome. Das trifft auch bei den kreis- und dreieckförmigen Punktsignaturen zu.

Für kreisförmige Punktsignaturen kann eine entsprechend große Kugel direkt an den entsprechenden Koordinaten (Kugelmittelpunkt) platziert werden und anschließend die Schnittfläche mit der virtuellen Kugeloberfläche gebildet werden. Ebenso kann ein Zylinder (so wie im Beispiel beschrieben) oder ein Kegel (mit der Spitze im Koordinatenursprung) verwendet werden.

Für gleichseitige Dreiecke kann einerseits ein Prisma, mit einem gleichseitigen Dreieck als Grundfläche, andererseits eine dreiseitige regelmäßige Pyramide (Tetraeder) mit Spitze im Koordinatenursprung verwendet werden.

Das Prisma kann ebenfalls für beliebige regelmäßige Vielecke (Pentagon, Hexagon, usw.) als Grundkörper verwendet werden.

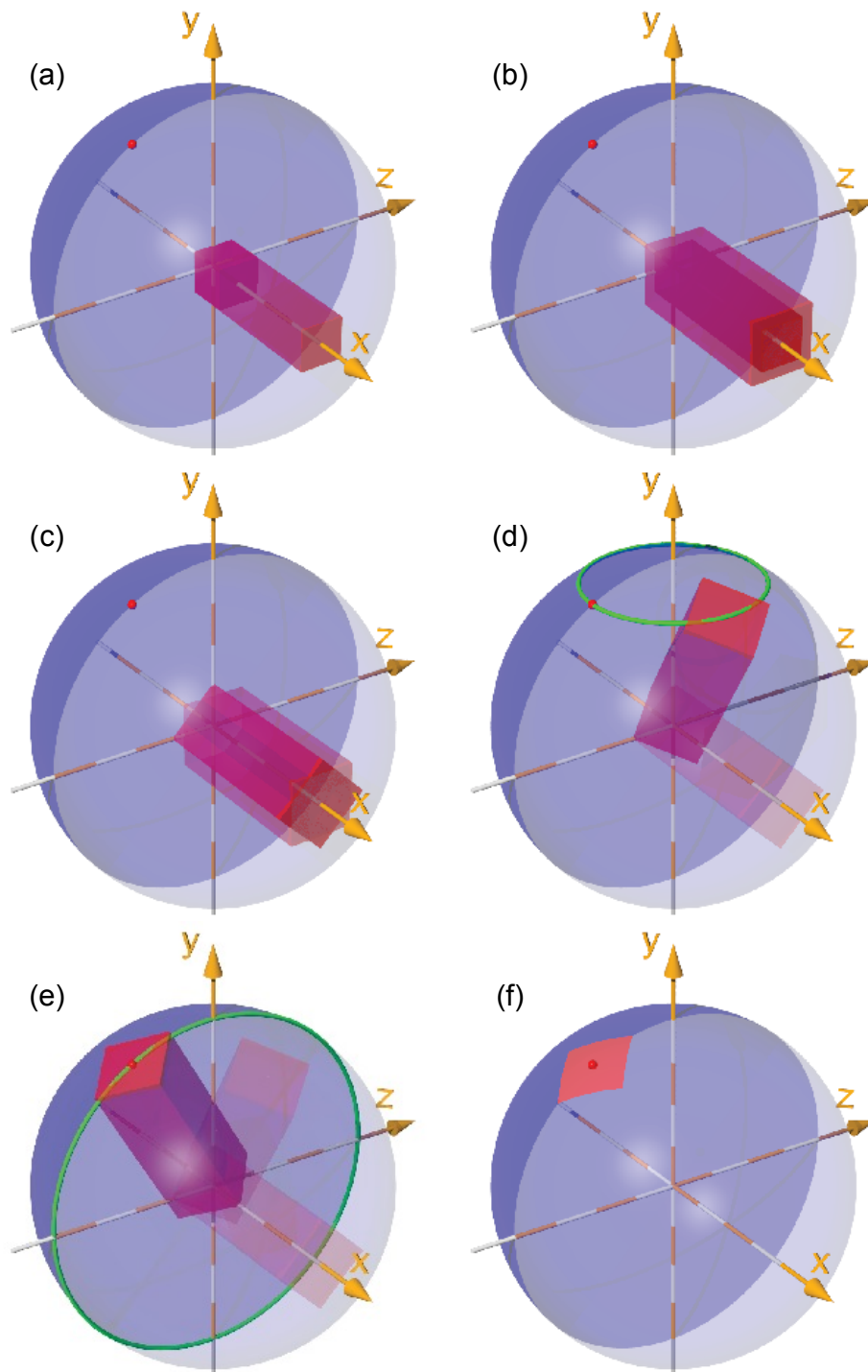


Abbildung 34: Generierung einer Darstellungsgeometrie in PovRay mit CSG

Schema zur Erstellung einer punkthaften Signatur in PovRay unter der Verwendung von CSG; (a) Grundkörper; (b) Symbolgröße; (c) Symbolrotation; (d) geographische Breite; (e) geographische Länge; (f) Schnittbereich von positioniertem Grundkörper und virtueller Erdkugel = Raute als Signatur

(Quelle: eigene Darstellung)

Abbildung 35 zeigt die möglichen graphischen Variationen punktförmiger Signaturen, die mit Hilfe von PovRay im Zug dieser Diplomarbeit implementiert wurden: die Größe, die Form, die Füllung, die Orientierung, der Tonwert und die Farbe. Diese lassen sich auch miteinander kombinieren.

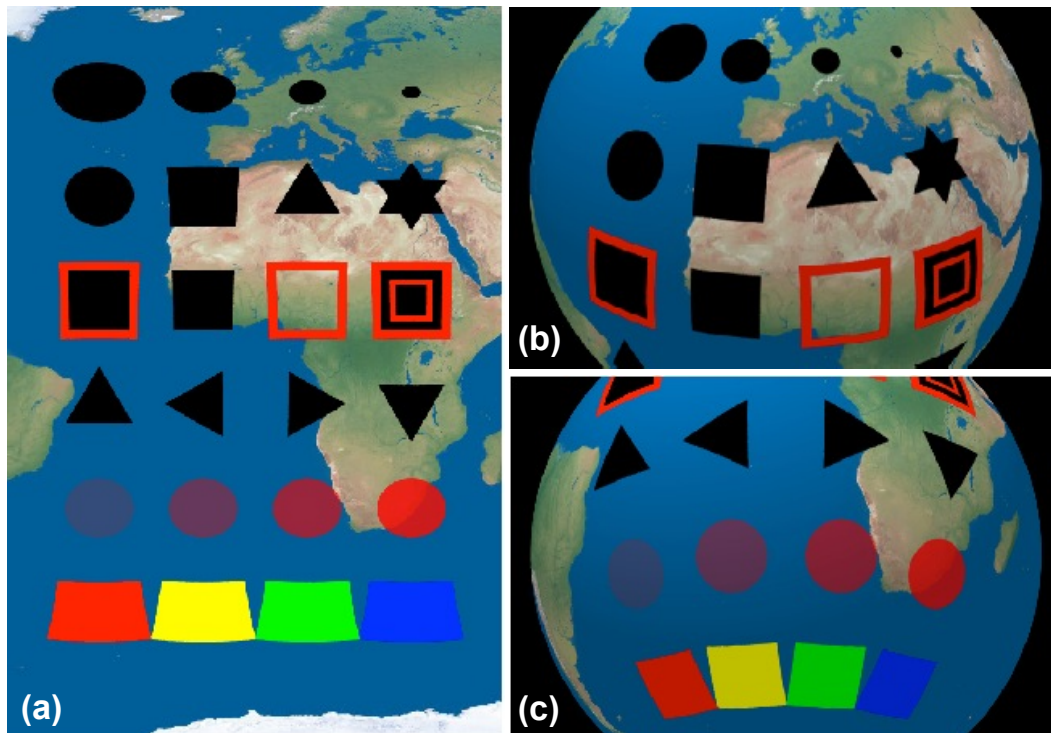


Abbildung 35: Grafische Variationsmöglichkeiten von Punktsignaturen (PovRay)

Grafische Variablen (von oben nach unten): Größe, Form, Füllung, Richtung, Tonwert, Farbe; (a) Darstellung auf der Platkarte; (b) und (c) Darstellung auf dem Globus.

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

6.2 Liniengeometrien

Liniengeometrien können ebenfalls, je nach Ansatz, unterschiedlich visualisiert werden. Im Zuge der Diplomarbeit wird nur die Visualisierung durchgezogener, nicht unterbrochener Liniensignaturen behandelt.

6.2.1 OpenGL ES / Sphärische Trigonometrie

Nachdem die Koordinaten eines LineStrings eingelesen wurden, wird kontrolliert, ob eine Segmentierung/Verfeinerung notwendig ist. Überschreitet ein Liniensegment eine festgelegte Distanz, werden neue Stützpunkte entlang der Orthodrome eingefügt. Dieser iterative Vorgang verwendet wiederum die in Kapitel 5.4.1.3 besprochenen Konzepte der sphärischen Trigonometrie.

Im nächsten Schritt werden weitere Punkte generiert, die dazu dienen, die äußere Begrenzung der Liniensignatur festzulegen. Dafür wird durch die Liste aller Punkte

iteriert und für jede Koordinate zwei neue Punkte ermittelt. Jeder Punkt, ausgenommen der Start- und der Endpunkt eines LineStrings, ist Teil (Verknüpfungspunkt) zweier benachbarter Liniensegmente. Mit den um $+90^\circ$ bzw. -90° rotierten Kurswinkeln beider Segmente und mit der halbierten Strichstärke werden zwei neue Zielpunkte berechnet. Bei Start- und Endpunkt wird nur das eine Segment in die Berechnung miteinbezogen. Die ursprünglichen Punkte, die im Zentrum der Liniensignatur verlaufen, werden nicht für die Generierung der Darstellungsgeometrie miteinbezogen.

Aus den Punktpaaren, die entlang der linken bzw. rechten Außengrenze der Liniensignatur verlaufen, wird in einem weiteren Schritt eine Dreiecksvernetzung, ein TRIANGLE_STRIP, gebildet.

Diese Vertex Daten werden auf die Grafikkarte geladen und dort weiterverarbeitet, umprojiziert (Vertex Shader) und für den Globus ausgegeben (Abbildung 36).

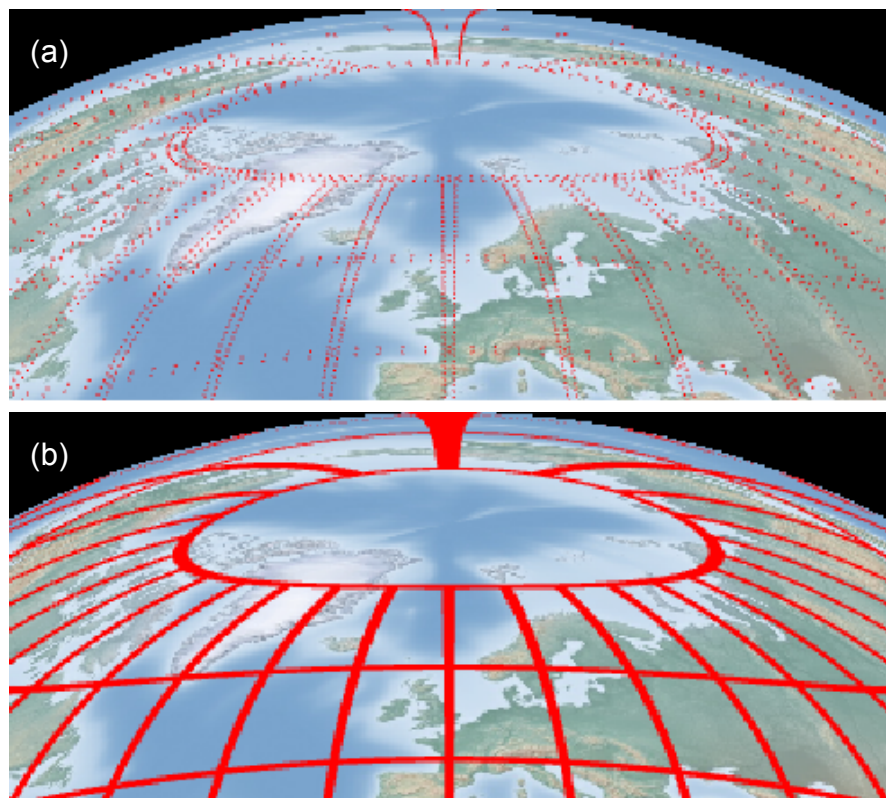


Abbildung 36: Darstellung von Liniensignaturen auf der Globusabbildung (Ausschnitt)

(a) anhand sphärischer Trigonometrie generierte Neupunkte für die Darstellung einer Liniensignatur; (b) Neupunkte zu einem TRIANGLE_STRIP vernetzt
(Quelle: eigene Darstellung, topographische Grundlage: [NED-12])

Separate Außenlinien können auf dieselbe Art und Weise, unter Verwendung einer größeren Strichstärke und einer anderen Farbe, generiert werden. Bei der Bildung des TRIANGLE_STRIPs für die Außenlinie müssen die neu generierten Punkte (äußere

Begrenzung der Außenlinie) sowie die äußere Begrenzung der eigentlichen Linie (und somit die innere Grenze der Außenlinie) verwendet werden.

6.2.2 Ogre3D / Vektorgeometrie

Wie bei dem Verfahren zuvor, werden Liniengeometrien als TRIANGLE_STRIPs gerendert. Bei diesem Ansatz werden die Vertices der Dreiecksnetze entlang der virtuellen Kugeloberfläche platziert.

Auch hier findet die Segmentierung bzw. Verfeinerung (wenn nötig) in einem iterativen Prozess statt und ist notwendig um die Krümmung der Erdkugel (bzw. den Kreisbogen/die Orthodrome) zwischen zwei Punkten der Ursprungsgeometrie zu simulieren.

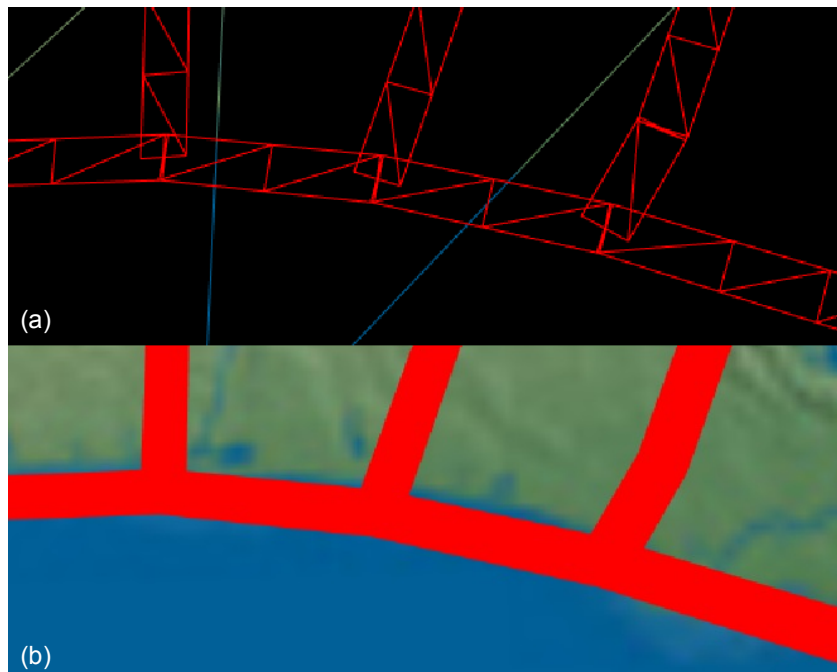


Abbildung 37: Liniengeometrien mit TRIANGLE_STRIPs visualisiert

*Die Wiedergabe von Liniensignaturen in Ogre3D; (a) Drahtgittermodell; (b) Normalansicht
(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])*

Um die Visualisierungsgeometrie zu erhalten wird die Liste der Vertices eines LineStrings mit einer Schleife abgearbeitet. Je Punkt werden folgende Berechnungen durchgeführt. Zuerst wird die Normale auf die Ebene, die durch den Startpunkt, den Endpunkt und den Ursprung verläuft, aufgestellt. Dieser Normalvektor wird anschließend normiert und mit der halbierten Strichstärke (in Kilometer) multipliziert. Durch Addition bzw. Subtraktion wird der Normalvektor mit der richtigen Länge an den Punkt einmal links und einmal rechts an den ursprünglichen Punkt angehängt. Die dadurch berechneten Punkte (linke und rechte Außenbegrenzungen der Liniensignatur)

werden nun auf die Kugeloberfläche transformiert (Normierung des Ortsvektors mit anschließender Multiplikation des Erdradius).

