

Masterarbeit

TITEL DER MASTERARBEIT

**„Web3D und dessen Potenzial zur
Visualisierung geologischer Daten
—
gezeigt am Beispiel der Stadt Wien“**

Verfasser

Frederik VON REUMONT

Angestrebter akademischer Grad
Master of Science, MSc

Wien, im November 2010
Studienkennzahl lt. Studienblatt:
Studienrichtung lt. Studienblatt:

Betreuer:

A 066 856
Masterstudium Kartographie und
Geoinformation
Ass.-Prof. Mag. Dr. Andreas Riedl

Für meine Eltern

Danksagung

Mein Dank gilt der Wiener Gewässermanagement Gesellschaft mbH und ihrem Geschäftsführer Ing. Martin Jank, sowie der Leiterin des Technischen Büros – Ingenieurbüro für Geologie Dr. Sabine Grupe und Mag. Thomas „Major Tom“ Payer für die Unterstützung in allen Fragen, ein offenes Ohr für alle Probleme und nicht zuletzt für die finanzielle Unterstützung dieser Masterarbeit.

Außerdem gebührt mein Dank Ass.-Prof. Mag. Dr. Andreas Riedl für die wissenschaftliche Betreuung dieser Arbeit und die sehr hilfreiche Unterstützung.

Mein persönlicher Dank geht an meine Studienkollegen, die mich stets mit allen notwendigen Informationen versorgt haben, allen voran Marko Einspieler, Joachim Ungar, Jamal Jokar und alle anderen. Ohne sie wäre ich nicht so weit gekommen. Ich danke außerdem meinem Bruder Till für die Maus. Und meiner Freundin Lara für ihre Geduld.

Inhaltsverzeichnis

Verwendete Abkürzungen	iii
Verzeichnis der Abbildungen	iv
Verzeichnis der Listings	viii
Verzeichnis der Tabellen	ix
Abstract	x
Zusammenfassung	x
Einleitung	1
1 Die dritte Dimension in der Kartographie	8
<i>1.1 Aktuelle Forschungsgebiete</i>	<i>9</i>
1.1.1 Multiperspektivische Ansichten	9
1.1.2 LiDAR	10
1.1.3 Statistische 3D-Oberflächen	12
<i>1.2 Sonderfall Geologie</i>	<i>13</i>
2 Modellierung	16
<i>2.1 Repräsentationsformen von 3D-Objekten</i>	<i>16</i>
<i>2.2 Topologische Probleme</i>	<i>28</i>
<i>2.3 Stand der Technik: wissenschaftliche 3D Software in der Geologie</i>	<i>32</i>
2.2.1 Plug-Ins für GIS-Programme	33
2.2.2 Eigenständige Softwarelösungen zur 3D-Modellierung geologischer Formen	37
3 Visualisierungsmöglichkeiten mit Web-Technologien	44
<i>3.1 Geo-Browser und virtuelle Globen</i>	<i>47</i>
<i>3.2 OGC Web Services</i>	<i>56</i>
<i>3.3 3D-Datenformate im Netz</i>	<i>60</i>
3.3.1 Universal 3D/3D-PDF	61
3.3.2 O3D/WebGL	62
3.3.3 Rich Internet Applications mit Adobe Flex, JavaFX, MS Silverlight	62
3.3.4 eXtensible 3D	66
<i>3.4 Beispiele aus der Wissenschaft</i>	<i>69</i>
4 Anforderungen an die Visualisierung	78
5 Datengrundlage	81
6 Die qualifizierte Entscheidung für X3D	84

<i>6.1 Gute Rahmenbedingungen</i>	84
6.1.1 Interoperabilität	84
6.1.2 Offenheit des Standards	84
6.1.3 Breite Akzeptanz	85
6.1.4 Verschlüsselung und Kompression	86
6.1.5 Erfahrung aus dem wissenschaftlichen Bereich	87
6.1.6 Einwände	87
<i>6.2 Praktische Gründe</i>	87
6.2.1 Vielfältigkeit	87
6.2.2 Unterstützung von Geodaten	89
7 Der Prozess der Visualisierung in X3D	90
<i>7.1 Erstellung der Schichtenvisualisierung</i>	90
<i>7.2 Erstellung des Orthofoto-Overlays</i>	98
<i>7.3 Visualisierung der Shape-Dateien</i>	102
<i>7.4 Zusammenführung der Einzelteile und Ausstattung mit Funktionen</i>	105
7.4.1 Interaktive Legende	111
7.4.2 Überhöhung	111
7.4.3 Explodieren	113
7.4.4 Profilschnitte	115
7.4.5 Ausgabe der Höhenwerte	117
<i>7.5 Das Design der graphischen Benutzeroberfläche</i>	118
<i>7.6 Das Problem des vertikalen Maßstabs</i>	121
8 Bewertung	123
9 Ausblick	129
10 Fazit	132
Literatur	133
Anhang	xii

Verwendete Abkürzungen

ACM	Association for Computing Machinery
API	Application Programming Interface
AQE	Augemnted Quad Edge
ASCII	American Standard Code for Information Interchange
BGS	British Geological Survey
B-Rep	Boundary Representation
CSG	Constructive Solid Geometry
DLR	Deutsches Zentrum für Luft- und Raumfahrt
Ecma	European association for standardizing information and communication systems
Esri	Envrionmental Systems Research Institute Inc.
GA	Geoscience Australia
GLOBE	Global Learning and Observation to Benefit the Earth
GS13D	Geological Surveyin and Investigation in 3 Dimensions
GUI	Graphical User Interface (graphische Benutzeroberfläche)
HTML	Hyper Text Markup Language
HUD	Heads-Up-Display
ISO	International Organization for Standardization
KML	Keyhole Markup Language
LAS	Log ASCII-Standard
LGPL	Lesser General Public License
LOD	Level of Detail
MGI	Militärgeographisches Institut
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NURBS	Non-uniform Rational B-Splines
OGC	Open Geospatial Consortium
PDF	Portable Document Format
SDK	Software Development Kit
SEG Y	Society of Exploration Geophysicists Datenformat
SIGGRAPH	Special Interest Group on Computer Graphics and Interactive Techniques
SRTM	Shuttle Radar Topography Mission
TIN	Triangulated Irregular Network
RIA	Rich Internet Application
U3D	Universal 3D
VRML	Virtual Reality Markup Language
W3DS	Web 3D Service
WGM	Wiener Gewässer Management Ges. mbH
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPVS	Web Perspective View Service
WTS	Web Terrain Service
X3D	eXtensible 3D Graphics
XAML	eXtensible Application Markup Language
XML	eXtensible Markup Language

Verzeichnis der Abbildungen

- Abbildung 1: Der Stephansdom in Wien als Punktwolke. Quelle: www.riegl.com** 12
- Abbildung 2: Die Bulandpreise Deutschlands als dreidimensionale statistische Oberflächen, links als extrudierte Prismen, rechts als Kontinuum. Quelle: Bundesinstitut für Bau-, Stadt- und Raumforschung 2009** 13
- Abbildung 3: Spatial Occupancy Enumeration bietet oft nur grobe Annäherungsmöglichkeiten an glatte, gebogene Oberflächen. Quellen: http://www.effectware.com/download/images/efx_voxel2.jpg, <http://www.one-largeprawn.co.za/2009/12/22/awesome-voxel-art-by-metin-seven/>** 17
- Abbildung 4: Delauney Tetraeder. Quelle: Alliez, P. et al. (2010) http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Mesh_3/Chapter_main.html** 18
- Abbildung 5: Der binäre Baum einer Constructive Solid Geometry (CSG). Quelle: http://de.academic.ru/pictures/dewiki/67/Csg_tree.png** 19
- Abbildung 6: Die Zerlegung eines 3D-Körpers in seine Einzelteile: Flächen, Kanten und Punkte. Quelle: http://www.ielm.ust.hk/dfaculty/ajay/courses/ieem317/lecs/3dcad/boundary_s.gif** 20
- Abbildung 7: Zwei gleich ausgerichtete Polygone, die Teil einer einzigen Oberfläche sind. Die im Uhrzeigersinn weisenden Pfeile zeigen die Ausrichtungen der Kanten (Rand) des Polygons an. Quelle: Eigene Bearbeitung nach OGC 2006** 21
- Abbildung 8: Winged Edge Struktur nach Greiner. Quelle: Greiner 2000** 23
- Abbildung 9: Quad Edge Struktur. Quelle: Greiner 2000** 24
- Abbildung 10: Veranschaulichung der Extrusionsoperationen, bei denen eine Grundfläche (schwarz dargestellt) durch den 3D-Raum „gezogen“ wird. Quelle: http://www.sd1.co.uk/images/portfolio/print/extrusion_concepts.jpg** 25
- Abbildung 11: Mit NURBS können im 3D-Raum komplexe Formen modelliert werden. Kontrollpunkte (hier nicht abgebildet) bestimmen die Krümmung**
-

der Kurven. Quelle: http://www.3dmax-tutorials.com/graphics/il_nurbs_sweep2.jpg	26
Abbildung 12: Die Benutzeroberfläche in GeoRover. Quelle: Gocke et al. 2010	37
Abbildung 13: 3D-Darstellung von Salzdiapieren in Gocad. Quelle: http://www.pdgm.com/products/seismic-processing-imaging/gocad-velocity-modeling.aspx	39
Abbildung 14: Beispiele für Modelle aus GSI3D. Quelle: http://www.gsi3d.org/index.html	41
Abbildung 15: Voxel-Darstellung eines Niederschlags-Voxelvolumenkörpers aus GRASS. Quelle: http://grass.fbk.eu/images/slovakiavol.jpg	43
Abbildung 16: World Wind Screenshot vom Berg Fuji, Japan mit SRTM-Daten. Quelle: http://worldwind.arc.nasa.gov/screenshots-sl.html	51
Abbildung 17: Die Gratisversion von Google Earth. Der Screenshot zeigt eine Ansicht von Wien. Quelle: Google Earth	53
Abbildung 18: Die verschiedenen Client/Server-Konzepte und entsprechende OGC-Webservices. Quelle: OGC 2005	60
Abbildung 19: Beispiel einer RIA. Im Browserfenster ist ein GUI für das CIA Factbook geöffnet. Es sind interaktive Legenden, Charts und 3D-Graphiken zu erkennen. Quelle: http://visunetdemos.demos.ibm.com/blogsamples/factbook2/FactBookSE.html	63
Abbildung 20: Kartographische RIA der Firma John L. Scott Real Estate als Mash-Up mit bing-maps. Quelle: www.johnlscott.com	66
Abbildung 21: Das Ergebnis aus Listing 1, wie es von einem X3D-Player interpretiert wird: ein gelber Zylinder. Quelle: Eigene Abbildung	68
Abbildung 22: Ein online verfügbares 3D-Modell von GA, hier die Region Gawler Craton. Verschiedene geologische Informationen werden im Plug-in visualisiert, während die Steuerelemente als JavaScript-Schaltflächen im HTML-Bereich der Webseite definiert werden. Quelle: http://www.ga.gov.au/resources/multimedia/3dmodels.jsp	73
Abbildung 23: Der Hype Zyklus, entwickelt von der Beraterfirma Gartner. Quelle: http://www.gartner.com/pages/story.php.id.8795.s.8.jsp	86