Wie zuvor sind Start- bzw. Endpunkte anders zu behandeln als Punkte innerhalb der Geometrie, bei denen die Normalvektoren beider Segmente in der Berechnung berücksichtigt werden müssen.

Die Punkte werden nun miteinander zu einer Folge von Dreiecken (TRIANGLE_STRIP) zur Darstellungsgeometrie vernetzt. Abbildung 37 zeigt eine Nahansicht in Ogre3D.

6.2.3 PovRay / CSG

Zwei Punkte auf der Kugeloberfläche sollen entlang der kürzesten Strecke zu einem Liniensegment verbunden werden. Dabei handelt es sich um die Orthodrome, die ein Teil eines Großkreises ist. In der Äquatorebene wird ein Großkreis, in Form eines Rings (Torus), konstruiert. (Abbildung 38 (a))

In Abbildung 38 (b) wird dieser Torus anhand zweier Rotationsmatrizen an die gewünschte Position gedreht. [FIL-12] bespricht die „Darstellung von Punkten, Vektoren, Strecken, Geraden und Ebenen mithilfe von POV-Ray“ und beschreibt u.a. das „Erzeugen von Ebenen [...] aus einer Parameterdarstellung“, wofür u.a. die eingesetzten Rotationsmatrizen verwendet werden.

Im nächsten Schritt (Abbildung 38 (c-e)) wird der Torus „zugeschnitten“. Dazu wird jene Ebene konstruiert, die durch den Startpunkt und den Koordinatenursprung geht und normal auf die Ebene steht, in der der Großkreis liegt. Genauso wird eine Ebene erstellt, die durch den Endpunkt und durch den Ursprung verläuft und ebenfalls normal auf den Großkreis steht. Nun wird der Torus mit den Innenseiten dieser zwei Ebenen verschnitten. Das Ergebnis ist die Orthodrome vom Start- bis zum Endpunkt. Um nun die fertige Signaturgeometrie zu erhalten muss das Ringsegment mit der virtuellen Kugeloberfläche verschnitten werden. Der Schnittbereich ist in Abbildung 38 (g) dargestellt.

Gilt es nun einen LineString zu visualisieren, so werden die einzelnen Ringsegmente, bevor sie mit der virtuellen Erdkugel geschnitten werden, mit dem *union* Operator zu einem Objekt verbunden. Ein Beispiel dafür ist in Abbildung 38 (h) zu sehen.

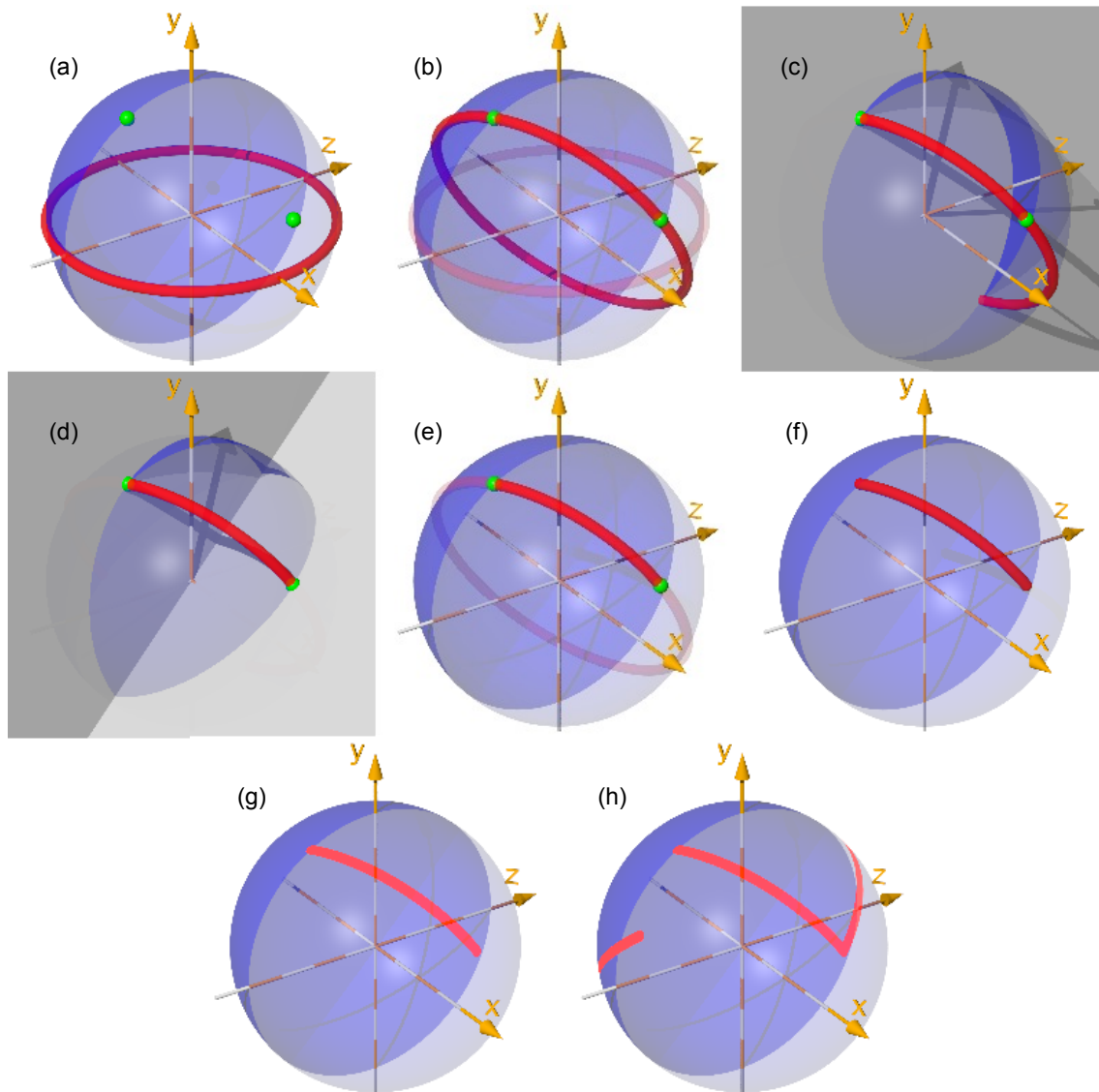


Abbildung 38: Generierung von Liniensegmenten bzw. LineStrings mit PovRay und CSG
 (a) Start- und Endpunkt des Liniensegments (grün) Torus (Ring) in der Äquatorebene (rot); (b) Rotation des Rings an die gewünschte Position; (c-e) Zuschnitt des Rings mit den Ebenen, die jeweils durch den Start- bzw. Endpunkt und den Ursprung verlaufen sowie normal auf den Großkreis stehen; (f) Liniensegment mit abgerundeten Enden (Kugelobjekte); (g) Schnittbereich von Ringsegment und virtueller Kugeloberfläche ergibt fertige Darstellungsgeometrie; (h) Darstellungsgeometrie eines LineStrings (Verbindung mehrerer Liniensegmente)
 (Quelle: eigene Darstellung)

Im Unterschied zu den bereits besprochenen Verfahren müssen die Eingangsvektordaten nicht segmentiert werden. Die Krümmung (entlang der Erdoberfläche bei dem 3D Mesh Ansatz) bzw. die Verzerrung (in Abhängigkeit der Abbildung bei dem 2D Mesh Ansatz) von Liniensegmenten wird hier nicht durch Segmentierung simuliert. Eine Orthodrome bzw. ein Segment einer Liniengeometrie wird durch ein geometrisches Objekt, einen Kreisbogen, umgesetzt.

Die möglichen (grafischen) Variationen, die im Zuge der Diplomarbeit umgesetzt wurden, sind in Abbildung 39 dargestellt. Neben der Farbe, dem Tonwert, der Strichstärke und der Füllung (Füllung und/oder Außenlinie) können auch die Enden unterschiedlich gestaltet werden. In diesem Fall kann zwischen runden und stumpfen Enden gewählt werden. Wie mit den Verbindungen zwischen einzelnen Liniensegmenten innerhalb eines LineStrings umgegangen werden kann, wird im folgenden Abschnitt besprochen.

Eine Außenlinie wird durch ein zweites Ringsegment, das an der selben Position platziert wird und einen etwas größeren Durchmesser hat, umgesetzt. Das innere Ringsegment wird anschließend von dem äußeren abgezogen. Abschließend wird wiederum der Schnittbereich zwischen dem tlw. hohlen Ringsegment und der virtuellen Erdkugel generiert.

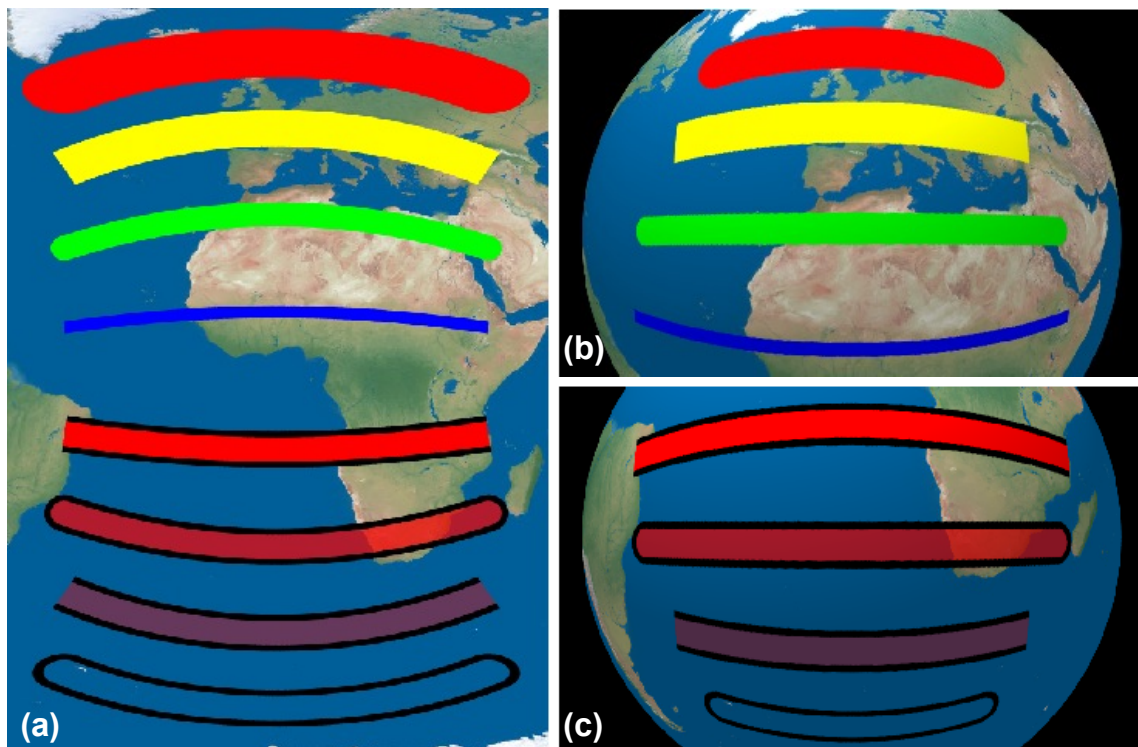


Abbildung 39: Grafische Variationsmöglichkeiten von Liniensignaturen (PovRay)
 Möglich (grafische) Variationen: Farbe, Strichstärke, Füllung, Tonwert, Enden (rund, stumpf)
 Darstellung (a) auf der Platkarte, (b) auf dem Globus.
 (Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

6.2.4 Verbindungen zwischen Liniensegmenten

Wird nun ein LineString, mehrere aneinander gereihte Liniensegmente, visualisiert, muss auf die Verbindung zwischen den Segmenten geachtet werden. In Abbildung 40 (a) wird dies nicht berücksichtigt.

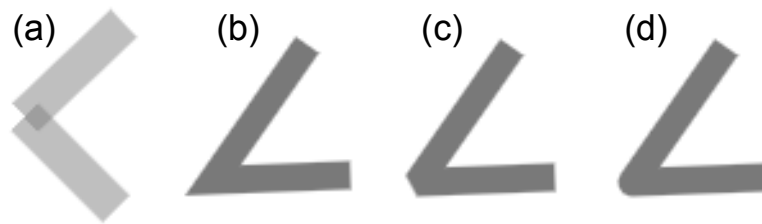


Abbildung 40: Unterschiedliche Verbindungsarten von Liniensegmenten

(a) ohne Berücksichtigung der Verbindung; (b) miter; (c) bevel; (d) round

(Quelle: [TSA-11])

[TSA-11] setzt sich in einem Beitrag über das Zeichnen von zweidimensionalen Linien mittels Tessellation mit den unterschiedlichen Verbindungstypen zwischen Liniensegmenten auseinander. Gängige Vektorprogramme und Vektorgrafikbibliotheken bieten folgende Verbindungsarten an: *miter*, *bevel* und *round*. (Abbildung 40).

Die bei [TSA-11] besprochenen Algorithmen, ähnlich wie die Wiedergabe der Liniensignaturen durch TRIANGLE_STRIPs, basieren ebenfalls auf Triangulierung. Je nach Winkel und gewählten Verbindungstyp werden die Dreiecke angeordnet. Abbildung 41 zeigt die Umsetzung für eine runde Verbindung (a) als TRIANGLE_FAN und eine eckige (b) als TRIANGLE_STRIP.

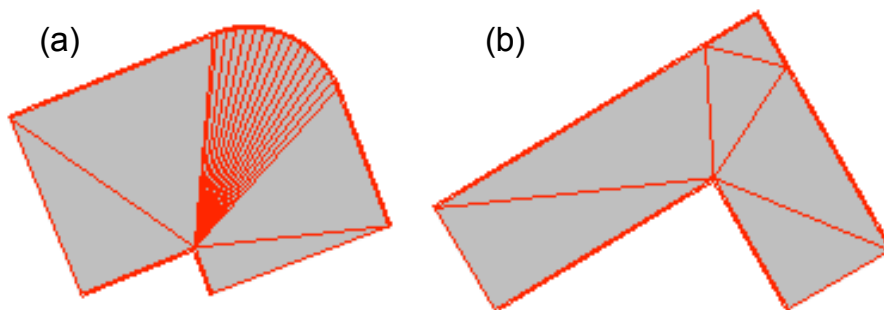


Abbildung 41: Beispiel für Verbindungstypen als Dreiecksnetz

(a) Verbindungstyp round als TRIANGLE_FAN; (b) miter-Verbindung als TRIANGLE_STRIP

(Quelle: [TSA-11])

Die zweidimensionalen Ansätze von [TSA-11] können ohne größere Schwierigkeiten in den dreidimensionalen Raum überführt werden, wurden im Zuge der Diplomarbeit jedoch nicht implementiert.

Bei Ansatz 2 (Ogre3D) wird zum aktuellen Stand etwas „geschummelt“ (Abbildung 42). Wie in der Wireframeansicht (a) zu erkennen ist, überlagern sich die benachbarten Dreiecke bei den Segmentverbindungen. Solche Überlagerungen sollten vermieden werden, führen aber in diesem Fall sehr einfach und schnell zu einem visuell akzeptablen Ergebnis (b), das mit dem bevel-Verbindungstyp vergleichbar ist.

Bei der Visualisierung mittels PovRay (Ansatz 3) ist die Umsetzung für runde Verbindungen sehr einfach zu implementieren. Bei jeder Verbindung von Liniensegmenten wird eine Kugel mit dem Durchmesser der Ringsegmente platziert und mit diesen verschmolzen, bevor die Schnittfläche gebildet wird (Abbildung 38 (f)).

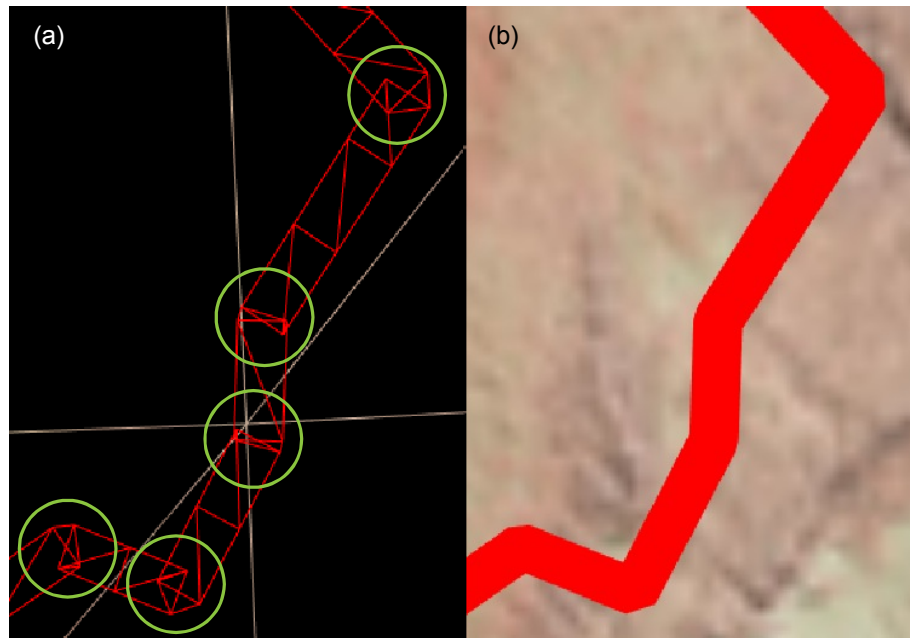


Abbildung 42: Verbindung von Liniensegmenten bei Ansatz 3 (Ogre3D)

In Wireframeansicht (a) zeigen die grünen Kreise die „unsauberen“ Verbindungen, die in der Normalansicht (b) zu einer visuell akzeptablen Darstellung führen.

(Quelle: eigene Darstellung, topographische Grundlage: [PAT-12])

6.2.5 Groß- und Kleinkreise (Meridiane und Parallelen)

Der Mittelpunkt eines Großkreises auf der Kugeloberfläche ist ident mit jenem der Kugel (folglich ist auch der Radius ident). Daraus ergibt sich, dass ein Großkreis immer die Kugel halbiert. Ein Großkreis kann also durch die Angabe zweier Punkte exakt definiert werden. Ein Kleinkreis, er liegt beliebig auf der Kugeloberfläche, kann durch die Angabe zweier Punkte nicht eindeutig definiert werden. Es wird ein weiterer Punkt, der Teil des Kleinkreises ist, benötigt.

Ausnahmen sind die Parallel- oder Breitenkreise, die normal auf die Erdachse stehen. Hier reicht die Angabe der geographischen Breite. Meridiane (Längengrade) sind hingegen halbe Großkreise, die vom Nordpol zum Südpol verlaufen. Der Äquator ist als einzige Parallele ein Großkreis.

Bei den Ansätzen 1 und 2 wird ein Meridian durch einen TRIANGLE_STRIP, der von Pol zu Pol verläuft, generiert. Mit den Koordinaten der Pole sowie einem Stützpunkt am Äquator, der die geographische Länge angibt, wird die Grundgeometrie erzeugt. Durch

den Schritt der Segmentierung wird wiederum die Erdkrümmung simuliert. Breitenkreise werden durch die Rotation eines Punktes auf der entsprechenden Breite um die Erdachse erzeugt. Bei einer etwaigen Verfeinerung ist darauf zu achten, dass neue Stützpunkte nicht entlang der Orthodrome, sondern entlang der Loxodrome zu setzen sind.

Bei Ansatz 3 wird ein Großkreis (auch ein Meridian), so wie in Abbildung 38 (a) und (b) dargestellt ist, erzeugt und platziert. Ein Parallelkreis ist bspw. in Abbildung 34 (d) zu sehen. Dazu wird ein entsprechend großer Ring in der Äquatorebene generiert und anschließend entlang der y-Achse (Erdachse) an die korrekte Position verschoben.

Ein Beispiel, bei dem Groß- und Kleinkreise verwendet werden, ist die Visualisierung eines Gradnetzes (Meridiane und Parallelen). Ein Gradnetz kann mathematisch (zur Laufzeit) generiert werden und muss nicht auf einer Vektordatei beruhen. Je nach gewünschtem Abstand zwischen den Breitenkreisen bzw. Längenkreisen (in Grad) kann durch eine Schleife die benötigte Geometrie generiert werden. Im Anhang sind Beispiele von Gradnetzen zu finden.

6.3 Flächengeometrien

Nachdem nun erläutert wurde, wie anhand der unterschiedlichen Ansätze Punkt- und Liniengeometrien in ihre entsprechende Visualisierungsgeometrie überführt werden, wird abschließend eine Möglichkeit erörtert Flächengeometrien zu visualisieren.

Bei der Wiedergabe von Flächengeometrien muss bei allen Ansätzen eine Triangulierung der Flächen vorgenommen werden. Bei den Punkt- und Liniengeometrien war nur bei den Ansätzen 1, 2a und 2b eine Dreiecksvernetzung für die Visualisierung notwendig. Nun müssen auch bei Ansatz 3 Dreiecke gebildet werden. Auf den Schritt der Verfeinerung kann jedoch trotzdem verzichtet werden.

Es gibt unterschiedliche Triangulierungsalgorithmen. Ein sehr einfacher ist das sog. Triangulieren durch Ear Clipping. Ein einfaches Polygon mit n Ecken kann in $n-2$ Dreiecke unterteilt werden.

[EBE-08] beschreibt Ear Clipping in der Ebene. [COZ-11] übertragen diesen Algorithmus in den dreidimensionalen Raum für die Visualisierung von Polygonen in virtuellen digitalen Globen. Dieser Ansatz kann in den Ansätzen 2a und 2b verwendet werden. Eine Alternative für die Triangulierung eines sphärischen Polygons im dreidimensionalen Raum ist die Projektion des Polygons auf eine Tangentialebene, auf der die Triangulierung im zweidimensionalen Raum durchgeführt wird. Abschließend werden die Punkte auf die Oberfläche des Referenzkörpers rück-projiziert. [COZ-11]

Im nächsten Abschnitt wird besprochen, wie ein sphärisches Dreieck visualisiert wird. Anschließend wird auf die Triangulierung eines sphärischen Polygons eingegangen. Der umgesetzte Triangulierungsansatz basiert ebenfalls auf Ear Clipping im dreidimensionalen Raum, wurde aber für PovRay adaptiert.

6.3.1 Beliebige (sphärische) Dreiecke (PovRay)

Da Polygone anhand von Dreiecken dargestellt werden können, wird zuerst geschildert, wie ein beliebiges sphärisches Dreieck mit PovRay konstruiert werden kann.

Ausgehend von den Eckpunkten des Dreiecks (Abbildung 43 (a)) wird ein allgemeiner Tetraeder gebildet, dessen Spitze im Koordinatenursprung (bzw. im Kugelmittelpunkt) liegt und dessen Kanten durch die Eckpunkte des Dreiecks verlaufen (Abbildung 43 (b)).

Abschließend wird, wie bereits erläutert, der Schnittbereich zwischen dem Tetraeder und der virtuellen Erdoberfläche gebildet (Abbildung 43 (c)). Die Seiten eines sphärischen Polygons verlaufen entlang der jeweiligen Orthodrome.

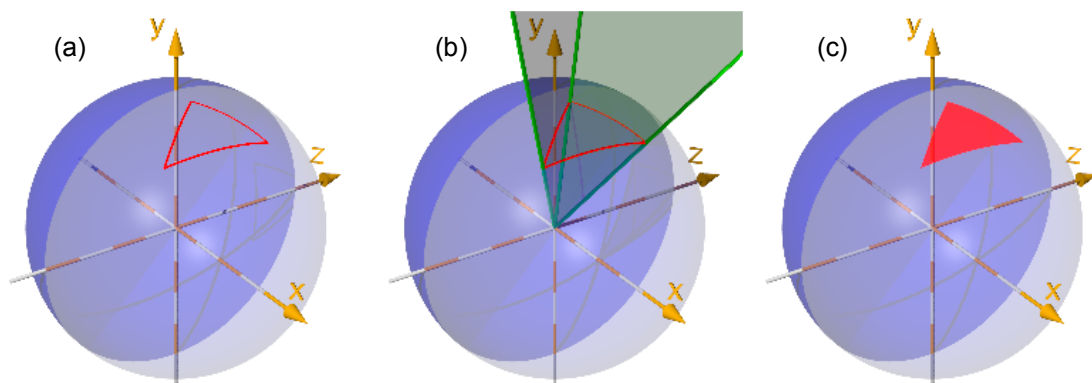


Abbildung 43: Generierung eines sphärischen Dreiecks (PovRay)

(a) das darzustellende Dreieck; (b) Bildung eines Tetraeders; (c) Schnittfläche von Tetraeder und virtueller Kugel ist die Visualisierungsgeometrie

(Quelle: eigene Darstellung)

Eine Voraussetzung bei der Bildung sphärischer Dreiecke (bzw. bei der Triangulierung sphärischer Polygone) ist bei diesem Verfahren, dass keine der drei Seiten länger als ein halber Großkreis sein darf. Ansonsten ist die Bestimmung der Innen- oder Außenseite des Dreiecks (oder Polygons) kein triviales Problem mehr. Diese und weiterer Einschränkungen bei der Triangulierung sphärischer Polygone werden bei [ORO-08] diskutiert.

6.3.2 (Sphärische) Polygone und deren Triangulierung (PovRay)

Im Folgenden wird der Algorithmus zum Ear Clipping, wie er im Zuge der Diplomarbeit umgesetzt wurde und in Abbildung 44 abgebildet ist, beschreiben.

Unter der Beachtung der Drehrichtung des zu triangulierenden linearen Rings werden beim Ear Clipping dreieckige „Ohren“ von der ursprünglichen Geometrie Schritt für Schritt abgeschnitten.

Ausgehend von dem Startpunkt muss zuerst kontrolliert werden, ob es sich um ein konvexes oder konkaves Eck handelt. Der Innenwinkel muss konkav sein, damit ein Eck als „Ohr“ in Frage kommt. Für diese Ermittlung müssen die benachbarten Punkte miteinbezogen werden. Punkt P in Abbildung 44 (a) ist bspw. konvex und wird vorerst übersprungen. Bei Punkt P in (b) trifft die Kondition zu, der Innenwinkel ist konkav.

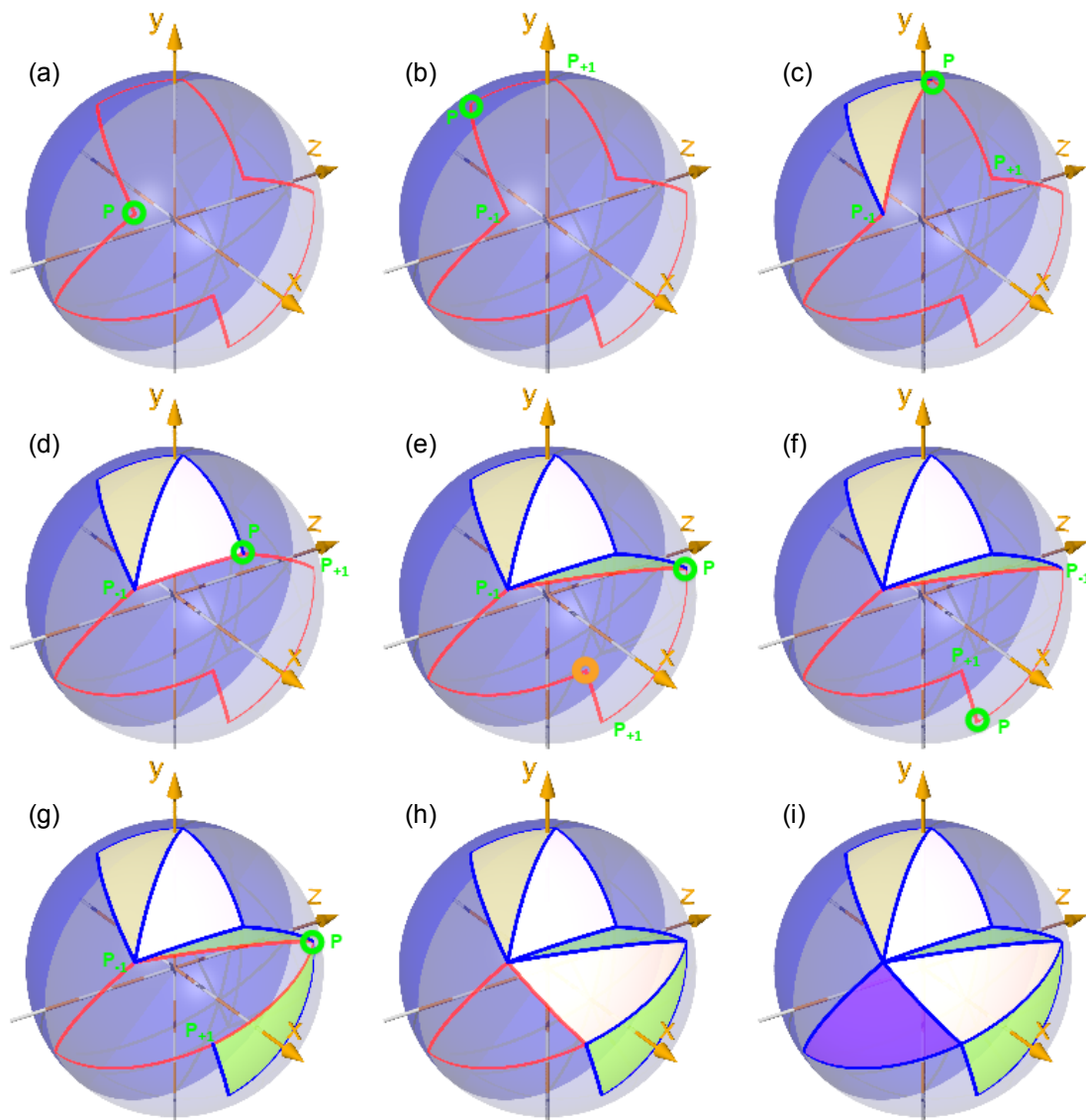


Abbildung 44: Triangulierung eines sphärischen Polygons mittels Ear Clipping (PovRay)

(a) Innenwinkel konvex = kein „Ohr“; (b), (c), (d), (f), (g) Innenwinkel konkav, kein Stützpunkt innerhalb des Dreiecks = „Ohr“ das „geclippt“ wird; (e) Innenwinkel konkav, Stützpunkt (orange) innerhalb des Dreiecks = kein „Ohr“; (h) das ursprüngliche Polygon wurde zu einem Dreieck verkleinert, der Algorithmus ist fertig, das Polygon ist trianguliert (i).

(Quelle: eigene Darstellung)

Neben dieser ersten Bedingung gibt es eine zweite, die ein Eck zu erfüllen hat, damit es als „Ohr“ von der Geometrie entfernt werden kann. Wird zwischen den zwei benachbarten Punkten (P_{-1} bzw. P_{+1}) eine Verbindung gezogen, also ein Dreieck mit den Eckpunkten P , P_{-1} und P_{+1} gebildet, darf kein anderer Stützpunkt des Polygons innerhalb dieses Dreiecks liegen. Wie bei der Konstruktion eines sphärischen Dreiecks wird ein Tetraeder (mit Spitze in der Kugelmitte und Seitenkanten durch die Eckpunkte) gebildet. Liegt nun ein Ortsvektor der anderen Stützpunkte innerhalb des Tetraeders, kommt diese Ecke nicht als „Ohr“ in Frage (Abbildung 44 (e)) und der Algorithmus fährt mit dem nächsten Eckpunkt fort.

Liegt kein anderer Vertex innerhalb des Dreiecks und ist der Innenwinkel des Eckpunkts P konkav, dann kann das Dreieck P , P_{-1} und P_{+1} von dem Polygon weggeschnitten (geclippt) und wie in Kapitel 6.3.1 beschrieben visualisiert werden (z.B. Abbildung 44 (c)).