- Abbildung 24: Der Aufbau des Gitternetzes des in Listing 1 beschriebenen Zylinders. Das Geometrische Primitiv wird mit Hilfe von B-reps aufgebaut, die das Innere des Objekts "leer" lassen. Quelle: Eigene Abbildung 88**
- Abbildung 25: In der Netzstruktur sind die Rasterpunkte als Knoten zwischen den Kanten zu erkennen. Trotzdem verwendet X3D Dreiecke zur Beschreibung der Geometrie. Es handelt sich also in Wirklichkeit um ein simuliertes Raster. Quelle: Eigene Abbildung 91**
- Abbildung 26: Deutlich sichtbar treten Artefakte ("Eiszapfen") an den steilen Übergängen von definierten Gebiet und nicht-definiertem. Quelle: Eigenen Abbildung 92**
- Abbildung 27: In der Gitternetz-Ansicht sind die Probleme an den Eckpunkten der definierten Oberflächen deutlich zu sehen. Quelle: Eigene Abbildung 93**
- Abbildung 28: Ein Beispiel für eine Grafikdatei, die als Textur verwendet wurde. Die hier weiß erscheinenden Teile sind transparent. Quelle: Eigene Abbildung 94**
- Abbildung 29: Bei extremer Überhöhung (hier: zehnfach) werden in den Schichten des Miozäns und des DGM noch vereinzelt Artefakte sichtbar. Quelle: Eigene Abbildung 96**
- Abbildung 30: Die Ersellung der Kacheln in ArcMap. Quelle: Eigene Abbildung 99**
- Abbildung 31: Die Lücken (rot markiert) zwischen den einzelnen Kacheln des von ArcScene ins VRML-Format exportierten Orthofoto-Overlays verändern sich je nach Zoomstufe/LOD. Quelle: Eigene Abbildung 102**
- Abbildung 32: Der Workflow von der Quelle bis zum fertigen Bestandteil des Gesamtmodells. WWS: Wienerwaldschotter, IK: Interkolluvium, JDS: Junger Donauschotter, GW: Grundwasser, MIO: Miozänoberkante, DGM: Digitales Geländemodell. Quelle: Eigene Abbildung 104**
- Abbildung 33: Die Dateistruktur zur Darstellung der Schichten. Die Hauptdatei verwendet Prototypen mit anderen LOD als die Kinddateien. Zwischen allen Dateien kann hin- und hergewechselt werden. Quelle: Eigene Abbildung 108**
- Abbildung 34: Die Struktur des gesamten Dateiensystems für die Darstellung aller Daten. Sowohl aus der Hauptdatei als auch aus allen Kinddateien**
-

können die Zusatzinformationen hinzugeschaltet werden, weshalb erstere im gestrichelten Rahmen zusammengefasst sind. Quelle: Eigene Abbildung	110
Abbildung 35: Schematische Darstellung der Funktion "Überhöhung". Quelle: Eigene Abbildung	113
Abbildung 36: Schematische Darstellung der Funktion "Explodieren". Quelle: Eigene Abbildung	115
Abbildung 37: Schematische Darstellung der Funktion "Profilschnitte". Quelle: Eigene Abbildung	117
Abbildung 38: Schematische Darstellung der Funktion "Höhenwert". Quelle: Eigene Abbildung	118
Abbildung 39: Das HUD der Steuerungsfunktionen. Rechts die Variante der Hauptdatei, links die der Kinddateien. Quelle: Eigene Abbildung	119
Abbildung 40: Die Elemente zum Ein- und Ausschalten von Zusatzinformationen. Quelle: Eigene Abbildung	120
Abbildung 41: Das so genannte z-Fighting verzerrt die Polygongrenzen. Der Effekt fällt besonders bei Kamerabewegungen auf. Quelle: Eigene Abbildung	126
Abbildung 42: Statistik über die Verbreitung von Internetbrowser nach Marke. Quelle: http://gs.statcounter.com/	130

Ich habe mich bemüht, sämtliche Inhaber der Bildrechte ausfindig zu machen und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir.

Verzeichnis der Listings

Listing 1: Aufbau einer einfachen Szene in X3D. Die Farben zur besseren Kennzeichnung der Knoten und Felder basieren auf X3D-Edit	67
Listing 2: der Aufbau eines Höhenmodells im ASCII-Format, wie es von der Esri-Software erstellt wird	81
Listing 3: Das Python-Script zur automatisierten Erstellung der Kacheln	100

Verzeichnis der Tabellen

Tabelle 1: Die verschiedenen 3D-fähigen Internet Formate und Anwendungen im Vergleich	69
Tabelle 2: Anforderungskatalog der WGM an die Visualisierung der geologischen Daten	79
Tabelle 3: Die Schichten des geologischen Modells in ihrer generellen Reihenfolge.	82
Tabelle 4: Die korrespondierenden Felder der ASCII-Dateien und des X3D-Knotens ElevationGrid	90
Tabelle 5: Umsetzungsstatus des Anforderungskataloges aus Tabelle 2	128

Abstract

Rapid development of computer technology allows the widespread use of 3D graphics in Geoinformation now more than ever. Especially important for geological data, 3D rendering enhances the intuitive grasp of the data and enables the user to interactively explore it. The Wiener Gewässer Management Gesellschaft mbH (WGM) is looking for a solution to distribute their data to a wide range of interested parties, experts and non-experts alike. The first step of this thesis is to examine the nature of 3D data in general, how it is displayed, how objects can be modelled and what the unsolved issues are. Different approaches from the field of geology are introduced to get an overview of the needs and practices that play a role in creating geological 3D models. Then, the internet was taken into consideration for the public distribution and visualization of 3D data and it was found that it offers numerous ways and formats to meet the purpose. Based on these preparational thoughts the second part of the thesis deals with the test of the most promising solution by actually creating a 3D geological map with WGM's data. X3D was chosen for several reasons, among them interoperability, the status of the ISO-standard and the X3D Earth component. After a description of the challenges and possibilities that were encountered during the realization of the visualization it was found that the use of X3D – although not fulfilling all the promises – lead to satisfactory results, that could probably not have been achieved with another technology. Additionally taking into consideration the current developments of X3D the use of this technology was recommended for the purposes of WGM.

Zusammenfassung

Die schnelle Entwicklung der Computertechnologie erlaubt heute mehr denn je die weit gestreute Verwendung von 3D-Grafiken in der Geoinformation. In der Geologie spielen 3D-Darstellungen eine besonders wichtige Rolle für das intuitive Begreifen der Daten, da sie eine synoptische Darstellung bieten und vom Nutzer interaktiv erkundet werden können. Die Wiener Gewässer Management Gesellschaft mbH (WGM) sucht nach Möglichkeiten, geologische Daten einem breiten Publikum aus sowohl Experten als auch Laien zugänglich zu machen. Im Rahmen dieser Arbeit wurde dementsprechend untersucht, in wie weit der aktuelle Stand der Technik dies zulässt. Dazu wurde zuerst die Beschaffenheit von 3D-Daten im Allgemeinen betrachtet. Es wurde untersucht, wie Objekte am Computer modelliert und dargestellt werden, und was der derzeitige Forschungsbedarf ist. Verschiedene Ansätze, geologische Daten dreidimensional umzusetzen, werden beschrieben, um einen Überblick über die Anforderungen an solche 3D-Darstellungen zu gewinnen. Für die Frage der Verbreitung und Visualisierung der 3D-Daten wurde das Internet untersucht, und die wichtigsten der vielen Möglichkeiten, die es bietet, werden erläutert. Aufgrund dieser Vorüberlegungen wurde als geeignete Lösung X3D identifiziert. Es wurde unter anderem wegen der Interoperabilität, des offenen ISO-Standards und der X3D-Earth-Komponente ausgewählt. Das gewählte Mittel wurde dann durch die Konstruktion einer geologischen 3D-Karte unter Verwendung von Daten der WGM getestet. Nach der Bewertung der Probleme und Möglichkeiten, die bei der Umsetzung der Visualisierung auftraten, konnte festgestellt werden, dass die Verwendung von X3D zu zufrieden stellenden Ergebnissen führte, die mit anderen

Lösungen wahrscheinlich nicht hätten erreicht werden können, auch wenn nicht alle Erwartungen an das Format eingehalten wurden. X3D wird daher für das Vorhaben der WGM empfohlen.

Einleitung

Die rasante Entwicklung der Computer- und Internettechnologie eröffnet neuartige Möglichkeiten der Kommunikation. Geodaten, nach modernen kartographischen Gesichtspunkten aufbereitet, können schnell und einfach auf jeden PC übertragen werden und Entscheidungsträgern aus Politik und Planung (oft geographische und kartographische Laien) zur Verfügung gestellt werden. Besonders reizend am Datenaustausch via Internet scheint die Interaktivität zu sein, die dieses Medium bietet. Von der zielgerichteten Auswahl der gewünschten Daten, sei es nach Lage, Thema oder ähnlichem bis hin zur Art der Darstellung bietet das Internet alle Möglichkeiten. Für 3D Ansichten wäre die Möglichkeit besonders wichtig, die Darstellung des Objekts interaktiv zu gestalten, sodass Drehen, Vergrößern, Verkleinern etc. möglich wird.

Die prominentesten Vertreter der Programme, die es erlauben 3D-Geodaten aus dem Internet anzusehen, sind wohl die Earth Browser. Durch eine Schnittstelle können alle Nutzer die Darstellungsweise beeinflussen sowie, noch wichtiger, eigene Daten darstellen lassen. Dies geschieht mit KML/KMZ-Dateien. Es gibt in diesem Bereich sehr offene Systeme, die weit mehr Manipulationsmöglichkeiten seitens der Nutzer zulassen (Bodum & Jaegly 2006). Von der NASA wird World Wind betrieben, ein in Java programmierter virtueller Globus. Versierte Nutzer können selber kleine Zusatzprogramme schreiben, um die Funktionen des Globus' zu erweitern oder in eigene Applikationen einzubauen und diese Technologie somit weiter in Richtung GIS drängen. Das bisher offenste Projekt ist GRIFINOR (vgl. Bodum & Jaegly 2006) von der Universität in Aalborg, von dem sich alle Interessierten den Sourcecode herunterladen können. Es sollte ein Rahmen im Open Source-Format geschaffen werden, der es jedem ermöglicht, eigene Daten in jeder erdenklichen Weise darzustellen. Eine Eignung für geologische Mehrschichtmodelle könnte möglich sein.

Ein anderer Ansatz, Geodaten im Internet zu veröffentlichen, sind die Webdienste, die vom OGC standardisiert werden, namentlich der noch in der Diskussion befindliche Web3DService (W3DS). Neben der Erstellung ist besonders die Visualisierung der dritten Dimension interessant, die z.B. für Digitale Geländemodelle (DGM) wichtig

ist. Die Eignung für geologische Schichten soll in dieser Arbeit untersucht werden. In diesem Zusammenhang sind auch der Web Perspective View Service des OGC zu nennen, weitere in der Entwicklung befindliche Internetdienste, die die dritte Dimension unterstützen. Hagedorn et al. (2010) diskutieren die unterschiedlichen Wege der beiden Services in der Darstellung von 3D-Daten im Internet. Web3DServices erstellen Szenengraphen und geben unter anderem Daten im X3D-Format aus. X3D-Earth ist ein Spezialformat, das georeferenzierte Daten unterstützt. Im Gegensatz dazu steht der WebPerspectiveViewService, der 2D-Abbilder von 3D-Daten ausgibt.