Das zu bearbeitende Polygon (Außenlinie in Abbildung 44 rot dargestellt) wird nun Schritt für Schritt mit der beschriebenen Vorgangsweise weiter in Dreiecke (blaue Außenlinien) unterteilt.

Hat das abzuarbeitende Polygon nur mehr drei Eckpunkte, ist also selbst zu einem Dreieck geworden, ist das Ende des Algorithmus erreicht (Abbildung 44 (h)). Das letzte Dreieck wird abschließend eingezeichnet (Abbildung 44 (i)).

Bei 3D Ansätzen, bei denen klassische Dreiecksnetze (Kapitel 5.4.3) produziert werden, müssten nun diese Dreiecke für eine Annäherung an die Krümmung der Kugel verfeinert werden. [COZ-11] besprechen einen Segmentierungsalgorithmus für die Konstruktion von virtuellen Globenkörpern, der jedoch auch für diese Zwecke eingesetzt werden kann.

Besitzt eine Flächengeometrie Löcher (Kapitel 3.1.1), muss zwischen dem Schritt der Generalisierung und der je nach Ansatz notwendigen oder optionalen Verfeinerung ein weiterer Schritt eingefügt werden. Eine sehr einfache Möglichkeit Polygonflächen, die Löcher haben, zu visualisieren, ist das „Aufschneiden“ des Polygons. Zwischen zwei möglichst benachbarten Stützpunkten des inneren bzw. äußeren Rings wird ein „Schnitt“ durchgeführt (rot Linie in Abbildung 45). Es entsteht ein einziges Polygon, das wiederum trianguliert werden kann. Bei diesem Verfahren, beschrieben bei [EBE-08], ist besondere Berücksichtigung auf die Drehrichtung der jeweiligen Ringe zu legen. Äußere und innere Ringe müssen eine entgegengesetzte Drehrichtung aufweisen, damit das für die Triangulierung erzeugte Polygon sich selbst nicht schneidet und zu einem komplexen Polygon wird.

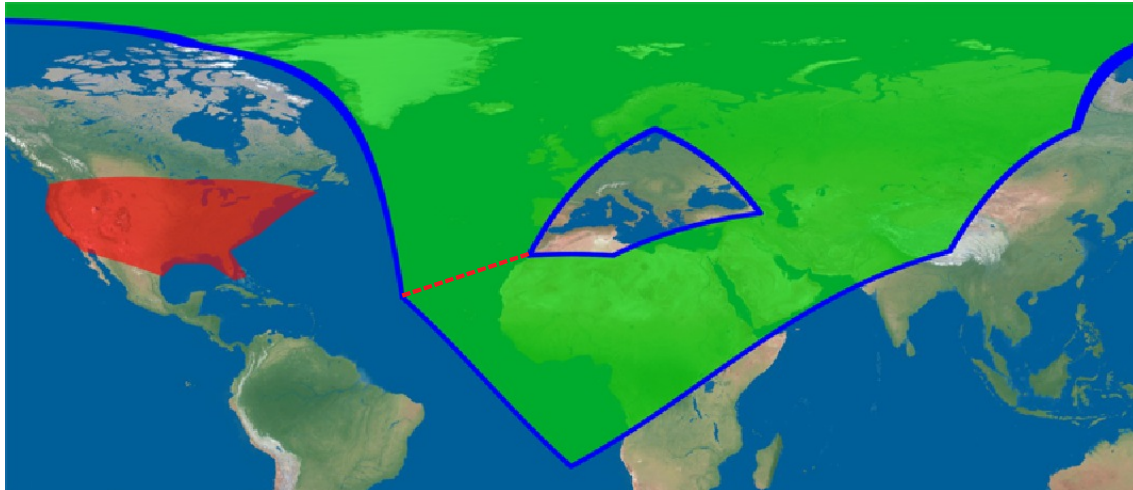


Abbildung 45: Graphische Variation von Flächensignaturen (mit Löchern) in PovRay

Variation der Farbe und des Tonwerts; Optionale Darstellung der Außenlinien;

Rote Linie: Schnitt durch das Polygon, um die Triangulierung durchführen zu können.

Ausschnitt aus der Plattkarte

(Quelle: eigen Darstellung, topographische Grundlage: [PAT-12])

Außenlinien (innerer und äußerer Ring) können dann separat, ohne Berücksichtigung des „Schnittes“, visualisiert werden. (Abbildung 45).

In derselben Abbildung sind auch die möglichen graphischen Variationen, die bei der Wiedergabe von Flächengeometrien bei dieser Umsetzung verwendet werden können, dargestellt. Neben dem Farb- und dem Tonwert kann, wie bei allen anderen Signaturtypen bei Ansatz 3, die Transparenz verändert werden.

6.4 Fallbeispiele

Im Anhang sind u.a folgende Visualisierungen von Punkt-, Linien- und Flächengeometrien zu finden:

- Vulkanstandorte
- Städte
- Gradnetz
- Flugroute über Pol
- Schiffstrack
- Staatsgrenzen
- Ergebnis der Polygontriangulierung aus Kapitel 6.3.2

7 Diskussion

In der abschließenden Diskussion werden die oben vorgestellten Verfahren (Kapitel 5 und 6) einem Vergleich, in Bezug auf unterschiedliche Faktoren, unterzogen und deren Vor- und Nachteile erörtert (Abbildung 46). Dadurch wird versucht die eingangs gestellten Forschungsfragen zu beantworten. Zur Erinnerung:

- Welche Methoden und Lösungsansätze sind für die Wiedergabe von Punkt-, Linien- bzw. Flächengeometrien am sphärischen Display geeignet? Welche Probleme und Hindernisse müssen bei der Visualisierung von Vektordatensätzen berücksichtigt werden?
- Welche Anforderungen sind an Geo-Vektordatensätze als Datenquelle für die Visualisierung am sphärischen Display zu stellen?

Die Möglichkeiten, die ArcGIS bietet, werden ebenfalls in die Schlussbetrachtung miteinbezogen (Ansatz 0). Zu beachten ist jedoch, dass dieser Ansatz nicht allen Anforderungen, die durch die anderen erfüllt werden, gerecht wird (Verzerrungen und Nähte).

Für die Visualisierung von Punktgeometrien (Abbildung 46 (a)) wird der Ansatz 3 (PovRay und CSG) als sehr geeignet betrachtet. Mit Hilfe von CSG oder Schriftsätzen könnten unterschiedlichste, auch nicht geometrische, sondern bildhafte Signaturen sehr einfach generiert werden. Die Verfahren, die Signaturen durch Dreiecksnetze generieren (Ansätze 1, 2a und 2b), eignen sich auch gut für die Wiedergabe von Punktgeometrien, beschränken sich jedoch auf geometrische Signaturen. Die Konstruktion bildhafter Signaturen wäre hier mit einem höheren (programmiertechnischen) Aufwand verbunden. Bei ArcGIS können lediglich kreisförmige Punktsignaturen korrekt visualisiert werden. Andere Formen sind nicht möglich.

Bei den Liniensignaturen (Abbildung 46 (b)) verhält es sich ähnlich. Es ist zwar möglich mit ArcGIS Linien zu bilden, bei der die Stützpunkte entlang der Orthodrome verbunden werden. Die Verzerrung der Platkarte kann aber nicht berücksichtigt werden, die Strichstärke der Linie ist unabhängig von der Lage konstant. Die Möglichkeit einen geodätischen Puffer zu bilden, wie es bei kreisförmigen Punktsignaturen funktioniert (Kapitel 4.1.1.2), gibt es für Liniendaten in ArcGIS derzeit nicht, soll aber ab der Version 10.1 zur Verfügung stehen.

Bei den Ansätzen, bei der die Visualisierung im Objektraum 3D umgesetzt wird, können Liniengeometrien sehr gut in ihre jeweilige Darstellungsgeometrie überführt

werden. Weder Verzerrungen noch Nähte müssen berücksichtigt werden. Hier liegt einer der größten Vorteile der drei- im Vergleich zu den zweidimensionalen Ansätzen. Bei Ansatz 1 muss die Verzerrung der Abbildung genau bekannt sein (Abbildungsfunktion sowie Inverse), damit die Signaturen entsprechend den Verzerrungen eingezeichnet werden können. Auch der Außenrand der azimuthalen Globenabbildung kann Probleme bereiten und es kann, wird dieser Effekt nicht abgefangen, zu überdimensionierten, den Rest der Darstellung abdeckenden Dreiecken kommen, deren Eckpunkte weit von einander entfernt auf dem „Außenkreis“ liegen.

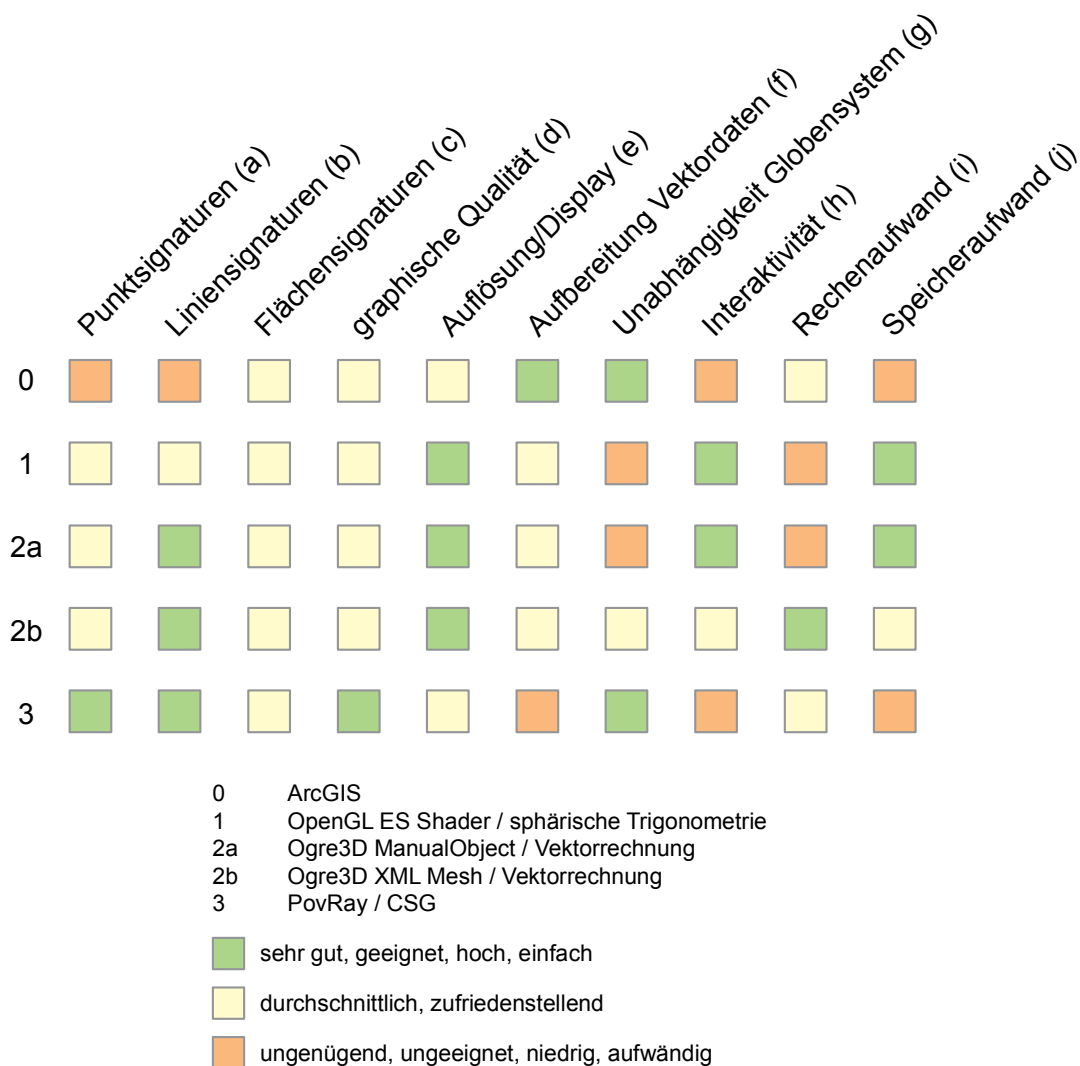


Abbildung 46: Bewertung der besprochenen Ansätze

Bewertung der besprochenen Ansätze (0, 1, 2a, 2b, 3) anhand unterschiedlicher Faktoren (a – j)
(Quelle: eigene Darstellung)

Bei den Flächengeometrien (Abbildung 46 (c)) werden alle vorgestellten Ansätze als durchschnittlich bewertet. Die Triangulierungen sind jeweils sehr aufwendige Prozesse, die je nach Verfahren auch an bestimmte Einschränkungen gebunden sind. Bei ArcGIS

(oder jedem anderen GIS) wird für die Visualisierung von Flächen keine Triangulierung vorgenommen werden. Die Bewertung bezieht sich jedoch auf real existierende Vektordatensätze mit einer gewissen Dichte an Vertices, wobei die Polygone aufgrund der Verzerrungen und der Datumsgrenze nicht mit einer Außenlinie visualisiert werden können.

Aus kartographischer Perspektive kann die höchste graphische Qualität (Abbildung 46 (d)), in Bezug auf eine feingliedrige Darstellung, am einfachsten mit Ansatz 3 erreicht werden. Bei den anderen Ansätzen können durch eine sehr feine Segmentierung bzw. Tessellation, bei vergleichsweise hohem Rechenaufwand, ähnliche Ergebnisse erzielt werden. In Abbildung 46 (e) wird hingegen das Erscheinungsbild der Vektordaten am sphärischen Display adressiert. Hier kann mit den Verfahren, die 2D oder 3D Modelle als Ausgabeformat haben, ein leicht besseres Ergebnis erzielt werden.

Bei Ansatz 3 ist ein Nachteil, dass PovRay geographische Vektordaten nicht einlesen kann. Dazu wird ein weiteres Programm benötigt, das zuvor die Geometrien, Attribute und ev. Stildefinitionen ausliest und als Textdatei speichert, die vom PovRay Skript eingelesen werden können. Eine Alternative ist die Erzeugung einer separaten PovRay Datei (*.pov), in der alle Objekt-Definitionen (CSG) geschrieben werden. Das kann bspw. durch das die Vektordaten einlesende Programm umgesetzt werden. Bei allen anderen Verfahren werden die Daten im selben Programm eingelesen, das auch für die Visualisierung zuständig ist. Hierbei werden, ausgenommen von ArcGIS, das als GIS selbstverständlich Vektordaten einlesen kann, bereits bestehende Bibliotheken verwendet (z.B. OGR). Daraus ergibt sich die Bewertung in Abbildung 46 (f).

Ein Vorteil der externen, asynchronen Verfahren, die 2D Basisraster ausgeben, ist die Unabhängigkeit vom Globensystem (Ansatz 0 und 3) (Abbildung 46 (g)). Da die vorgestellten gängigen Globensysteme alle eine quadratische Platkarte als Basismedium verwenden, kann nur mit solchen Verfahren, mit denen Platkarten erzeugt werden, eine Plattformunabhängigkeit gewährleistet werden. Der Ansatz 2b ist zwar auch extern und asynchron, ist aber aufgrund des Ausgabeformates fest an Ogre3D und somit an OmniSuite gebunden. Somit ist bei den Verfahren mit vorgerenderter Ausgabe die Interaktion eingeschränkt.

Um auf Interaktionen (Abbildung 46 (h)), die durch die Betrachtenden getätigt werden, zu reagieren, muss die Vektordatenintegration eine interne (also synchrone) Lösung sein. So könnte bspw. auf nicht vorgerenderte, spontane Rotationen oder Selektionen reagiert werden. Bspw. könnten Punktsignaturen, wie bei Earthbrowsern, bei Rotationen gerade ausgerichtet werden, damit sie bei einer größeren Rotation des

Globus nicht am Kopf stehen. Hier ist jedoch die Integration in die Globensoftware eine unbedingte Voraussetzung, die sich auch im Vergleich der Faktoren Interaktivität und Unabhängigkeit vom Globensystem widerspiegelt.

Darüber hinaus besteht ein Zusammenhang zwischen dem Rechenaufwand (Abbildung 46 (i)) zur Generierung der Darstellungsgeometrie und dem Speicheraufwand der Ausgabe (Abbildung 46 (j)). Interne Verfahren, die zur Laufzeit die Daten aufbereiten und visualisieren, sind mit einem eher hohen Rechenaufwand verbunden. Das resultiert jedoch in einem tendenziell geringen Speicherbedarf. Bei externen Methoden wird die Visualisierung einmal durchgeführt und resultiert bspw. in einem Rasterbild, das je nach Auflösung, einen vergleichsweise großen Speicherbedarf hat. Der Rechen- sowie der Speicheraufwand hängen natürlich auch sehr direkt von der Anzahl der dargestellten Features bzw. deren Genauigkeit (Menge der Punkte) ab.

Neben den bereits genannten Problemen können je nach Verfahren weitere Hindernisse auftreten.

Bei dem Verfahren mit PovRay kann es passieren, dass bei sehr vielen Objekten und CSG-Operationen eine intern definierte Schranke an möglichen Operationen erreicht wird und somit der Rendervorgang abgebrochen wird. Diesem Hindernis kann durch eine geschickte Methode des Einlesens der Vektordaten entgegengewirkt werden. Dieses Problem tritt aber bei den Ansätzen, bei denen ein Mesh für die Darstellungsgeometrie generiert wird, nicht auf. Moderne Grafikkarten sind im Stande pro Sekunde Vertices im Milliardenbereich zu verarbeiten.

Bei den 3D Ansätzen ist das Problem des Z-Fightings präsent. Wenn Flächen unterschiedlicher Objekte idente Bereiche haben, kann die Render-Engine oft nicht entscheiden, welche Teile der jeweiligen Objekte sichtbar sein sollen. Hier gibt es jedoch Algorithmen und Techniken die Wahrscheinlichkeit des Auftretens zu verringern.

Im Zuge der Diplomarbeit hat sich herausgestellt, dass neben technischen auch inhaltliche Anforderungen an Vektordaten zu stellen sind. Eine der wichtigsten Anforderungen, um den Rechenaufwand zu optimieren, ist die maßstabsabhängige Generalisierung der Daten, wodurch die Anzahl der Vertices verringert werden kann.

Des Weiteren müssen die Vektordaten in einem geographische Koordinatensystem vorliegen oder ggf. um-projiziert werden.

Je nach Verfahren bzw. den Vektordaten selbst muss vor der Visualisierung eine Segmentierung bzw. Verfeinerung entlang der Orthodrome vorgenommen werden um

die Verzerrung der Abbildung bzw. die Krümmung der Erde zu simulieren. Hier ist ein Mittelweg zwischen graphischer Genauigkeit und Rechenaufwand zu wählen.

Die Segmentierung ist jedoch bei tatsächlichen Datensätzen, wie bspw. bei den Staatsgrenzen von Natural Earth Data im kleinsten Maßstab, nur vereinzelt oder gar nicht notwendig. Bei Verbindung einer Fluglinie (z.B. Wien (VIE) – Portland (PDX)), bei der nur die Flughafenkoordinaten bekannt sind, ist eine Segmentierung entlang der Orthodrome unbedingt notwendig um eine korrekte Darstellung zu erhalten. Bei Ansatz 3 (PovRay) kann auf diese Segmentierung der Vektordaten verzichtet werden. Die einzelnen Punkte werden nicht durch lineare Segmente im Raum verbunden, sondern durch Geometrien, durch Kreisbögen, die entlang der Orthodrome zwischen zwei Punkten verläuft.

Eines der größten Probleme bei der Integration von Vektordaten mittels Plattkarten ist deren Verzerrung, die zu den Polen hin zunimmt. Die vorgestellten Verfahren berücksichtigen diese Problematik. Solange aber keine Geometrien im unmittelbaren Polbereich sind, ist eine Aufbereitung ohne Berücksichtigung der Verzerrung, vor allem in Bezug auf den Zeitaufwand, vertretbar.

Die Diskussion zeigt abschließend, dass es keinen Sinn ergibt ein einziges Verfahren als optimale Lösung zu präsentieren. Vielmehr sind Aspekte einzelner Verfahren, ob intern oder extern, ob im Objektraum 2D oder 3D oder ob mittels Meshgenerierung oder CSG, für unterschiedliche Zwecke und Ziele besser oder schlechter geeignet.

Für den Autor hat sich jedoch im Zuge der Diplomarbeit eine gewisse Präferenz für interne Lösungen herauskristallisiert. Solche Verfahren bieten eine sehr große Möglichkeit die gegenwärtigen Globensysteme voran zu treiben und, aus Sicht der Geoinformation und kartographischen Kommunikation, zu einem ausgereifteren Produkt weiter zu entwickeln.

8 Zusammenfassung und Ausblick

Taktile Hypergloben bzw. die eingesetzte Globensoftware verwenden derzeit quadratische Platkarten, Rasterbilder mit einem Seitenverhältnis von 2:1 als Eingabemedium. Anhand dieser Diplomarbeit wurden bestehende Verfahren aufgegriffen sowie neue entwickelt und vergleichend untersucht, mit denen Vektordaten korrekt auf dem sphärischen Display angezeigt werden können. Punkt-, Linien- und Flächengeometrien können einerseits unter Berücksichtigung der Verzerrung der Basisrasterabbildung in diese eingezeichnet werden, andererseits können sie direkt in der Globensoftware verarbeitet und visualisiert werden.

Ein Globensystem setzt sich aus der Hardware, dem sphärischen Display, und der Software zusammen. Die Globensoftware generiert das Bild für die Darstellung am Globus. Weltweit gibt es mehrere Anbieter solcher Systeme. Dementsprechend gibt es unterschiedliche Softwarepakete für Globen. In dieser Diplomarbeit wurde die Globensoftware OmniSuite, die von der HRG an der Universität Wien entwickelt wird, besonders berücksichtigt.

Basierend auf einleitenden Ausführungen über taktile Hypergloben und sphärische Displays, geographische Vektordaten, bestehende Ansätze der Vektordatenintegration und der Funktionsweise von Earthbrowsern in Bezug auf die Darstellung von Vektordaten, wurde eine Klassifizierung möglicher Methoden und Verfahren entwickelt. In dieser Klassifikation werden die Implementierung der Visualisierung (intern oder extern in Bezug auf die Globensoftware), der Zeitpunkt der Bilderzeugung (synchron oder asynchron in Bezug auf die Laufzeit der Globensoftware), der Objektraum der Visualisierung (2D oder 3D) und die Ausgabe- bzw. Speicherformate der Visualisierung unterschieden.

Es wurden dreieinhalb mögliche Ansätze vorgestellt, wie Vektordaten für die Darstellung am sphärischen Display verarbeitet und aufbereitet sowie anschließend visualisiert werden können. Diese Verfahren beziehen sich u.a. stark auf die Funktionsweise von OmniSuit. Der erste interne Ansatz verwendet OpenGL ES Sahder um die Vektorgeometrien im Objektraum 2D direkt in die auszugebende Globenabbildung zu integrieren. Der zweite Ansatz lässt sich in ein internes und ein externes Verfahren unterteilen, beide verwenden die Grafik-Engine Ogre3D (die auch von OmniSuite eingesetzt wird). Diese ersten beiden Ansätze generieren die Darstellungsgeometrie durch die Bildung von Dreiecksnetzen (Meshes). Der dritte Ansatz (extern und asynchron) wurde mit dem Raytracer PovRay und dem

mathematischen Konzept der konstruktiven Festkörpergeometrie im Objektraum 3D umgesetzt. Hierbei wird durch eine sphärische Kamera eine Platkarte in beliebiger Auflösung generiert.

Unabhängig von dem Verfahren zählen zum Schritt der Datenverarbeitung und Datenaufbereitung das Einlesen der Geometrien, die Identifikation der einzelnen geographischen Ebenen bzw. eine Selektion der Features, die ev. benötigte Um-Projektion der Geometrien in geographische Koordinaten, eine maßstabsgebundene Generalisierung sowie eine verfahrensabhängige Segmentierung der Geometrien. Neben diesen technischen Anforderungen sind jedoch auch thematische und inhaltliche Anforderungen an Vektordaten zu stellen.

Bei der Datenvisualisierung spielt vor allem der Objektraum eine wichtige Rolle. Je nachdem, ob die Visualisierung im zwei- oder dreidimensionalen Raum vorgenommen wird, wird ein anderer Zeichenträger verwendet und es muss auf Verzerrungen und Nähte keine (3D) oder besondere (2D) Rücksicht genommen werden. Dabei werden unterschiedliche mathematische Grundlagen verwendet.

Nach der Erläuterung der technischen Umsetzungen wurden die einzelnen Verfahren vergleichend diskutiert. Aufgrund unterschiedlicher Faktoren (graphische Qualität und Erscheinungsbild, Unabhängigkeit vom Globensystem und Interaktivität, Rechen- und Speicheraufwand) und Eignungen (je nach Geometrietyp) der verschiedenen Verfahren, kann ein Schluss gezogen werden. Einerseits kann die Feststellung getroffen werden, dass sich je nach Projektzielen und Zeitaufwand die eine oder andere Methode besser eignet. Andererseits sind aus Sicht des Autors interne (somit synchrone) Lösungen, obwohl dadurch eine fixe Bindung an das Globensystem besteht und die Visualisierungen nicht anderweitig verwendet werden können, zu forcieren. Interne Lösungen, oft mit einem hohen Entwicklungsaufwand verbunden, bieten im Hinblick auf die Weiterentwicklung von Globensystemen und auf den wichtigen Faktor der Interaktivität viele Möglichkeiten. So kann bspw. auf Interaktionen, wie einer Selektion zur Laufzeit, bei touchfähigen Globen reagiert werden. Auch für (Lern)Spiele, bei denen der Spielverlauf auf Interaktionen basiert, könnten selektierbare (nicht nur geographische) Vektordaten eingesetzt werden.

Aus diesem Grund wird sich der Autor in nächster Zeit u.a. noch genauer mit den vorgestellten, internen Lösungsansätzen beschäftigen. Einerseits soll der Ansatz mit OpenGL ES Shadern weiterentwickelt werden, andererseits wäre eine Integration der Ansätze, die in Ogre3D implementiert wurden, in OmniSuite wünschenswert.

Darüber hinaus könnten im Zuge der Weiterentwicklung von Globensystemen und basierend auf den vorliegenden Ausarbeitungen folgende Themen bearbeitet werden:

- die Integration von geeigneten Generalisierungsalgorithmen in den Prozess
- die Wiedergabe von Realtime Vektordaten u.a. mittels Datenbankverbindungen (basierend auf der Arbeit von Kristen [KRI-12])
- die Entwicklung eines Signaturen-Baukastens für geometrische, bildhafte und symbolhafte Darstellungen für bestimmte Anwendungsfälle, vorzugsweise für ein internes Verfahren
- die korrekte Darstellung und Platzierung von Schrift und Beschriftungen entlang eines Großkreises bzw. entlang von Liniensignaturen unter der Berücksichtigung der Rotation
- die Wiedergabe animierter Signaturen basierend auf Vektordaten

All diese möglichen und geplanten Weiterentwicklungen sowie die Forschungsarbeit der HRG an der Universität Wien, die sich mit diversen, technischen und inhaltlichen Aspekten von Globensystemen auseinandersetzt und auch die Ergebnisse dieser Arbeit werden dazu beitragen, dass taktile Hypergloben Schritt für Schritt ihre kleine Nische verlassen und die Erde (die große Schwester/der große Bruder im Maßstab 1:1) erobern werden.

Quellenverzeichnis

- [ARC-12a] ARC Science Simulations: <http://www.arcsience.com/>, 24.01.2012
- [ARC-12b] ARC Science Geometer User Guide: <http://www.arcsience.com/Helpgeo.htm>, 24.01.2012
- [BAR-05] Bartelme N. (2005): Geoinformatik: Modelle, Strukturen, Funktionen. – 4. Aufl., Springer, Berlin u. Heidelberg, 2005, 466 S.
- [BEN-08] Benko H., Wilson A. u. Balakrishnan R. (2008): Sphere: multi-touch interactions on a spherical display. – Proceedings of the 21. annual ACM symposium on User interface software and technology, New York, 2008.
- [CIA-12] CIA – The World Factbook: <https://www.cia.gov/library/publications/the-world-factbook/index.html>, 23.08.2012
- [COZ-11] Cozzi P. u. Ring K. (2011): 3D Engine Design for Virtual Globes. – CRC Press, Taylor & Francis Group, Boca Raton, 2011, 499 S.
- [EBE-08] Eberly D. (2008): Triangulation by Ear Clipping. – <http://www.geometrictools.com/Documentation/TriangulationByEarClipping.pdf>, 24.09.2012
- [ESR-98] ESRI (1998): ESRI Shapefile Technical Description. – <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>, 28.09.2012
- [ESR-12a] ArcGIS Desktop 10 Online Help – How Buffer (Analysis) works: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/How_Buffer_Analysis_works/00080000001s000000/, 22.08.2012
- [ESR-12b] ArcGIS Desktop 10.1 Online Help – How Buffer (Analysis) works: http://resources.arcgis.com/en/help/main/10.1/index.html#/How_Buffer_Analysis_works/00080000001s000000/, 22.08.2012
- [FAN-10] Fang L. u. Hu B. (2010): Multi-scale Representation of Global Vector Data on Sphere Based on Map Accuracy. – Proceedings of the 3. International Symposium on Computer Science and Computational Technology (ISCST '10), Jiaozuo, China, 2010.
- [FIL-12] Filler A. (2012): 3D-Computergrafik und die Mathematik dahinter. <http://www.mathematik.hu-berlin.de/~filler/3D/>, 24.09.2012
- [FOO-08] Foody M. (2008): A New Tool to Help Build “High Production Value” Media for Spherical Displays. – NOAA's SOS Users Collaborative Network Workshop Report, Honolulu and Hilo, 2008. – http://www.oesd.noaa.gov/network/Workshop_rpt_2008/Presentations/day_3/Magic_Planet_day3.ppt, 26.01.2012
- [GIM-12] Global Imagination: <http://www.globalimagination.com/>, 24.01.2012
- [GLO-12a] Globocess AG: <http://www.globocess.com/>, 24.01.2011

- [GLO-12b] Themen Bibliothek der Globocess AG in Kooperation mit der HRG:
http://www.globocess.com/index.php?article_id=3, 24.01.2011
- [HAK-02] Hake G., Grünreich D. u. Meng L. (2002): Kartographie: Visualisierung raum-zeitlicher Informationen. – 8. Aufl., Walter de Gruyter, Berlin u. New York, 2002.
- [HRU-09] Hruby F., Miranda R. u. Riedl A. (2009): Bad Globes & Better Globes – multilingual categorization of cartographic concepts exemplified by “map” and “globe” in English, German and Spanish. – Proceedings of the 24. ICA International Cartographic Conference (ICC), Santiago, Chile, 2009.
- [IMH-72] Imhof E. (1972): Thematische Kartographie. – Walter de Gruyter, Berlin u. New York (=Lehrbuch der Allgemeinen Geographie, Band 10)
- [O3D-12] Object-Oriented Graphics Rendering Engine (Ogre3D):
<http://www.ogre3d.org/>, 11.09.2012
- [OGC-08] OGC, Wilson T. (Hrsg.) (2008): OGC® KML (Version 2.2.0). – [OGC-12]
- [OGC-10] OGC, Herring J.(Hrsg.) (2010): OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture (Version 1.2.1). – [OGC-12]
- [OGC-12] Open Geospatial Consortium (OGC) Standards:
<http://www.opengeospatial.org/standards>, 24.08.2012
- [OGR-12] OGR Simple Feature Library: <http://www.gdal.org/ogr>, 22.08.2012
- [ORO-08] O’Rourke J. (2008): Computational Geometry Column 51: Can a simple spherical polygon always be triangulated? – In: ACM SIGACT News, Vol. 40, No. 3., 2008. – <http://cs.smith.edu/~orourke/Papers/51.pdf>, 28.09.2012
- [KRI-12] Kristen J. (2012): Entwicklung von interaktiven kartographischen 3D-Applikationen am Beispiel eines taktilen Hyperglobus. – Diplomarbeit, Institut für Geographie und Regionalforschung, Universität Wien, 133 S.
- [LAM-11] Lampert A. (2011): Herausforderungen und Lösungsvarianten bei der Aufbereitung von Globenbildern als Basis für Texturen am taktilen Hyperglobus. – Diplomarbeit, Institut für Geographie und Regionalforschung, Universität Wien, 113 S.
- [LOH-12] PovRay online Tutorial von Lohmüller F.: http://www.f-lohmueller.de/pov_tut/pov_ger.htm, 18.09.2012
- [MUR-99] Murphy B., Collier P., Mitchell D. und Hirst W. (1999): Maritime Boundary Generation from Straight Baselines Defined as Geodesics. – Proceedings of the International Conference on Technical Aspects of Maritime Boundary Delineation and Delimitation, Monaco, 1999.
- [MUR-11] Murray G.(2011): Rotation About an Arbitrary Axis in 3 Dimensions. – <https://sites.google.com/site/glennmurray/Home/rotation-matrices-and-formulas/ArbitraryAxisRotation.pdf>, 19.09.2012
- [NED-12] Natural Earth Data: <http://www.naturalearthdata.com/>, 10.03.2012