Anlass für die Überlegungen dieser Arbeit ist das 10-jährige Forschungsprojekt „Angewandte Hydrogeologische Forschung – Stadtgebiet Wien“ der Wiener Gewässermanagement Gesellschaft mbH (WGM), einer 100% Tochter der Stadt Wien.

Das Projektziel ist die 3D-Visualisierung von geologischen Daten und deren Nutzung via Internet. Das Technische Büro für Geologie der WGM beschäftigt sich mit Fragen rund um die Grundwassersituation Wiens und wertet hydrogeologisch wertvolle Daten über den Untergrund Wiens aus. Die Ergebnisse werden den anderen Magistratsabteilungen als Entscheidungsgrundlage zur Verfügung gestellt. Da es für Laien nicht unbedingt einfach ist, geologische Karten zu lesen und zu verstehen, versucht die WGM komplexe Inhalte einfach zu visualisieren. Für die Geologie scheinen 3D-Darstellungen auf den ersten Blick besonders gut geeignet zu sein, da nicht nur eine einzige Oberfläche dargestellt werden soll, sondern gleich mehrere übereinanderliegende Schichten. Die Darstellungen sollen über das Internet als ubiquitäres und allgemein genutztes Medium verfügbar gemacht werden.

Die vorliegende Arbeit soll nun einen Ansatz aufweisen, mit dem das Projekt umzusetzen ist und beschäftigt sich daher mit folgenden Forschungsfragen:

- Was ist der derzeitige Stand der Technik, und was ist demnach möglich in Bezug auf 3D-Visualisierungen im und durch das Internet?

- Welche der angeführten Methoden sind für die Umsetzung des Projektes am besten geeignet, bzw. welche Modifikationen müssen dazu durchgeführt werden?

Die Möglichkeiten, die die aktuellen Technologien für die heutige Kartographie theoretisch bereithalten, sollen erörtert und Grenzen bzw. Entwicklungsmöglichkeiten aufgezeigt werden. Die Anwendbarkeit der Web- und 3D-Technologie auf reale Fragestellungen wird bei der Entwicklung von Lösungsansätzen für das Forschungsprojekt der WGM überprüft.

In dieser Arbeit soll grundsätzlich zwischen zwei Prozessen im Zusammenhang mit der Erstellung von 3D-Modellen unterschieden werden:

- Modellierung
- Visualisierung

Ein Modell in den Naturwissenschaften ist ein auf die wesentlichen Aspekte reduziertes Abbild der Wirklichkeit. Es ist eine Abstraktion der Wirklichkeit (Brockhaus 2006). Die Reduktion auf ‚das Wesentliche‘ impliziert bereits, dass eine gewisse Interpretation der Daten vorgenommen werden muss. Im Allgemeinen müssen naturwissenschaftliche Modelle nicht unbedingt einer bildlichen Darstellung entsprechen. So können zum Beispiel Naturgesetze durch mathematische Formeln dargestellt werden. Trotzdem handelt es sich bei diesen Axiomen um vereinfachte und abstrahierte Beschreibungen der Natur und damit um Modelle. Der Inhalt des Modells ist eine Idee oder Aussage. Es ist eine Zusammenfassung des Kenntnisstands zu einem bestimmten Zeitpunkt. Lassen sich gute Voraussagen mit einem Modell treffen, kann es als wirklichkeitsnah oder realistisch bezeichnet werden (Brockhaus 2006). Unter Modellierung soll hier der Prozess verstanden werden, der sich mit der Frage beschäftigt, auf welche Weise man die zu beschreibenden Phänomene abstrahiert. Im Rahmen dieser Arbeit bedeutet das die Frage, wie Geodaten im – hier: virtuellen – dreidimensionalen Raum dargestellt werden können.

Dagegen soll der Begriff der Visualisierung abgegrenzt werden. Visualisierung beschreibt die „bildliche Aufbereitung, Darstellung und Kommunikation von Informationen“ und ist den ästhetischen Wissenschaften zuzurechnen (Brockhaus 2006). Die Funktion der Visualisierung besteht darin, Zusammenhänge oder „simultane und synopt. Vergegenwärtigungen“ (Brockhaus 2006) darzustellen. Visualisierung muss nicht immer auf Graphiken beschränkt sein. Beispielsweise können Maschinenteile gefräst werden, ohne dass vorher eine Visualisierung durchgeführt wurde, nur aufgrund der Definition des Objekts in einer Programmiersprache (Requicha 1980).

Eine strikte Trennung der beiden Begriffe ist nicht immer gegeben, da sich gerade in den Geowissenschaften einige Überschneidungen ergeben. So ist die Formulierung geographischer oder geologischer Ideen eng an die Verbildlichung geknüpft, ja auf andere Weise fast nicht möglich. Zwar könnten Texte oder Tabellen zur Beschreibung der Landschaft herangezogen werden, aber diese können Kontinua, fließende Übergänge, etc. nicht in eine Beschreibung einbeziehen und würden schnell im wahrsten Sinne des Wortes unübersichtlich werden. Eine große Anzahl von Tabellen mit Höhendaten ist für einen Menschen nicht mehr zu verstehen und erlaubt keinerlei Deutung des Phänomens. Also muss die prinzipielle Eignung solcher Methodik in Frage gestellt werden, da der Erkenntnisgewinn aus reinen Zahlenkolonnen (auch wenn sie systematisch angeordnet sind) für den Menschen unmöglich ist. Die Darstellung mithilfe von Zahlen ist daher auch generell eher auf diskrete Phänomene zu beziehen (vgl. Brockhaus 2006).

In vielen GIS ist trotzdem die Speicherung der Rohdaten von der Visualisierung getrennt zu sehen, was es ermöglicht, die Daten je nach Verwendung auf unterschiedliche Weise zu verbildlichen. Auch wenn im 3D-Bereich wiederum diese Trennung nicht so strikt zu sehen ist, da oftmals das 3D-Modell selbst als eine Art der Visualisierung angesehen wird (Basanow et al. 2007). Obwohl noch nicht voll auf den 3D-Bereich übertragbar, soll hier trotzdem diese Unterscheidung durchgeführt werden, weil für die Modellierung von 3D-Daten in der Praxis auch oft ganz andere Methoden verwendet werden als für die Visualisierung, was sich auch grundsätzlich in den verwendeten Werkzeugen widerspiegelt. Beispielsweise wird die Erstellung

geologischer Modelle aus Interpolierungen von punkthaften Messdaten oft mit anderer Software umgesetzt als die Visualisierung der Modellierungs-Ergebnisse. Zwar hat jede Modellierungssoftware auch geeignete Funktionen zur Visualisierung zur Überprüfung des Modells und intuitiven Zusammenstellung der Daten, sonst könnte sie gar nicht verwendet werden. Aber wenn vor allem der Aspekt der Kommunikation in Betracht gezogen wird, sind die Grenzen dieser Programme schnell erreicht. Verbreitungen und Austausch der Daten als bildliche Darstellung sind hier meist nur im jeweiligen, eng begrenzten Rahmen der Software-Pakete möglich, über die nicht jeder verfügt. Beim Export oder Ausdruck der Modelle als reine Abbildungen geht zusätzlich ein besonderer Aspekt der Visualisierung im Computerzeitalter verloren: Die Interaktivität.

Die Ansprüche an eine Visualisierung betreffen also stark die Aspekte des Austauschs in möglichst allgemein gültigen (Computerdaten betreffend) oder verständlichen (für Menschen interpretierbaren) Formaten, die alle Funktionen des Modells erhalten oder sogar noch erweitern sollen. Das heißt, dass die Software zur Visualisierung von Geodaten den Entwickler der Visualisierung in die Lage versetzen muss, die Daten, aus denen die Modelle bestehen, „zusammensetzen“, die Software muss sie „lesen“ können, um sie daraufhin sichtbar zu machen. Nachdem nun also der Begriff des Modellierens in dieser Arbeit eher als Vorstufe zur Visualisierung verstanden werden kann, soll in hier aber nicht die geologische Prozessfindung nachempfunden werden. Denn die grundlegende Interpretation der Daten muss den geologischen Fachleuten obliegen. Die Identifizierung und Anwendung der richtigen Interpolationsverfahren, der Einfluss unterschiedlicher Faktoren auf den Aufbau der geologischen Schichten etc. bleibt den Experten des geologischen Fachgebiets vorbehalten und soll nicht Teil dieser kartografischen Überlegungen sein. In dieser Arbeit werden bei der kartographischen Umsetzung eines geologischen Modells aber sehr wohl die der Visualisierung zugrunde liegenden Prozesse als Modellierung verstanden. Dementsprechend werden hier vor allem die Fragestellungen der technischen Umsetzung der Computer 3D-Grafik im Kapitel 3 **Modellierung** erörtert werden, also die Algorithmen zur Erstellung von 3D-Körpern. In diesem Zusammenhang werden einige ausgewählte Probleme der kartografischen Analyse von 3D-Datenmodellen, die bisher noch ungeklärt sind, näher erläutert werden. Im Abschnitt 4 **Visualisierung**

werden dagegen Verfahren und Regeln zur kartographischen Darstellung der Objekte im Vordergrund stehen, womit die bildliche und interaktive Repräsentation der Ergebnisse aus dem Modellierungsprozess gemeint ist (vgl. Requicha 1980).

Die Arbeit ist in zwei Teilen organisiert. Der erste beschäftigt sich mit Fragen der Technik und analysiert die grundsätzlichen Fragestellungen der 3D-Visualisierungen im Internet. Teil II beschreibt die Umsetzung des speziellen Modells der WGM, nachdem eine qualifizierte Entscheidung für das Mittel zur Visualisierung getroffen wurde. Er schließt mit einer Bewertung der gewählten Methode in Hinblick auf Eignung und zukünftigen Entwicklungen ab.

Teil I
Theoretische Vorüberlegungen

1 Die dritte Dimension in der Kartographie

Die Darstellung der dritten Dimension in der Kartographie hat seit der großen Verbreitung von Heimcomputern mit einer gewissen Leistung, was die Graphik angeht, einen großen Schub erfahren. Zwar wurde in der Kartographie immer wieder nach Lösungen gesucht, nicht nur Papierkarten im 2D-Format herzustellen. Pioniere des „alten Schlags“, wie zum Beispiel E. Imhof, bauten noch eigenhändig plastische Modelle aus Holz und anderen Materialien. So faszinierend und schön anzuschauen diese „realen“ Modelle auch sind, so teuer und aufwändig sind die Herstellungsverfahren. Ebenso wird eine weite Verbreitung dieser Modelle allein schon durch die physische Größe, das entsprechende Gewicht und die Unhandlichkeit verhindert. Daran ändern auch moderne computergestützte Herstellungsverfahren nichts, die mit Hilfe von CNC-Fräsen oder spezieller 3D-Drucktechniken die Herstellung maßgeblich beschleunigen. So bleiben die wahrscheinlich anschaulichsten und beeindruckendsten Darstellungen der dreidimensionalen Erdoberfläche meist nur Museen oder anderen Ausstellern vorenthalten. Eine andere Möglichkeit, sich der dritten Dimension zu nähern sind perspektivische Darstellungen, zum Beispiel aus der Vogelschau. Besonders perfektioniert hat diese Art der Darstellung in künstlerischer Sicht H.C. Berran, der weltberühmte Panoramabilder von Berglandschaften erschaffen hat. Diese Bilder sind zwar keine Karten im strengen Sinne der Wissenschaft. Der Maler hat sich einige künstlerische Freiheiten herausgenommen, wie zum Beispiel die perspektivische Vergrößerung wichtiger Bildelemente. Es werden markante Berge im Hintergrund vergrößert dargestellt, ja manchmal sogar in ihrer Position verschoben, um dem Betrachter die Orientierung anhand von bekannten und herausstechenden Naturmerkmalen zu vereinfachen und das Thema der Abbildung zu betonen. Da sie nicht den kartometrischen Ansprüchen einer „echten“ Karte genügen, sind sie wohl eher den kartenverwandten Ausdrucksformen zuzurechnen. Sie zeigen aber den unbestrittenen Reiz von perspektivischen Ansichten der Landschaft und lassen einige Vorteile der eher vertrauten und anschaulichen, weniger abstrakten Abbildung der Wirklichkeit erkennen. Eine weitere Alternative stellt die 3D-Computergrafik dar. Der heutige Stand der Technik lässt schon eine weitestgehend ruckelfreie Darstellung auch komplexer 3D Grafiken zu, die einigermaßen realitätsnah aussehen.