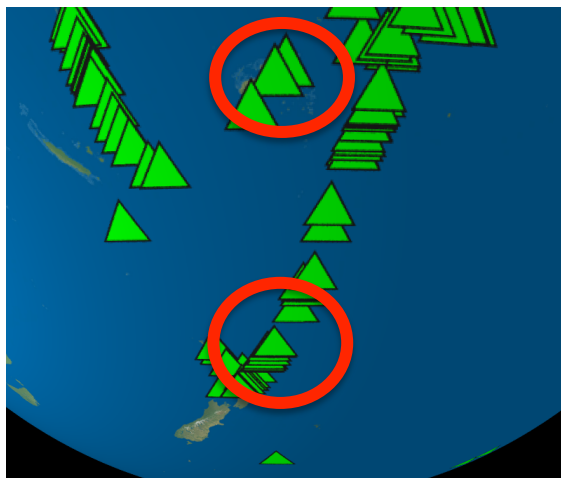
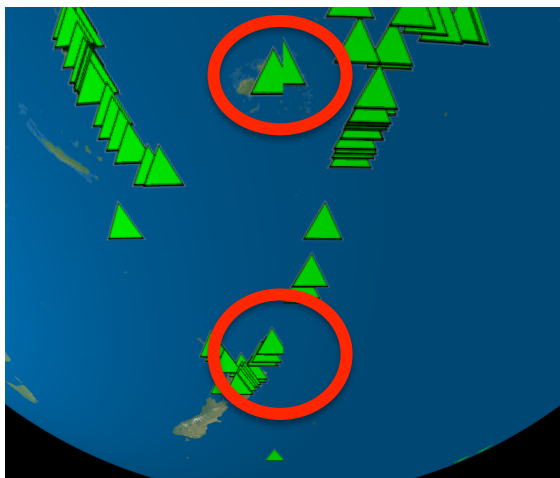
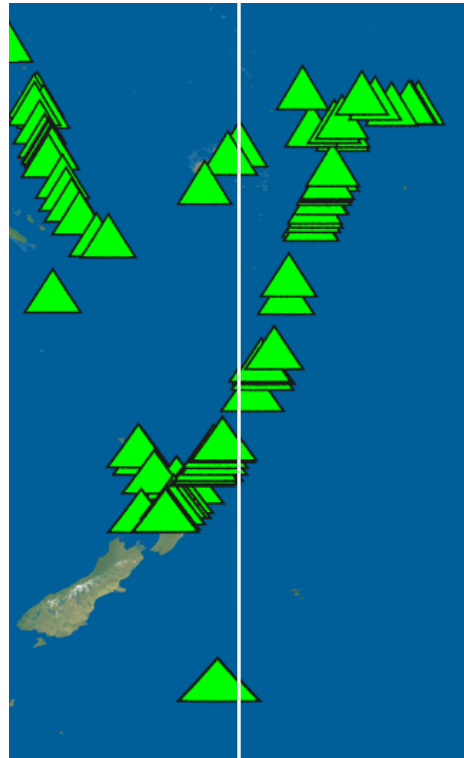
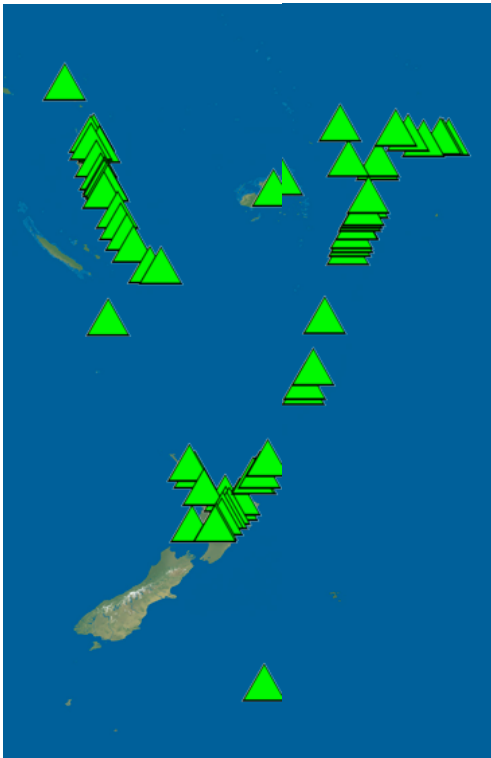
- [PAT-12] Patterson T. (2012): Natural Earth III - Data for Visualizing Earth from Space. – <http://www.shadedrelief.com/natural3>, 24.01.2012
- [POV-12] Persistence of Vision Raytracer (PovRay): <http://www.povray.org/>, 11.09.2012
- [RIE-00] Riedl A. (2000): Virtuelle Globen in der Geovisualisierung – Untersuchungen zum Einsatz von Multimediatechniken in der Geopräsentation. – Dissertation, Grund- und Integrativwissenschaftliche Fakultät der Universität Wien, 196 S.
- [RIE-08] Riedl A. (2008): Entwicklung und Perspektiven von taktilen Hypergloben. – In: Mitteilungen der Österreichischen Geographischen Gesellschaft, 150, S. 340-356.
- [RIE-10] Riedl A. u. Kristen J. (2010): Der Einsatz sphärischer Displays zur Visualisierung globaler Phänomene. – In: Kartographische Nachrichten, 60, 3, S. 129-137.
- [RIE-11a] Riedl A. (2011): Entwicklungsgeschichte digitaler Globen. – In: Der Globusfreund. Wissenschaftliche Zeitschrift für Globenkunde, 57/58, S. 153-166.
- [RIE-11b] Riedl A. (2011): Der Globus ist tot, es lebe der Globus! – In: Kriz K., Kainz W., Riedl A. (Hrsg.): 50 Jahre Österreichische Kartographische Kommission. Wien, Institut für Geographie der Universität Wien, 2011 (= Wiener Schriften zur Geographie und Kartographie, Band 20) S. 79-87.
- [SOS-12a] NOAA Science On a Sphere: <http://sos.noaa.gov/>, 24.09.2012
- [SOS-12b] Science On a Sphere (2012): User Manual. – <http://sos.noaa.gov/Docs/SOSUserManual.pdf>, 24.09.2012
- [SSG-12] Science On a Snow Globe Projekt: <http://eclecti.cc/>, 27.01.2012
- [TSA-11] Tsang C. (2011): Drawing Polylines by Tessellation. – <http://www.codeproject.com/Articles/226569/Drawing-polylines-by-tessellation>, 21.09.2012
- [VEN-12] Veness C. (2012): Skripte und Formeln für Berechnungen auf der sphärischen Erdoberfläche. – <http://www.movable-type.co.uk/scripts/latlong.html>, 18.09.2012
- [VMA-12a] Vector Map: <http://geoengine.nga.mil>, 24.08.2012
- [VMA-12b] Vector Map (VMAP0) im Shape Format: <http://gis-lab.info/qa/vmap0-eng.html>, 24.08.2012
- [VMA-12c] Vector Map auf der englischen bzw. deutschen. Wikipedia: [http://en\[de\].wikipedia.org/wiki/Vector_Map](http://en[de].wikipedia.org/wiki/Vector_Map), 24.08.2012
- [WIN-12] Wintner S. (2012): Digital Storytelling angewandt auf einen taktilen Hyperglobus. – Diplomarbeit, Institut für Geographie und Regionalforschung, Universität Wien, 159 S.

Anhang

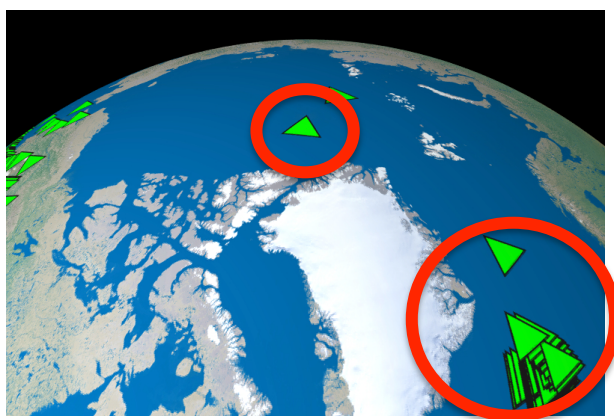
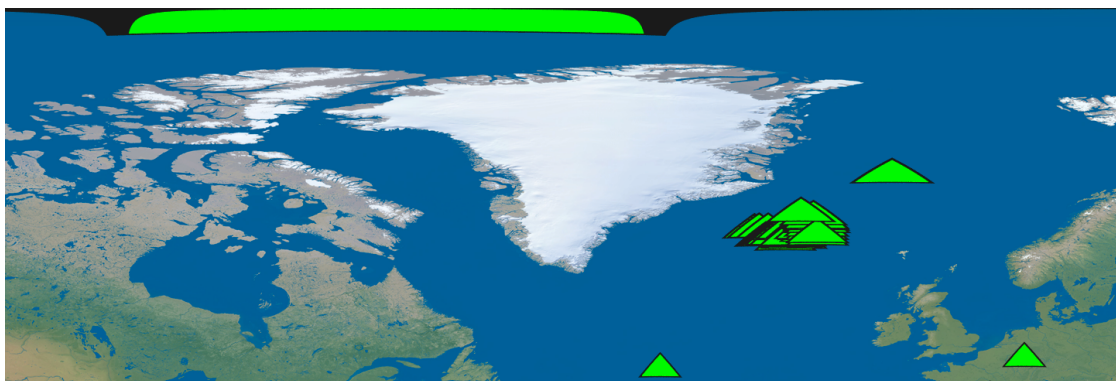
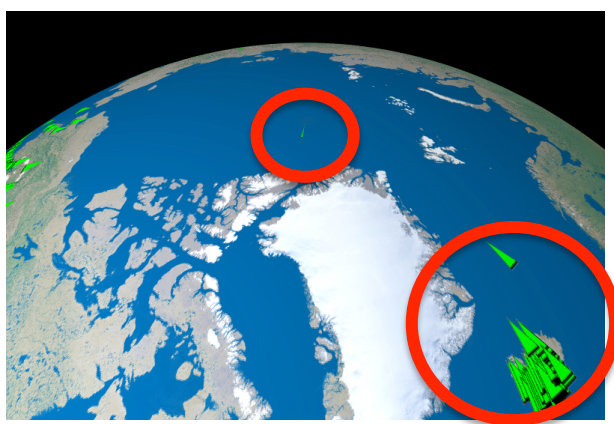
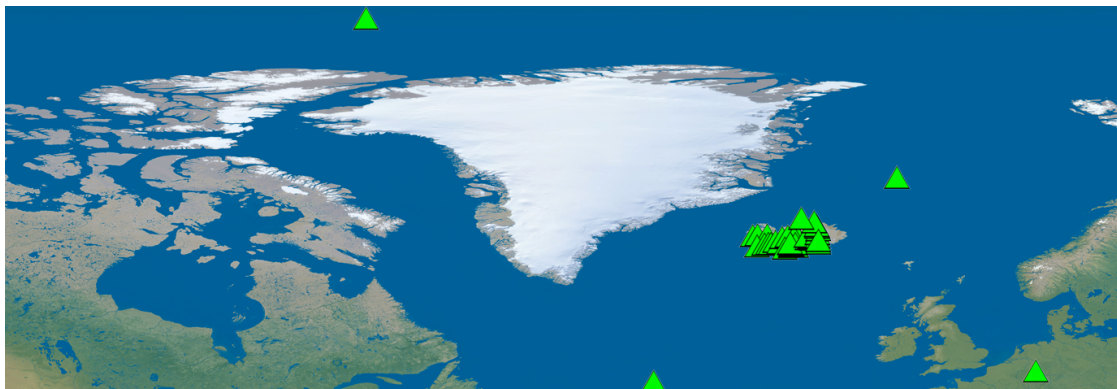
Anhang A: Vulkanstandorte

Hier wird ein „herkömmlich“ visualisierter Datensatz mit einem in PovRay generierten verglichen. Neben den Plattkarten werden in den Globenansichten die Problemstellen durch rote Kreise hervorgehoben.

Nähte (alt = links; neu = rechts)



Pole (alt = oben; neu = unten)

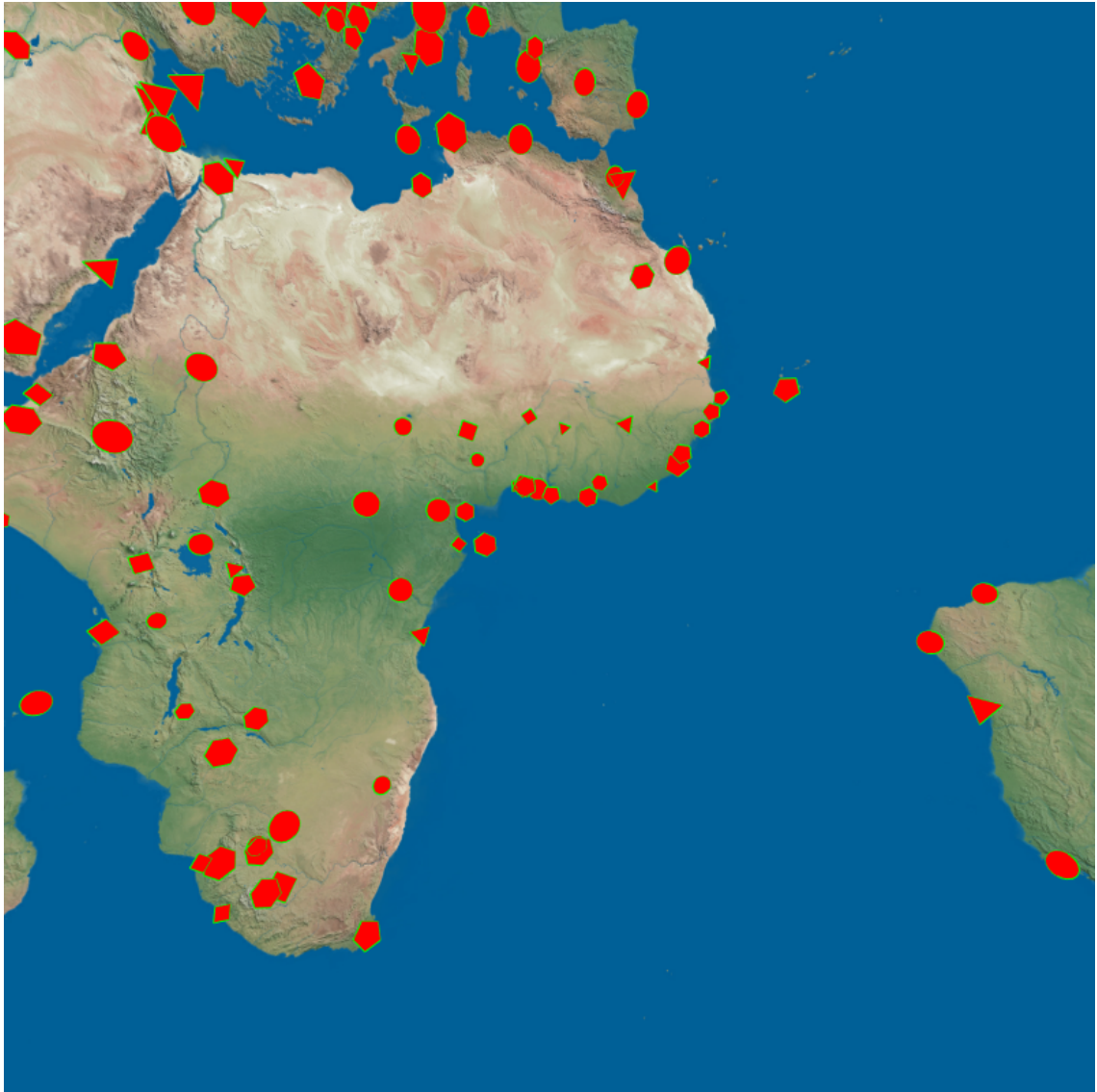


Anhang B: Städte

Visualisierung von Punktgeometrien (Städte von Natural Earth Data) in Ogre3D mit ManualObjects. Zufallswerte für Größe, Eckenanzahl und Symbolrotation als CubeMap.



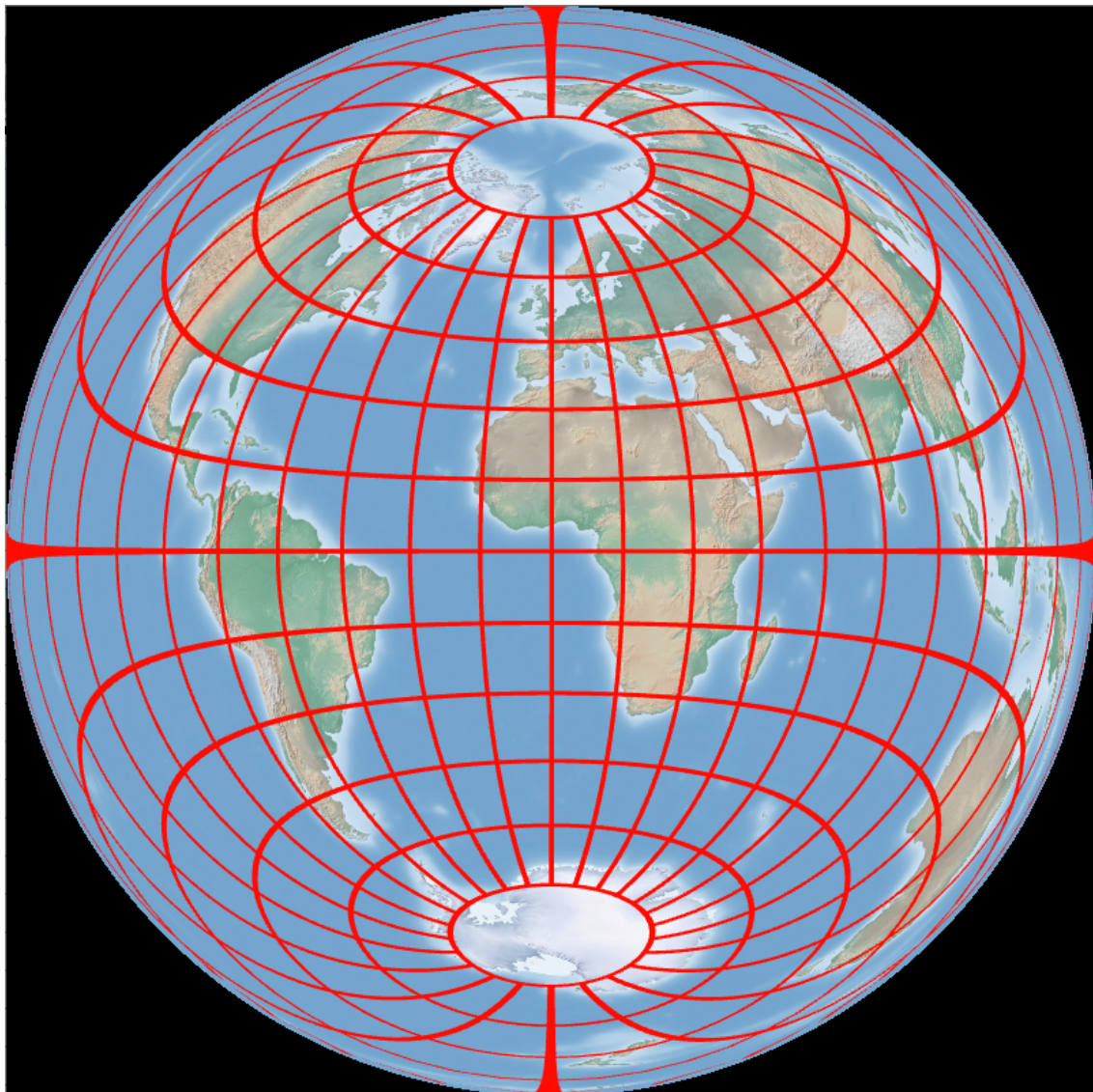
Detailansicht:



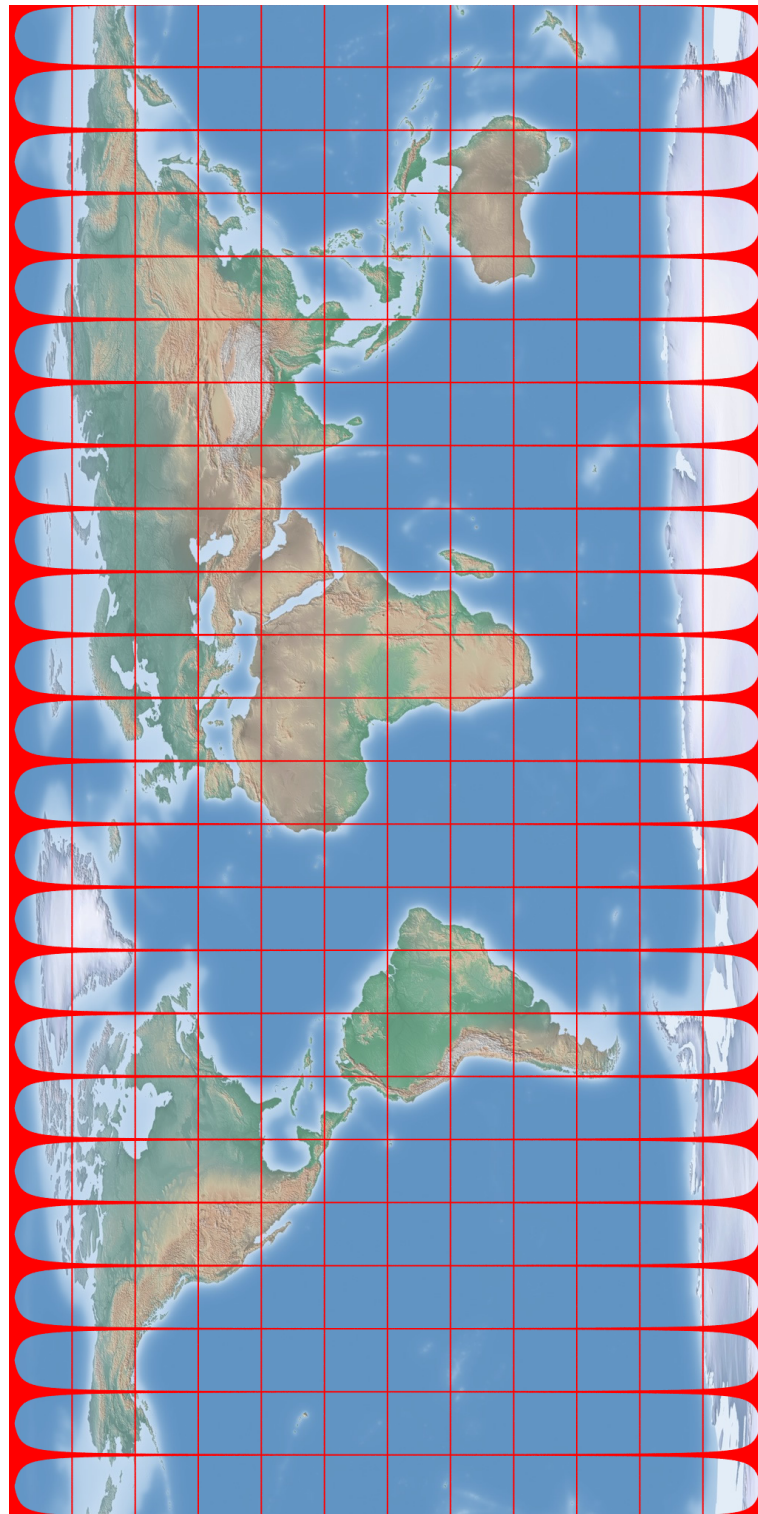
Anhang C: Gradnetz

Für das Gradnetz werden keine Vektordaten benötigt, es kann programmtechnisch mit verschachtelten Schleifen generiert werden. Die Abstände zwischen den Breiten- bzw. Längengraden können unterschiedliche Gradwerte haben. In beiden Fällen handelt es sich um ein 15°-Gradnetz, wobei bei dem ersten Beispiel die Längengrade von 75° Nord nach 75° Süd verlaufen. So werden bspw. Ereignisse am Pol nicht überdeckt.

OpenGL ES Shader:



PovRay:



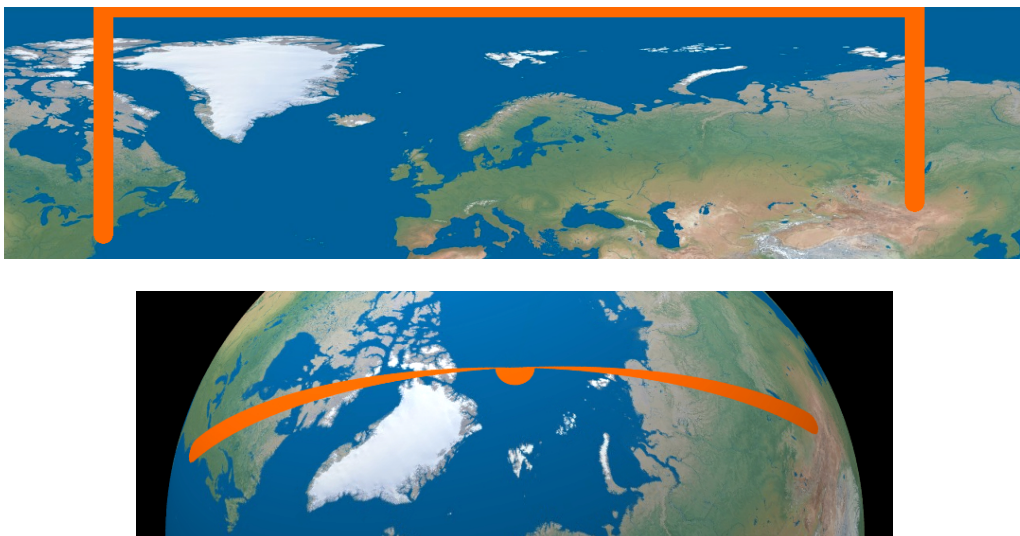
Anhang D: Flugroute über Pol

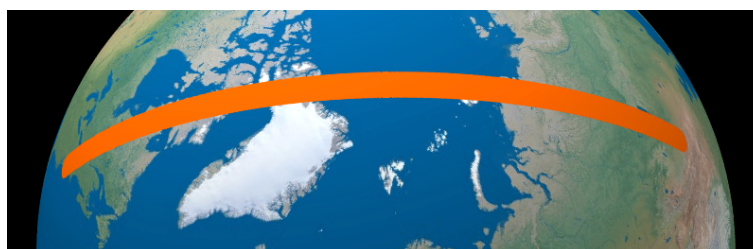
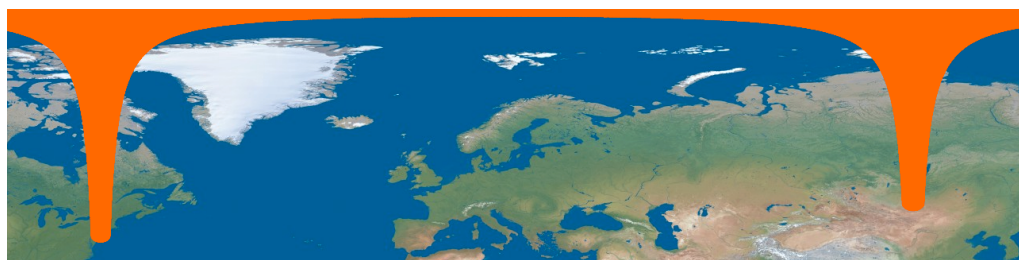
Dieses Beispiel zeigt auf, dass für eine korrekte Darstellung am Globus es einerseits wichtig ist, dass Linien entlang der Orthodrome eingezeichnet werden und andererseits, dass auch die Verzerrungen berücksichtigt werden müssen. Bei den Beispielen ist jeweils ein Ausschnitt aus der Plattkarte sowie aus der Globenansicht dargestellt.

Lineare Verbindung (ArcGIS):



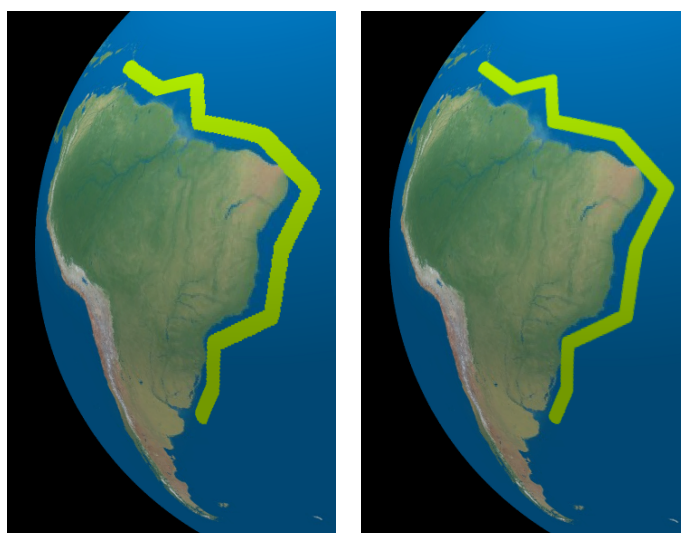
Geodätische Linie/Orthodrome (ArcGIS):



Geodätische Linie/Orthodrome (PovRay):**Anhang E: Schiffstrack**

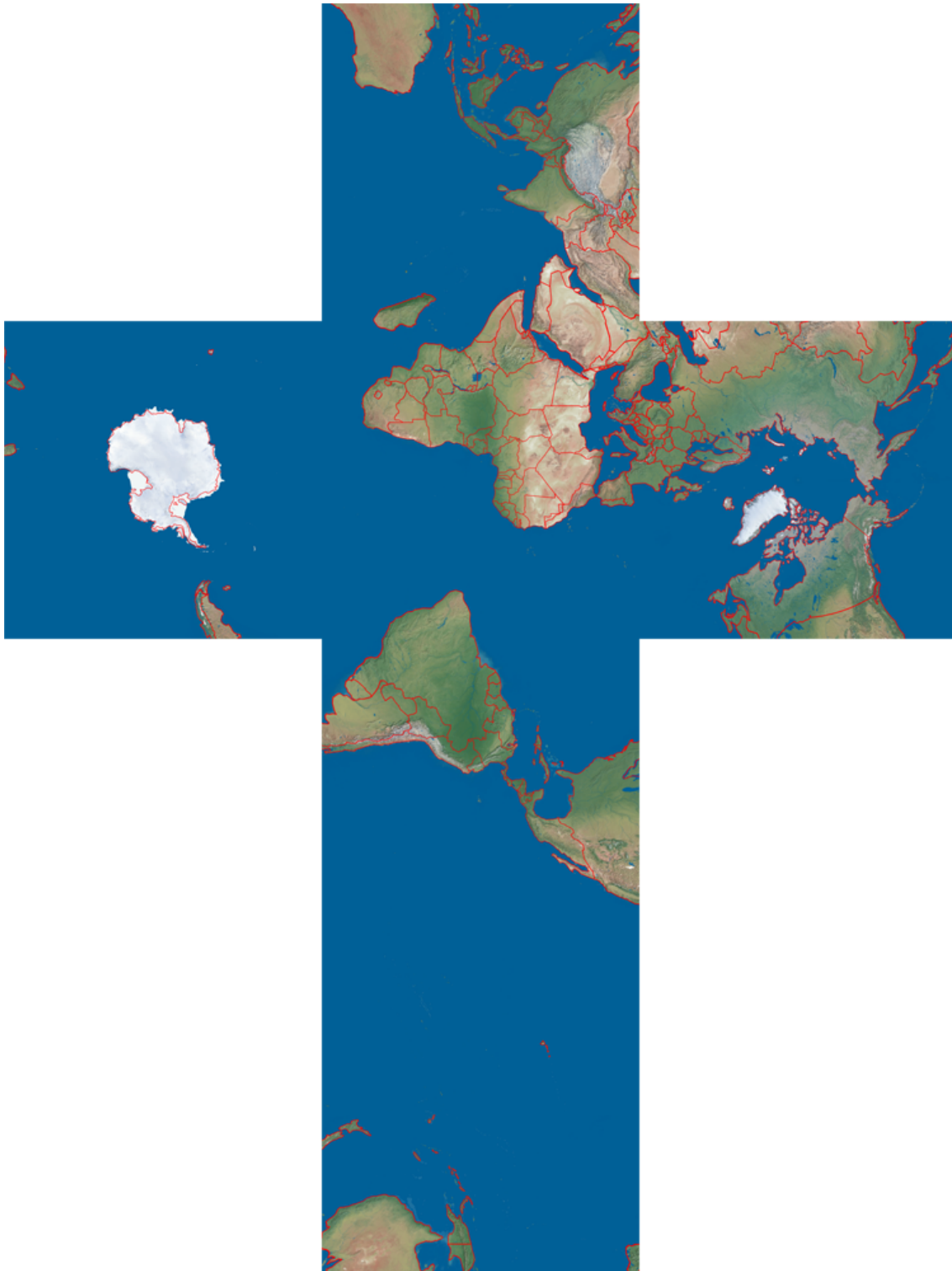
Dieses Beispiel zeigt einen Schiffstrack entlang der südamerikanischen Ostküste. Links wurde eine Plattkarte verwendet, die mit ArcGIS und ohne Berücksichtigung der Verzerrungen erstellt wurde. Die Abbildung rechts wurde mit dem PovRay-Ansatz, bei dem Verzerrungen berücksichtigt werden, erstellt.