Der große Vorteil der Computerdatentechnologie liegt nicht nur in der relativ einfachen Übertragung der Daten via Datenträger oder Netzwerke, sondern auch in der Flexibilität, die solche Daten zulassen. Computermodelle können ohne großen Aufwand zu jedem Zeitpunkt der Erstellung manipuliert und verändert werden. Das betrifft Fehlerausbesserungen oder Aktualisierungen, also die zugrunde liegenden Daten des Modells selbst. Es kann dem Nutzer des Modells auch die Möglichkeit eingeräumt werden, auf die Darstellung des Gesehenen Einfluss zu nehmen, ja sogar in die Grundstruktur des Modells einzugreifen. Am Computer können heute virtuelle Welten entstehen, die von den Nutzern „erforscht“, also interaktiv entdeckt werden können. Lorenz et al. (2009) stellen die mittlerweile sehr weite Verbreitung von virtuellen 3D-Welten heraus. Diese werden sowohl von vielen Laien genutzt, dienen aber auch einem wissenschaftlichen und Fachpublikum zur Planung, Analyse und Visualisierung von Geodaten. In jedem Fall müssen die 3D-Darstellungen „technischen, gebrauchstauglichen und ästhetischen“ (Lorenz et al. 2009) Anforderungen entsprechen.

1.1 Aktuelle Forschungsgebiete

Im Folgenden sollen exemplarisch einige aktuelle Forschungsgebiete der 3D-Kartographie näher erläutert werden. Das Feld der Thematik ist jetzt schon so groß, dass es unmöglich wäre, einen vollständigen Überblick über alle Strömungen der 3D-Geovisualisierung zu geben. Es soll in diesem Abschnitt deutlich gemacht werden, wie unterschiedlich sich die Tätigkeitsfelder auf dem Gebiet darstellen und es sollen einige der neuesten Entwicklungen und Fragestellungen aufgezeigt werden.

1.1.1 Multiperspektivische Ansichten

Lorenz et al. (2009) greifen in ihren virtuellen Welten auf den multiperspektivischen Ansatz zurück, der erstmals von H.C. Berann im kartographischen Kontext verwendet wurde. Dies soll das Verständnis der kartographischen Darstellung intuitiver machen. Bei diesem Verfahren finden sich in einer einzigen Darstellung mehrere Perspektiven wieder. Der Vordergrund der Karte ist aus der Vogelperspektive, also von schräg oben, abgebildet, während der Horizont aus einer flachen Perspektive, wie zum Beispiel der eines Fußgängers, dargestellt wird. Der Übergang zwischen den beiden Bildabschnitten ist fließend. Man könnte sich zur Veranschaulichung ein Relief

vorstellen, das auf eine Walze aufgetragen wurde: der untere Teil des Bildes steht senkrecht zum Betrachter, während der obere Teil „auf der Walze“ aufliegt und dem Betrachter die weiter entfernten Merkmale der Landschaft in horizontaler Sichtweise präsentiert. Eine andere Variante der Multiperspektive stellt die Fußgängerperspektive in den Vordergrund. Hier wird beispielsweise ein virtuelles 3D-Stadtmodell mit Gebäuden, Bäumen etc. so gezeigt, wie es ein Fußgänger in der realen Welt sehen würde. In Richtung Hintergrund wird dann der Horizont „aufgeklappt“, sodass der Bereich, der normalerweise dem Himmel zufallen würde, jetzt mit einer Karte (2D, Parallelprojektion) ausgefüllt ist, die sich nahtlos an die 3D-Welt aus der Fußgängerperspektive anfügt. Um beim Vergleich mit der Walze zu bleiben: Es wird der Eindruck vermittelt, als befände sich der Beobachter im Innern des Zylinders, in dessen unteren Bereich sich das 3D-Modell befindet, während die oberen, höheren Bereiche mit einer Karte bedruckt wären. Lorenz et al. (2009) beschäftigen sich mit der automatischen Generierung solcher multiperspektivischen Darstellungen von beliebigen Standpunkten aus einem virtuellen 3D-Modell heraus. Bei virtuellen Interaktionen, wie beispielsweise einer Bewegung der Kamera (Änderung des Sichtwinkels in fließender Manier), muss ein möglichst anschaulicher, das heißt natürlicher, Eindruck der Bewegung erzeugt werden. Mithilfe der Multiperspektive in 3D werden neue Formen der Detail- und Überblickskartierung aufgezeigt, die die Darstellung in der einfachen Perspektive bereichern und die Gestaltungsfreiheitsgrade in der Kartographie erweitern soll.

1.1.2 LiDAR

Laser-Scanning, auch unter der Bezeichnung LiDAR (für „Light Detection And Range“) bekannt, ist eine der neuesten Technologien in der Fernerkundung, die direkt 3D-Daten aufnimmt. In den letzten zehn Jahren hat sie sowohl die topographische Kartierung als auch die 3D-Aufnahme von Objekten revolutioniert (Kraus 2007 S. 400) Der erste Laserscanner wurde im Jahr 1994 verkauft. Seither fand eine rasante Entwicklung statt, in deren Lauf sich die Leistungen der Laserscanner verzehn- wenn nicht sogar ver Hundertfacht haben (Pfeifer et al. 2008).

Die Scanner sind mittlerweile auf allen möglichen Gebieten im Einsatz. Diese reichen von Architektur, über Archäologie, bis hin zur Roboternavigation oder der Medizin. Der Fokus im kartographischen/geographischen Bezug liegt dabei auf zwei Gruppen

von LiDAR: das terrestrische Laser-Scanning (TLS) vom Boden aus und das Airborne Laser-Scanning (ALS), bei dem der Laser von einem Flugzeug oder Hubschrauber getragen wird.

Laser-Scanner gehören zu den aktiven Fernerkundungsmethoden. Das bedeutet, dass sie selber Energie ausstrahlen, deren Rückstreuung zur Vermessung verwendet wird. Laserlicht ist ausgesprochen kollimiert (parallel gerichtet), monochromatisch und kohärent (zusammenhängend) (Kraus 2007 S. 400), und daher für genaue Messungen besonders gut geeignet. Die Rückstreuung verhält sich analog zum akustischen Echo (Pfeifer et al. 2008). Vom Laser wird ein Signal ausgegeben, das auf die Erdoberfläche oder die Oberfläche irgendeines Objektes trifft. Von dort wird es reflektiert und gelangt zum Empfänger zurück, der sich direkt am Laser befindet. Aus dem Zeitintervall, das zwischen Aussenden und Empfangen des Signals vergeht, wird die Entfernung des Reflexionspunktes errechnet. Seine Koordinaten im dreidimensionalen Raum werden direkt als polare Koordinaten gespeichert. Wenn die Lage des Scanners im Bezug zur Erdoberfläche bekannt ist, kann die Lage des Punktes in ein kartesisches Koordinatensystem transferiert werden. Die Lage des Scanners wird mit Hilfe eines GPS (Global Positioning System) bestimmt, das für eine möglichst große Genauigkeit im differentiellen Modus betrieben wird (dGPS). Neben der Laufzeit des Signals wird auch die Richtung des Laserstrahls verzeichnet, die durch einen Spiegel beständig verändert wird. Auf diese Weise tastet der Laserstrahl die Oberfläche in ihrer Gesamtheit ab. So werden die 3D-Koordinaten zahlreicher Punkte auf der reflektierenden Oberfläche erfasst, die zusammen das Objekt oder Gelände geometrisch beschreiben. Die Gesamtheit dieser Punkte wird als Punktwolke bezeichnet (Abbildung 1).

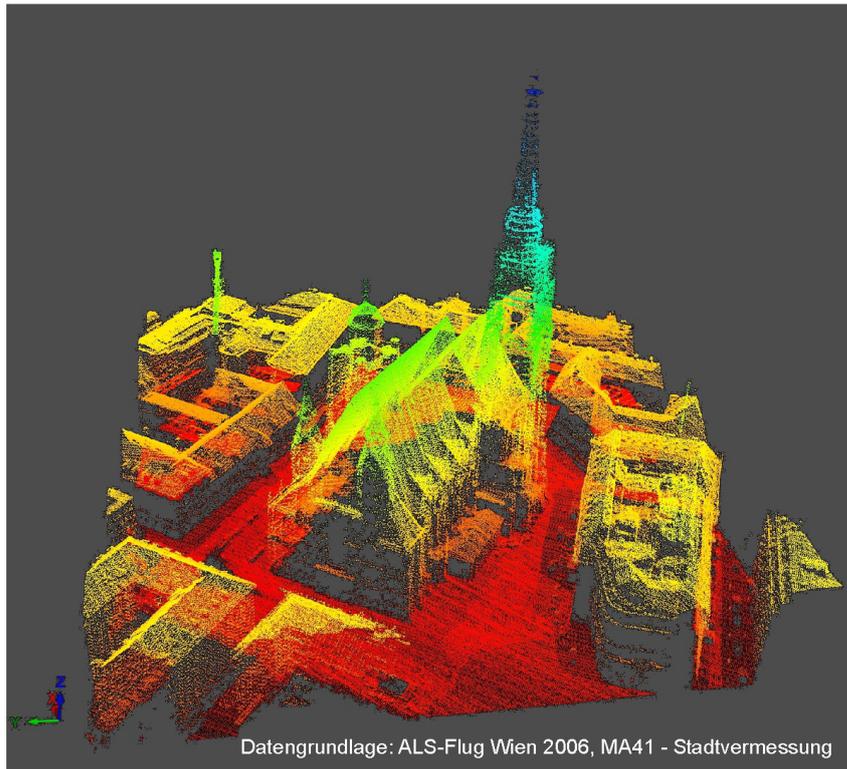


Abbildung 1: Der Stephansdom in Wien als Punktwolke. Quelle: www.riegl.com

Nach der Aufnahme und Einordnung der Punkte in ein kartesisches Koordinatensystem kann mit der Visualisierung begonnen werden. In den Punktwolken können die einzelnen Punkte anhand von Intensität, Strecke und/oder Höhe farblich oder in Graustufen codiert werden. Sie können aber auch mit Digitalbildern und Orthofotos verbunden werden, sodass noch realistischere 3D-Abbildungen entstehen. Es können weiters auch Höhenlinienpläne oder Animationen aus den Daten abgeleitet werden und vieles mehr.