Das Beispiel zeigt auf, dass bei der Visualisierung von Geometrien in weniger hohen Breiten, da die Verzerrungen gering sind, auch mit einfachen Methoden (ArcGIS) gute Ergebnisse erzielt werden können.

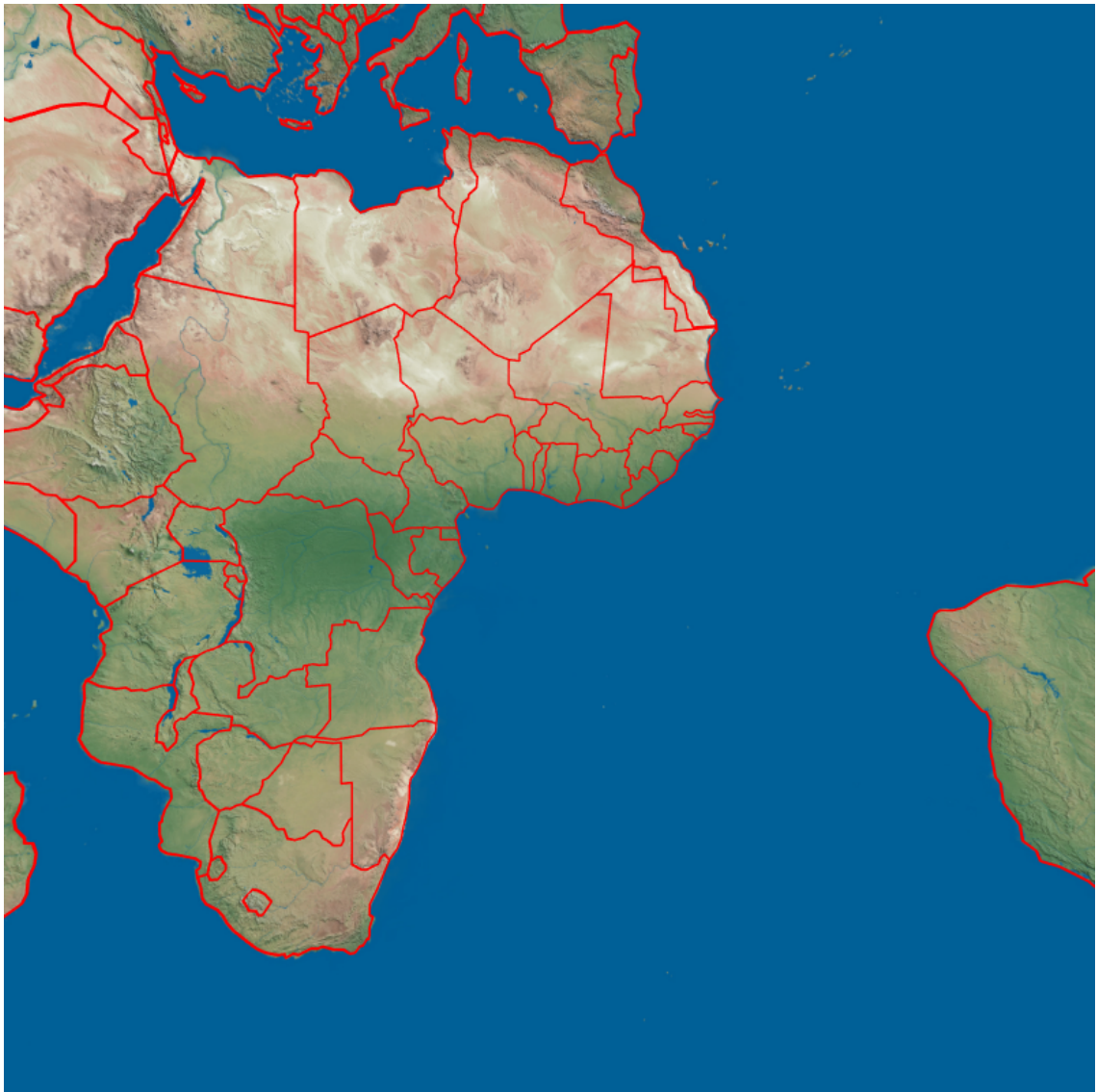


Anhang F: Staatsgrenzen

In Ogre3D mit ManualObjects visualisierte Staatsgrenzen (Natural Earth Data) als CubeMap ausgegeben.

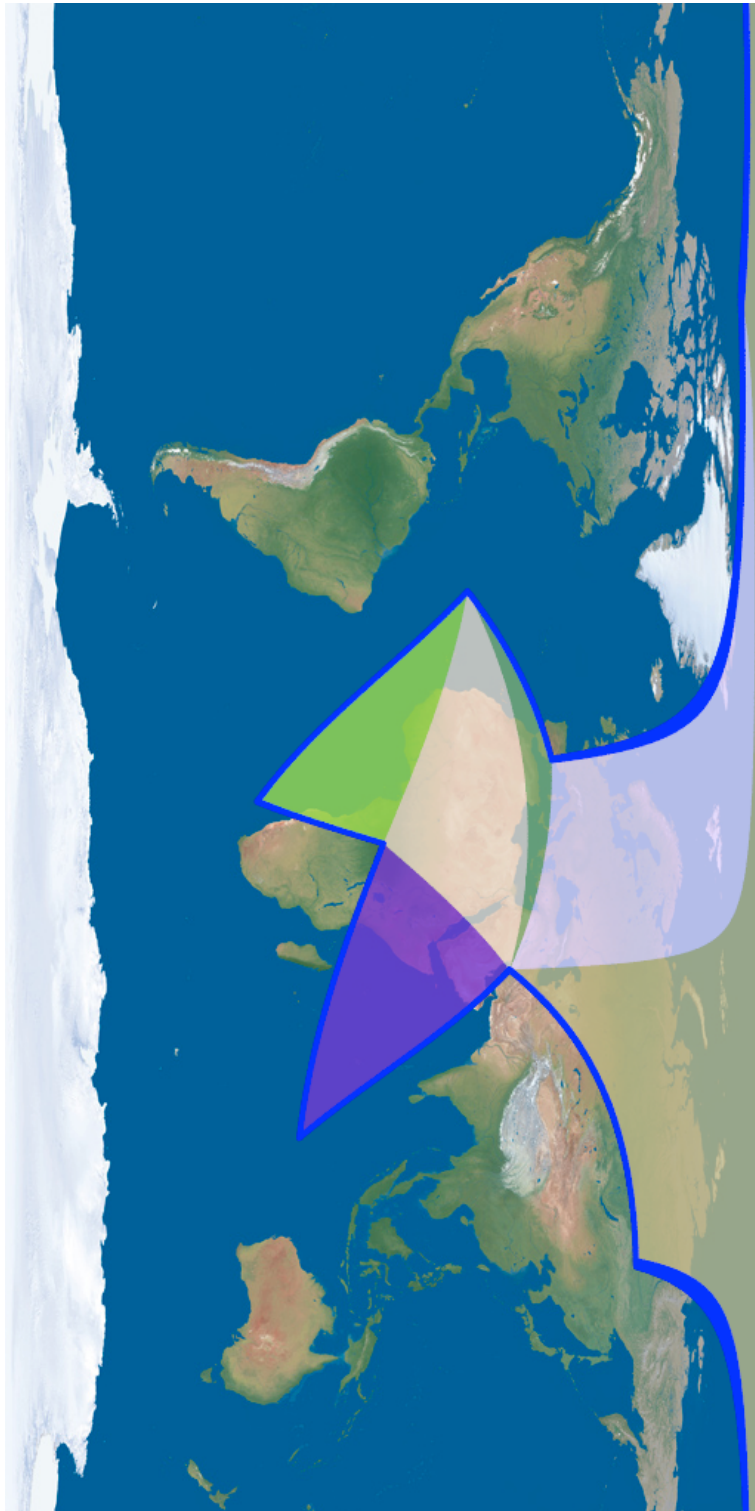


Detailansicht:



Anhang G: Polygontriangulierung mit PovRay

Das Ergebnis der Triangulierung in Kapitel 6.3.2 auf der Plattkarte.



Anhang H: Kurzfassung

Ein taktiler Hyperglobus ist eine moderne, digitale Präsentationsform, mit der unterschiedlichste globale Zusammenhänge und Phänomene wiedergegeben werden können. Globensysteme verwenden gegenwärtig quadratische Plattkarten (Raster) als Eingabemedium.

Im Zuge dieser Diplomarbeit werden bestehende Verfahren, die es erlauben Vektordaten korrekt auf dem sphärischen Display anzuzeigen, aufgegriffen sowie neue entwickelt und vergleichend untersucht. Punkt-, Linien- und Flächengeometrien können einerseits unter Berücksichtigung der Verzerrungen und Nähte (z.B. Datumsgrenze) in den Basisraster eingezeichnet werden, andererseits können sie direkt in der Globensoftware verarbeitet und visualisiert werden.

Basierend auf Ausführungen über taktile Hypergloben und sphärische Displays, über geographische Vektordaten sowie über bestehende Ansätze der Vektordatenintegration und der Funktionsweise von Earthbrowsern, wird eine Klassifizierung möglicher Methoden und Verfahren vorgestellt.

Es werden mehrere Verfahren in Theorie und Praxis besprochen und mit Beispielen belegt. Neben der Datenverarbeitung und -aufbereitung sowie der Datenvisualisierung werden Anforderungen, die an die Vektordaten zu stellen sind, erörtert. Abschließend werden die unterschiedlichen Methoden verglichen und bewertet.

Die Diplomarbeit reiht sich in die Forschungsarbeit der Hyperglobe Research Group (HRG) des Instituts für Geographie und Regionalforschung der Universität Wien mit ein.

Anhang I: Abstract

A tactile hyperglobe is a new digital presentation tool, which facilitates graphic display of a wide variety of phenomena and facts about the planets. These globe systems use a plate carrée projection (grid) as an input.

Existing methods for projection of vector data on a spherical display are gathered and new methods are developed and compared. These procedures handle point, line, and polygon geometries, which can be pre-rendered directly into the base grid, accounting for distortions and seams (eg. dateline), or they can be processed and displayed during runtime of the globe software.

Based on the discussion of: tactile hyperglobes and spherical displays; geographic vector data and existing approaches for the vector data integration; and the principles of earthbrowsers a classification of possible methods and procedures is introduced.

The theory and practice of several methods are presented and explained with examples. Data processing and preparation, data visualization and requirements for the vector data are discussed. In the conclusion, these different methods are compared and evaluated.

The thesis is part of the work of the Hyperglobe Research Group (HRG) of the Department of Geography and Regional Research (University of Vienna).

Anhang J: Curriculum Vitae

Persönliche Daten

<i>Name</i>	Johannes Liem
<i>Geburtsdatum</i>	12.12.1985
<i>Geburtsort</i>	Luzern (CH)
<i>Staatsbürgerschaft</i>	Österreich, Schweiz
<i>Adresse</i>	Söllheim 1, A-5300 Hallwang
<i>Telefon</i>	0043 699 10082801
<i>Email</i>	johannes.liem@gmx.at

Ausbildung

Seit 10/2006	Universität Wien, Institut für Geographie und Regionalforschung; Diplomstudium Geographie (1. Abschnitt; beendet: 01/2009); Diplomstudium Kartographie und Geoinformation (2. Abschnitt)
10/2005 – 09/2006	Universität Wien, Fakultät für Informatik Bachelorstudium Informatik/Medieninformatik
09/2000 – 07/2004	BORG Nonntal, Salzburg; Schwerpunkt Informatik; Matura
09/1992 – 07/2000	Volks- und Hauptschule Mattsee, Salzburg

Berufserfahrung

Seit 10/2009	Mitglied der Hyperglobe Research Group (HRG) an der Universität Wien, Institut für Geographie und Regionalforschung, Arbeitsgruppe Kartographie und Geoinformation
Seit 02/2009	Web- und IT-Projekte für • Globoccess AG • Geologic Dr. Benedikt • Österr. Verband für Angewandte Geographie
04/2012 – 07/2012	Praktikum Cartography and Geovisualization Group, Oregon State University (USA)
03/2009 – 07/2012	Tätigkeiten als Tutor/Assistent: • Angewandte Geoinformation (Uni Wien) • Multimediatechnologien in der Geoinformation (Uni Wien) • Mobiles GIS/GPS (Uni Wien) • Web Mapping (Oregon State University)

10/2010 – 11/2010 Praktikum Globoccess AG (Deutschland)

10/2004 – 09/2005 Zivildienst bei der Lebenshilfe Salzburg

Wissenschaftliche Interessen

- Digitale (physische) Globen
- Programmierung und Multimediatechniken in der Kartographie und Geoinformationswissenschaft
- Automatisierung im kartographischen Prozess
- Vektordaten und Web Mapping

IT Kenntnisse

Programmierung: PHP, MySQL, HTML 5, JavaScript, WebGL, CSS, XML, Action Script, C++, Python, Delphi (Pascal), Basiswissen über wichtige Algorithmen, Datenstrukturen und (Geo-)Datenbanksysteme

Design/3D/Kartographie: Adobe Creative Suite, Cinema 4D, Bryce, PovRay, Bender, Ogre3D, GeoServer, OpenLayers, Google Maps API, Mapnik, Mapyrus, Terrain Bender, Flex Projector

GIS: ArcGIS, QuantumGIS, GDAL/OGR, Geodatenbanken, Basiswissen über OGC-Standards und Datenformate der Geoinformation

Sprachen

Deutsch (Muttersprache)
Englisch (fließend in Wort und Schrift)
Italienisch (Maturaniveau)

Leistungsstipendien der Universität Wien

Akademisches Jahr 2010 (Diplom Kartographie und Geoinformation)
Akademisches Jahr 2009 (Diplom Geographie)
Akademisches Jahr 2006 (Bachelor Medieninformatik)

Referenzen

Andreas Riedl (Betreuer)
Universität Wien – Institut für Geographie und Regionalforschung
Universitätsstraße 7
1010 Wien (Österreich)
andreas.riedl@univie.ac.at / 0043 1 4277 48642

Bernhard Jenny
Oregon State University – Cartography and Geovisualization Group
104 CEOAS Admin Bldg
Corvallis, OR 97331 (USA)
jennyb@geo.oregonstate.edu / 01 541 737 1204

Anhang K: Eidesstattliche Erklärung

Ich versichere:

- dass ich die Diplomarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- dass ich dieses Diplomarbeitsthema 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