1.1.3 Statistische 3D-Oberflächen

Die 3D-Visualisierung muss sich aber nicht auf die gegenständliche Darstellung real existierender Objekte beschränken, wie in den oben angeführten Beispielen gesehen. Sie kann auch dazu dienen, statistische Größen zu visualisieren. Laut Sieber et al. (2009) haben statistische Oberflächen viele Vorteile. Durch die Belegung der dritten Dimension mit einer oder mehrerer Variablen lassen sich Zusammenhänge erkennen und Vergleichen. Kleine Unterschiede fallen sofort auf.

Statistische Oberflächen werden auf Grundkarten von real existierenden Oberflächen angebracht und extrudieren betreffende Gebiete, zum Beispiel die Flächen von

Bundesländern, je nach Größe der Variable zum Beispiel zu einem Säulendiagramm. Dadurch nehmen sie aber keinen Einfluss auf die Grundfläche, die in der Kartengenauigkeit der Grundkarte erhalten bleibt. Das macht solche Karten oft einfacher zu lesen, als verzerrte Kartogramme (Sieber et al. 2009). Auch die Höhenwerte einer Oberfläche als Kontinuum können direkt durch statistische Werte ersetzt werden und es können dann glatte kontinuierliche Veränderungen in den statistischen Daten anschaulich dargestellt werden. Die hohen und niedrigen Werte werden so buchstäblich als Höhen und Täler dargestellt (Abbildung 2).

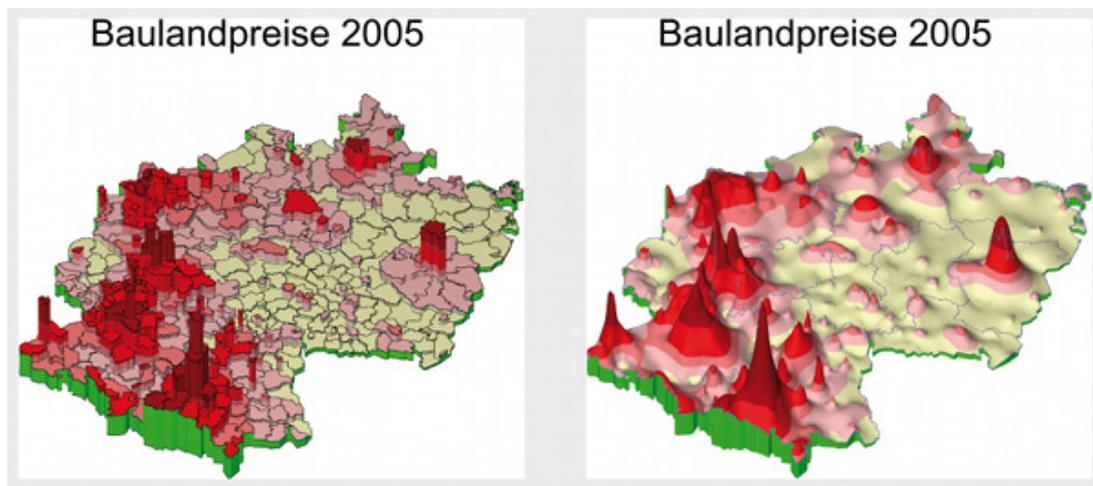


Abbildung 2: Die Baulandpreise Deutschlands als dreidimensionale statistische Oberflächen, links als extrudierte Prismen, rechts als Kontinuum. Quelle: Bundesinstitut für Bau-, Stadt- und Raumforschung 2009

Auf diese Weise können sehr komplexe Sachverhalte, wie zum Beispiel Tatorstatistiken, sehr gut verständlich verbildlicht werden, sodass statistische Oberflächen eine wertvolle Unterstützung für Entscheidungsträger sein können (Wolff 2009).

1.2 Sonderfall Geologie

Die 3D-Modellierung von Daten der Erdoberfläche ist zum Teil sehr weit vorangekommen, wie die oben genannten Beispiele illustrieren. Für den geologischen Untergrund sieht die Situation allerdings anders aus. Neber et al. (2006) beklagen, dass Daten von unterhalb der Erdoberfläche schicht sehr oft ignoriert werden. Baojun et al. (2009) stellen zwar fest, dass in den letzten 10 Jahren ein großer Fortschritt in diesem Bereich zu beobachten sei. Allerdings seien die Voraussetzungen für eine 3D-

Modellierung von geologischen Daten mit viel Vorbereitungen, hohen Kosten und komplizierten Verfahren verbunden. Die kommerziellen Softwarepakete erlauben zudem meist nur eine Ansicht der Daten in der proprietären Modellierungsumgebung. Beard (2006) sieht weitere Herausforderungen bei der Modellierung und Kommunikation von 3D-Modellen im Allgemeinen aber insbesondere für geologische Modelle. Für die Erstellung ist oftmals eine komplizierte Software erforderlich, deren Bedienung meist nur Experten auf diesem Gebiet vorbehalten ist. Die Software, mit der die Modelle erstellt wurden, ist dann meistens auch für die Betrachtung notwendig. Die spezialisierten Dateiformate lassen sich meist nur in Programmen desselben Herstellers öffnen, gerade wenn die gesamte Interaktivitätsfunktion erhalten bleiben sollen.

Warum ist die Darstellung der dritten Dimension gerade in der Geologie so wichtig? Werden geologische Karten in 2D auf Papier herangezogen, so bleibt es oft nicht bei der Verwendung einer einzigen Karte allein. Profile, Höhenlinienkarten etc. werden zur Veranschaulichung hinzugezogen. Beard et al. (2005) sehen im Gegensatz dazu in den digitalen 3D-Modellen die Möglichkeit, die wahren räumlichen Zusammenhänge zu beobachten. Ein dynamisches 3D-Modell kann die Geologie des Untergrundes detailreicher darstellen als eine traditionelle 2D-Karte. Also kann eine gute Visualisierung von geologischen Daten sogar dazu verhelfen, unnötige Erhebungen im Feld, wie zum Beispiel Bohrungen, zu vermeiden (Neber et al. 2006). Die Erfahrung zeigt, dass die geologischen Zusammenhänge in einem 3D-Modell auch von Nicht-Experten gut verstanden werden können, da sie in der vertrauten dreidimensionalen Form präsentiert werden, und nicht erst aus der 2D-Darstellung heraus abstrahiert werden müssen, was viel Übung erfordert (Beard et al. 2005). Aber nur benutzerfreundliche, gebrauchsfertige 3D-Daten ermöglichen das intuitive Begreifen der Geologie. Die dreidimensionale Visualisierung von geologischen Daten kann dann beispielsweise eine leichtverständliche Entscheidungsgrundlage für eine nachhaltige Regionalplanung werden (Neber et al. 2006).

Im Rahmen dieser Arbeit werden Wege der Visualisierung von dreidimensionalen, geologischen Daten besprochen und getestet. Zuvor sollen jedoch einige Fragestellungen zur 3D-Modellierung aufgezeigt werden. Die Analyse von

dreidimensionalen Objekten und die Beziehungen zwischen ihnen, die deren räumliche Anordnung beschreibt, bleiben nämlich vorerst recht schwer, wie das folgende Kapitel (2 Modellierung) zeigen wird.

2 Modellierung

Um eine qualifizierte Entscheidung bezüglich der Methodenwahl zur Visualisierung treffen zu können, müssen vorerst die Modellierungsverfahren untersucht werden, die für geologische Daten in Frage kommen. Nur so kann ein späterer Konflikt zwischen Modellierung und Visualisierung vermieden werden, wenn beispielsweise beide Methoden inkompatibel sind oder unterschiedliche Anforderungen unterstützen.

2.1 Repräsentationsformen von 3D-Objekten

Requicha (1980) unterscheidet konkret drei Möglichkeiten, dreidimensionale Geometrien im Computersystem zu repräsentieren:

1. Cell Decomposition
2. Boundary Representation
3. Constructive Solid Geometry

Unter Cell Decomposition versteht man die Zerlegung eines 3D-Objektes (Polyeders) in viele kleinere Zellen (zum Beispiel Tetraeder). Das einfachste Beispiel und zugleich eine Art Sonderfall der Cell Decomposition ist die Spatial Occupancy Enumeration. Darunter versteht Requicha (1980) die gleichmäßige Aufteilung des Raumes in Zellen gleicher Größe und Form. Im Prinzip werden dann diejenigen Zellen aufgelistet, die von einem Volumenkörper in Anspruch genommen werden. Die Zellen werden manchmal als Voxel bezeichnet. Dies steht – analog zu Pixel: *picture element* – für *volume element*. Die Zellen haben immer die gleiche Form, wie zum Beispiel Würfel, Tetraeder usw. und die gleiche Größe. Sie liegen in einem fest vorgegebenen dreidimensionalen Raster. Die Lage der Zellen kann dann mit Hilfe der Koordinaten des Zentroiden der jeweiligen Zelle angegeben werden. Spatial Occupancy Enumeration eignet sich besonders für entweder kastenähnliche Gegenstände, wie also Gebäude, die aus verschiedenen eckigen Modulen zusammengesetzt werden können, oder für extrem unregelmäßige Formen, wie sie im biologischen Kontext, beispielsweise der Computertomographie, vorkommen. Für Gegenstände aus der Industrie aber, die oft weder allzu eckig noch extrem unregelmäßig sind, stößt diese Art der Modellierung an ihre Grenzen. Dann können

sie meist nur relativ grobe Annäherungen an die Formen des Gegenstandes leisten (Abbildung 3).



Abbildung 3: Spatial Occupancy Enumeration bietet oft nur grobe Annäherungsmöglichkeiten an glatte, gebogene Oberflächen. Quellen: http://www.effectware.com/download/images/efx_voxel2.jpg, <http://www.onelargeprawn.co.za/2009/12/22/awesome-voxel-art-by-metin-seven/>

Generell müssen beim Verfahren der Cell Decomposition oder Zellzerlegung die einzelnen Zellen aber nicht zwingendermaßen alle gleich groß sein, wie bei der Spatial Occupancy Enumeration (siehe Abbildung 4).

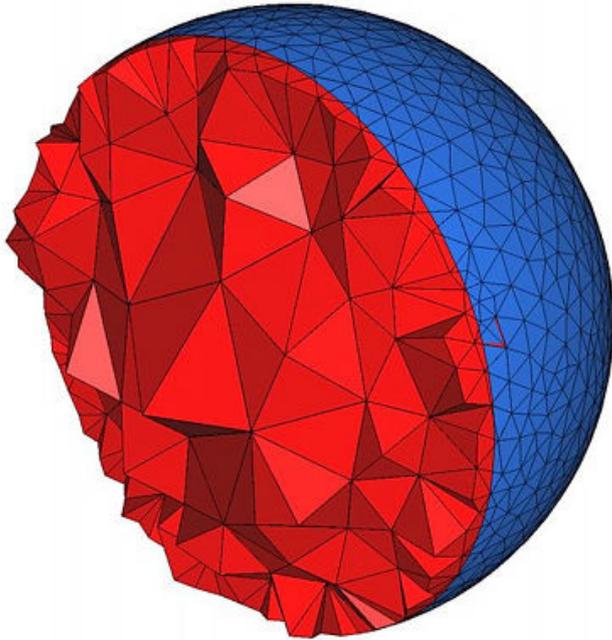


Abbildung 4: Delauney Tetraeder. Quelle: Alliez, P. et al. (2010)
http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Mesh_3/Chapter_main.html

Bei der Zerlegung bietet sich auch die Aufteilung eines Körpers in Delaunay-Tetraeder an, die, analog zu den Delaunay-Dreiecken, so aufgebaut sind, dass die umschließende Sphäre eines Tetraeders keinen Eckpunkt außer den eigenen umfasst. Dabei handelt es sich dann um eine unregelmäßige Zerlegung eines Objektes (Gold 2008).

Bei der Methode der Constructive Solid Geometry (CSG) bedient sich das Grafikprogramm aus einem vorgegebenen Set von primitiven geometrischen Figuren, wie zum Beispiel Zylinder, Sphären oder Quader, die dann entsprechend dem gewünschten Endergebnis zusammengesetzt werden. CSG sind in binären Bäumen organisierte Anordnungen von Einzelteilen eines Ganzen (siehe Abbildung 5).

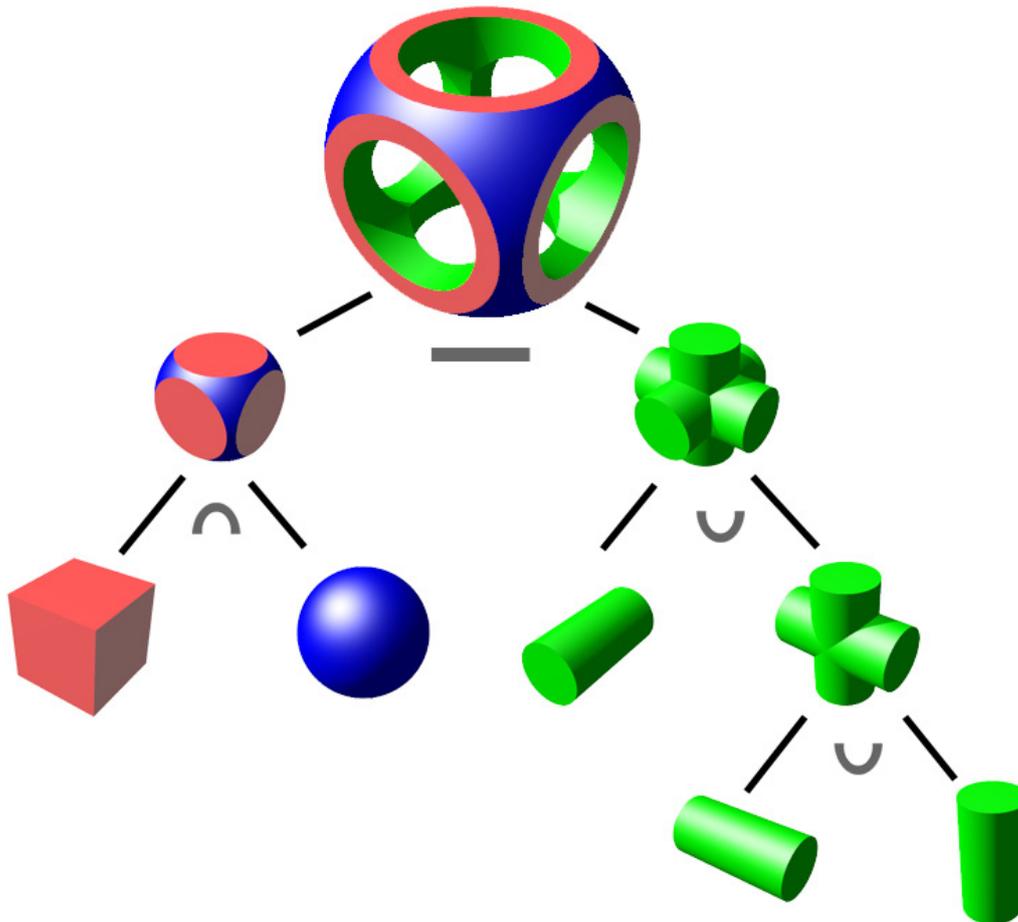


Abbildung 5: Der binäre Baum einer Constructive Solid Geometry (CSG).
Quelle: http://de.academic.ru/pictures/dewiki/67/Csg_tree.png

Die Einzelteile entsprechen den oben genannten Grundformen, und werden durch Boole'sche Operatoren zusammengesetzt, vereint, voneinander abgezogen etc., um einen einzigen Volumenkörper zu erstellen. Jeder Endpunkt des Baums steht für ein Grundobjekt, jede Zwischenstation enthält einen Operator, der anzeigt, was mit dem Objekt gemacht werden soll. CSG können sehr präzise und gleichzeitig auch kurz und bündig sein. Das hängt unter anderem von der Anzahl der verschiedenen Grundobjekte ab. CSG sind nicht gut geeignet für einige grafische Operationen, wie das „anfassen“ von Kanten; linienhafte Darstellungen von Objekten sind nicht möglich. Nach Requicha (1980) könnte man Spatial Occupancy Enumeration auch als Sonderfall von CSG ansehen, in dem nur ein Grundobjekt (zum Beispiel Würfel) und ein Operator (zusammenfügen; nicht vereinen!) zur Verfügung stehen.

Schon 1980 sieht Requicha die dritte Methode, die Boundary representations (B-rep), als am weitesten verbreitete Methode für das Modellieren von 3D-Volumenkörpern an. Das dürfte sich bis heute nicht geändert haben. Bei dieser Methode wird die äußere Begrenzung des 3D-Körpers in viele kleine 2D-Flächen segmentiert, was der Bildung eines 3D-Mosaiks entspricht (englisch: *tessellation*). Die 2D-Flächen werden durch ihre Ränder und Eckpunkte beschrieben (siehe Abbildung 6).

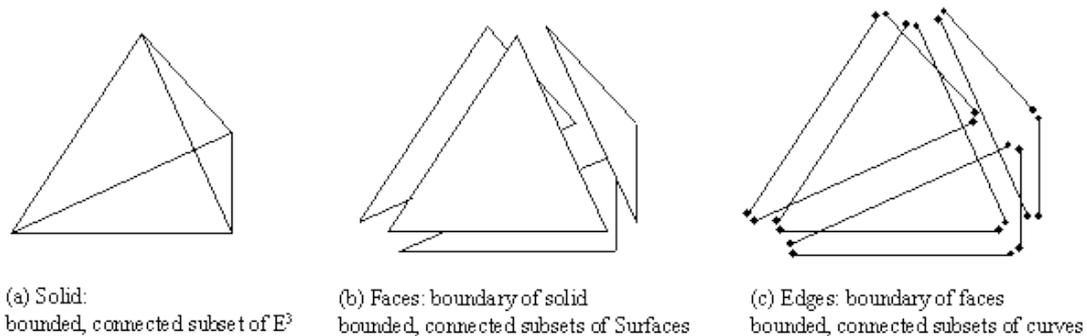


Abbildung 6: Die Zerlegung eines 3D-Körpers in seine Einzelteile: Flächen, Kanten und Punkte. Quelle:
<http://www.ielm.ust.hk/dfaculty/ajay/courses/ieem317/lects/3dcad/boundaries.gif>

Oft werden Dreiecke als kleinstes Oberflächensegment gewählt. Dreiecke sind die einzigen Oberflächenprimitive, die auch von der aktuellen Hardware direkt gerendert werden können. Deswegen ist es günstig, wenn Oberflächen für interaktive Visualisierungen als angenäherte Dreiecksnetze modelliert werden (Greiner 2000). Es kann prinzipiell aber jede Form von geradlinigen Figuren verwendet werden. Requicha (1980) stellt vier Bedingungen auf, um gültige B-reps zu erstellen.

1. Jede der kleinen 2D-Flächen ist Teil einer Oberfläche eines Objektes.
2. Alle 2D-Flächen zusammen bilden die Grenze des Objekts.
3. Jede 2D-Fläche ist eine Teilmenge des Grundobjekts (zum Beispiel Dreieck).
4. Jede 2D-Fläche muss eine Ausdehnung haben und darf keine überstehenden Linien etc. aufweisen.

Punkt zwei und vier entspricht den Bestimmungen des OGC zu den Simple Features (OGC 2006). In den Spezifikationen des OGC wird noch eine weitere Bedingung hinzugefügt:

5. Alle 2D-Flächen müssen gleich ausgerichtet sein.

Das bedeutet, dass bei diesen Flächen zwischen Ober- und Unterseite unterschieden wird. Polygone bestehen aus Kanten, die jeweils durch Anfangs- und Endpunkte definiert sind. Der Endpunkt einer Kante ist der Anfangspunkt der nächsten. In einem geschlossenen Polygon ist das bei allen Punkten der Fall. Wenn alle Polygone einer Oberfläche gleich ausgerichtet sein müssen, heißt das, dass alle Oberflächen in die gleiche Richtung zeigen müssen. In anderen Worten: zwei Liniensegmente, die aneinander angrenzende Polygone definieren, müssen so ausgerichtet sein, dass sie die Grenze in entgegengesetzte Richtungen passieren (Abbildung 7). Das hat zur Folge, dass alle Polygone die Oberfläche auf der gleichen Seite haben.

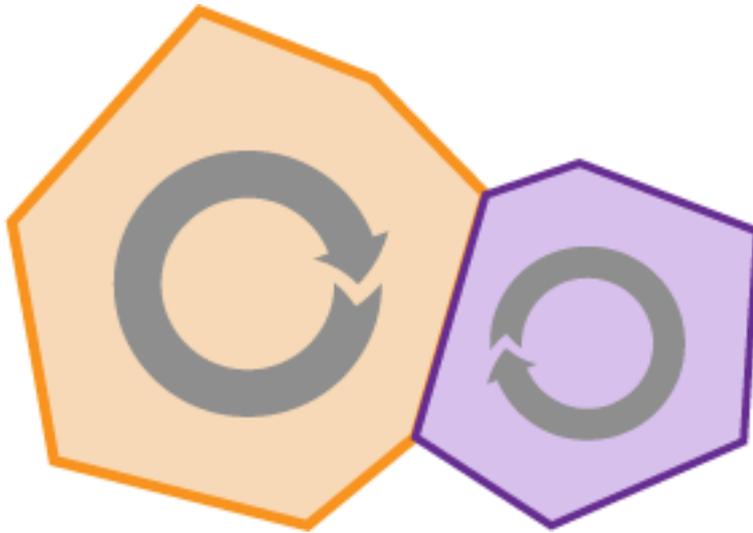


Abbildung 7: Zwei gleich ausgerichtete Polygone, die Teil einer einzigen Oberfläche sind. Die im Uhrzeigersinn weisenden Pfeile zeigen die Ausrichtungen der Kanten (Rand) des Polygons an. Quelle: Eigene Bearbeitung nach OGC 2006

Wenn in einer polyedrischen Oberfläche jedes Liniensegment die Grenze von zwei Polygonen darstellt, handelt es sich um ein Gebilde, das mit einer Kugel isomorph ist und einen Volumenkörper umschließt (OGC 2006). Viele Fußbälle sind nach diesem zusammengesetzt. Ober- und Unterseite der 2D-Flächensegmente werden dann zu Innen- und Außenseite des Volumenkörpers. In den Spezifikationen des OGC werden 3D-Volumenkörper also mittels B-rep modelliert.

Bei Greiner (2000) findet sich eine gute Beschreibung der gängigsten Methoden zum Erstellen von Polygonnetzen für B-reps. Grundsätzlich muss zwischen der Topologie des Netzes und der Geometrie unterschieden werden. Die Topologie beschreibt allein

die Struktur des Netzes, zum Beispiel Dreiecke und deren Kanten, während die Geometrie durch die Koordinaten der Eckpunkte (Vertizes) definiert wird. Die einfachste Art, ein solches Dreiecksnetz zu beschreiben, ist die explizite Netzstruktur (Explicit Mesh Structure). Jedes Dreieck wird durch die Koordinaten seiner Eckpunkte beschrieben. Die Koordinatentupel enthalten die drei Werte für x, y und z. Da die meisten Punkte von mehreren Dreiecken geteilt werden, müssen viele dieser Tupel auch mehrfach abgelegt werden. Das bedeutet, dass man im Endeffekt durchschnittlich etwa 18 Werte pro Vertize abspeichert.

Um die Datenmenge zu reduzieren, setzt man indizierte Netzstrukturen (Indexed Mesh Structure (Shared Vertex)) ein. Es werden zwei Listen getrennt nach Geometrie und Topologie erstellt. In der ersten sind die Koordinatentripel eines jeden Eckpunktes aufgelistet (Vertize-Liste). In der zweiten werden mit Hilfe von jeweils drei Indexzahlen, die auf die Koordinatentripel verweisen, die Dreiecke beschrieben. Dadurch reduziert sich die Zahl der abzuspeichernden Werte bereits auf neun pro Vertex.

Viele Algorithmen fordern Informationen über die Verhältnisse in der Umgebung eines Dreiecks an (Was sind die benachbarten Dreiecke, Kanten oder Eckpunkte?), was auch für topologische Analysen in der Geoinformation eine große Rolle spielt (siehe 2.2 Topologische Probleme). Deswegen werden Strukturen gebraucht, die dies schneller ermöglichen als die einfachen, indizierten Netzstrukturen. In der hierarchischen Ringstruktur beispielsweise (Hierarchical Ring Structure) werden in der Vertize-Liste zusätzlich zu den Koordinaten noch so genannte Zeiger (Pointers) gespeichert. Diese Zeiger verweisen auf alle benachbarten Vertizes und auf alle angrenzenden Dreiecke. Da die Anzahl der benachbarten Punkte und Dreiecke abhängig vom Aufbau des Netzes variiert, wird das Löschen oder Einfügen von Vertizes zu recht komplexen Operationen. In dieser Struktur werden ca. 21 Werte für jeden Eckpunkt benötigt.

Die wohl beliebteste Struktur für Netze aus frei wählbaren Polygonen (nicht nur Dreiecken) ist die Winged-Edge Netzstruktur (Abbildung 8).

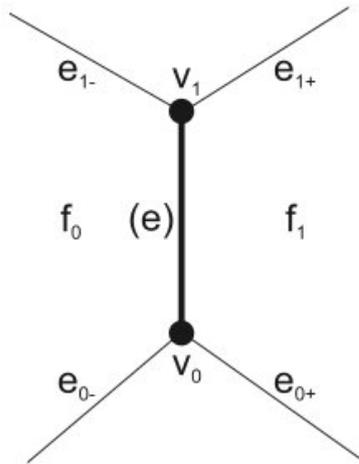


Abbildung 8: Winged Edge Struktur nach Greiner. Quelle: Greiner 2000

Nicht die Vertices (Eckpunkte) der Polygone spielen hier die entscheidende Rolle sondern die Kanten. Für jede Kante werden Zeiger für die jeweiligen Endpunkte, die links und rechts angrenzenden Flächen, und die jeweils zwei Kanten, die von den Endpunkten ausgehen, und die die linke und die rechte Fläche begrenzen, abgespeichert. In der Vertice-Liste wird zu den Koordinaten ein Zeiger zu einer beliebigen der angrenzenden Kanten hinzugefügt. Auf diese Weise werden jegliche topologische Informationen schnell zugänglich. Die Struktur erfordert etwa 30 Werte pro Punkt.

Die flexibelste Methode des 3D-Netzaufbaus ist die Quad-Edge Netzstruktur, in der zwischen Ober- und Unterseite eines Polygons unterschieden werden kann, die sogar lose Enden erlaubt und in der die Endpunkte einer Kante derselbe Vertex sein kann und dergleichen Vorteile mehr (Abbildung 9).

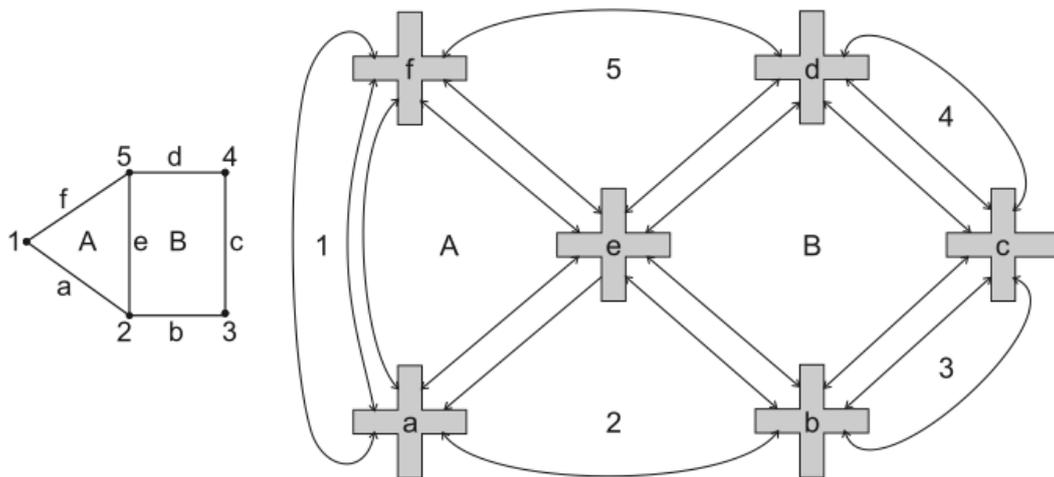


Abbildung 9: Quad Edge Struktur. Quelle: Greiner 2000

Für jede Kante werden vier Listen angelegt, in denen in kreisförmiger Reihenfolge die Zeiger für die zwei Endpunkte und die zwei angrenzenden Flächen abgelegt werden. Diese als Ring angelegten Zeiger weisen aber im Gegensatz zu den vorher besprochenen Strukturen nicht auf Koordinaten in der Vertize-Liste hin, sondern auf den nächsten Eintrag in der Index-Liste für den benachbarten Eckpunkt oder die benachbarte Kante. Die Vertize-Liste enthält Zeiger zu einer frei wählbaren Kante. Darüber kann dann der zur Kante gehörende Ring angesprochen werden. Es werden durchschnittlich 18 Werte pro Punkt gespeichert.

Beim Erstellen solcher B-rep-Modelle fallen, wie gesehen, sehr großer Datenmengen an, die es Nutzern oft unmöglich machen, komplexe 3D-Körper zu modellieren, ohne auf vereinfachende Interfaces zurückzugreifen (Requicha 1980), zum Beispiel in Form eines Graphical User Interface (GUI) oder einer vereinfachenden Programmiersprache.

Auch diverse Methoden vereinfachen die Erstellung von Objekten in B-rep-basierten grafischen Systemen, von Requicha (1980) als Sweeping vorgestellt, in der heutigen Literatur eher als Extrusion bekannt (Abbildung 10).

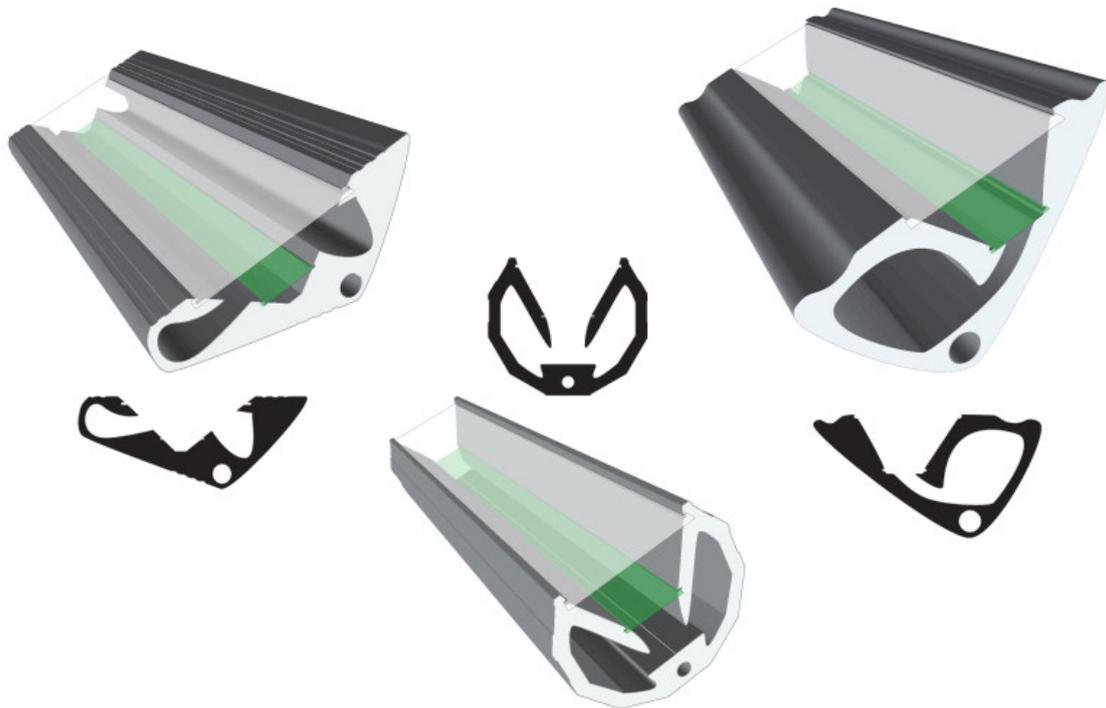


Abbildung 10: Veranschaulichung der Extrusionsoperationen, bei denen eine Grundfläche (schwarz dargestellt) durch den 3D-Raum „gezogen“ wird. Quelle: http://www.sd1.co.uk/images/portfolio/print/extrusion_concepts.jpg

2D-Flächen werden entlang von Trajektorien gezogen um die Oberseite eines Objektes zu definieren. Die Unterseite bildet die alte Position der Fläche, die Seiten werden durch die Verbindung der äußeren Kanten beider Flächen gebildet. Auch durch Drehung eines 2D- oder 3D-Objektes kann ein ähnlicher Effekt erzielt werden. Die Flächen können auf ihrer „Reise“ durch den Raum auch kleiner oder größer werden, um beispielsweise Kegel zu erzeugen, und sich um die eigene oder eine andere Achse drehen. Der Extrusionspfad muss keine Gerade sein, sondern kann Kurven beschreiben, etwa um Spiralen entstehen zu lassen (Kloss 2010, S. 108).

Eine weitere wichtige Technik um dreidimensionale Formen mit B-reps zu erzeugen stellt die Verwendung von NURBS dar. NURBS sind Non-Uniform Rational B-Splines, also eine Sonderform der Bézier-Kurve („B-Spline“). Es handelt sich also um Kurven, deren Verlauf zwischen Anfangs- und Endpunkt nicht als eine Gerade interpoliert wird, sondern denen unter Verwendung von gewichteten Kontrollpunkten eine Krümmung im Raum gegeben wird (siehe Abbildung 11).

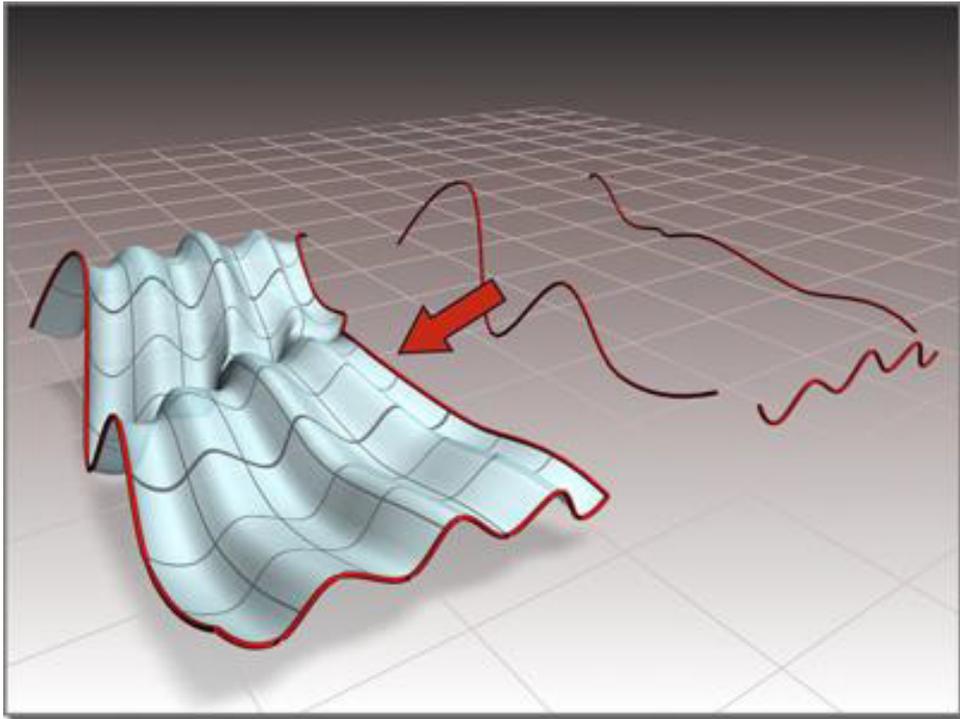


Abbildung 11: Mit NURBS können im 3D-Raum komplexe Formen modelliert werden. Kontrollpunkte (hier nicht abgebildet) bestimmen die Krümmung der Kurven. Quelle: http://www.3dmax-tutorials.com/graphics/il_nurbs_sweep2.jpg

Mit NURBS können fast alle komplexen Formen dargestellt werden. Somit haben sie auch für geologische Modellierungs- und Visualisierungsanforderungen eine große Bedeutung, wie Neumann et al. (2006) berichten. Sie verwenden NURBS, um die Datenlage zwischen Bohrkernen zu interpolieren und erstellen so die Unter- und Oberseiten der Volumenkörper.

Das Problem an B-reps ist jedoch, dass das Innere eines Objekts nicht beschrieben werden kann. Es wird schließlich nur die Außengrenze des Volumenkörpers definiert. Der Körper selber kann also nicht mit Attributdaten, wie Dichtemessungen, Material oder ähnlichem, versehen werden. Trotzdem hält beispielsweise Gold (2008) gerade B-reps für besonders viel versprechend, vor allem in Verbindung mit der unregelmäßigen Zellzerlegung (irregular Cell Decomposition), bei denen also ein Objekt nicht in immer gleiche „Voxel“ derselben Form und Größe zerlegt werden, sondern in unregelmäßige Teile – in diesem Fall Tetraeder – unterteilt wird. Er zählt die beiden Methoden (B-rep und unregelmäßige Zellzerlegung) zu den wichtigsten für die Modellierung von 3D-Objekten (Gold 2008). Ledoux&Gold (2007) stellen ein

Konzept namens Augmented Quad Edge (AQE) vor, dass das B-rep-Konzept mit dem der unregelmäßigen Cell Decomposition verbindet. Dabei werden die 3D-Objekte in Delaunay-Tetraeder zerlegt, anstatt nur die Außengrenzen in Delaunay-Dreiecke zu unterteilen, wie es bei einfacheren B-rep-Methoden der Fall wäre. Jedes einzelne der Tetraeder wird aber durch die B-rep-Methode (hier Quad edge) beschrieben. Gleichzeitig wird das Gegenstück der Dreiecke, das Voronoi-Diagramm, als Polyeder im dreidimensionalen Raum gespeichert. Für weitere Erklärungen zur Verbindung einzelner Zellen und der Navigation zwischen den Zellen siehe Ledoux&Gold (2007). Ledoux&Gold (2007) streichen besonders die Bedeutung des Modells für die Geologie heraus. Ähnlich den aus TIN modellierten Oberflächen können mit Hilfe der Tetraeder kontinuierliche 3D-Körper modelliert werden, in denen kontinuierliche Phänomene wie Goldgehalt in Prozent oder Salinität wie bei Rasterdaten enthalten sein können. Während also die Delaunay-Tetraeder zur Konstruktion des 3D-Körpers herangezogen werden, sind die Voronoi-Diagramme für Operationen wie Interpolationen zu gebrauchen. Außerdem können die jeweiligen Unterteilungen des 3D-Raumes mit einer Vielzahl von Attributen ausgestattet werden, was die Methode für Anwendungen aus den unterschiedlichsten Bereichen attraktiv macht. Das AQE zeichnet sich durch extreme Flexibilität aus. Es ist leicht veränderbar: Wenn Teile eines Modells gelöscht oder hinzugefügt werden müssen, kann dies lokal geschehen, ohne das gesamte Modell neu zu berechnen. Allerdings braucht dieses Modell sehr viel Speicherplatz, da jeder Tetraeder zusammen mit seinem Gegenstück, dem Voronoi-Polyeder, gespeichert wird. Jede Zelle wird somit durch 144 Zeiger repräsentiert. Die Konstruktionsoperatoren konnten aber nicht vollständig für AQE definiert werden (Gold 2008). Euler'sche Operatoren sollten den Weg für eine Implementierung des AQE in CAD-Systemen frei machen (Boguslawski&Gold 2009). Um diese Operatoren zu definieren, stellen Gold&Boguslawski (2010) aufbauend auf dem AQE-Modell die neue Dual Half Edge-Struktur vor. In der neuen Struktur werden Operationen für die Konstruktion erleichtert und der benötigte Speicherplatz wird reduziert. Anwendung soll diese Methode besonders in jenen Fällen finden, in denen sowohl Topologie als auch Geometrie von Bedeutung sind, wie zum Beispiel im Katastrophenschutz bei der Berechnung von Fluchtwegen aus Gebäuden.

Topologien stellen derzeit eine große Herausforderung bei der Modellierung von 3D-Körpern dar, wie auch der nächste Abschnitt dieser Arbeit zeigen wird.

2.2 Topologische Probleme

Es gibt viele offene Fragen, die topologische Analyse von 3D-Objekten in einer Szene betreffend, von denen hier einige untersucht werden sollen, die für geologische Fragestellungen von Bedeutung sind. Die 3D-Modellierung in der Geologie unterscheidet sich in einigen Punkten grundlegend von der Modellierung von beispielsweise Stadtmodellen. Bei geologischen Körpern handelt es sich um Festkörper, die eigenständige Einheiten, wie Schichten, Linsen, etc. darstellen. Die räumliche Lage dieser Festkörper kann entweder geometrisch beschrieben werden, also mit ihren absoluten x-, y- und z-Koordinaten; oder sie kann topologisch beschrieben werden, also durch ihre relative Lage, den Nachbarschaftsbeziehungen zu anderen Objekten (Bode et al. 1994). Die geologischen Einheiten sind zwar prinzipiell voneinander zu trennen, sie berühren sich aber, liegen nahtlos aneinander an. Das bedeutet beispielsweise, dass die Untergrenze der einen Schicht zugleich die Obergrenze der darunterliegenden Schicht ist. Bei der Modellierung muss es aber möglich sein, ein Objekt von einem anderen abzugrenzen – denn es handelt sich ja um zwei *verschiedene* Schichten – auch wenn die Grenze der beiden Objekte *dieselbe* ist. Diese Art von Berührung, die in der Geologie eine entscheidende Rolle spielt, zu beschreiben ist keine einfache Aufgabe. Dies spielt vor allem bei der Analyse eine wichtige Rolle. Die meisten 3D-GIS von heute sind dementsprechend im Prinzip einfache 3D-Viewer mit ein paar Sonderfunktionen (Poupeau 2008, S. 19). Das liegt an der Komplexität der Aufgabe, ein durchgängiges Datenmodell zu erstellen, das sowohl Geometrie als auch Topologie umfasst, und das erst gewisse Analysen und Untersuchungen zu den Beziehungen der Objekte untereinander ermöglichen würde (vgl. Poupeau 2008, S. 23).

Kurata (2010) beschäftigt sich eingehend mit dem Thema der Topologien im dreidimensionalen Raum. Er stellt das 9-contact-Modell vor, das im Prinzip auf dem 9-intersection Modell von Egenhofer (1991) aufbaut. Das 9-intersection-Modell ist eine neunfache Matrix, die alle erdenklichen topologischen Beziehungen zweier

Objekte (Punkt, Linie, Fläche) in 2D beschreiben kann. Sie wird in leicht erweiterter Form von der OGC in ihren Spezifikationen zu den SimpleFeatures verwendet (vgl. OGC 2006). Es ist also ein universell anerkanntes Konzept, das in einem internationalen Standard Verwendung findet. Deswegen macht es besonders Sinn, auf dieses Modell aufzubauen, und herauszufinden, inwieweit die angewendete Methode auch in drei Dimensionen funktioniert.

Exkurs: Das 9-intersection-Modell

Das 9-intersection-Modell dient zur Beschreibung räumlicher Beziehungen zwischen Punkten, Kurven und Flächen. Es ermöglicht also topologische Untersuchungen. Das 9-intersection Modell ist die Weiterentwicklung eines vierfältigen Modells, das die Überschneidungen von Rand und Innenfläche geometrischer Objekte untersuchte. Dieses wurde um das Äußere der jeweiligen Objekte ergänzt (OGC 2006), so dass eine 3x3-Matrix mit neun Möglichkeiten entstand.

Um zwei geometrische Objekte im topologischen Sinne zu vergleichen, werden paarweise die Überschneidungen zwischen Innerem, Rand und Äußerem getestet. Die Ergebnisse werden in die Überschneidungsmatrix eingetragen. Inneres, Rand und Äußeres werden im zweidimensionalen Raum wie folgt definiert: Der Rand eines geometrischen Objekts ist immer eine Menge geometrischer Objekte der nächst geringeren Dimension. Der Rand eines Polygons besteht also aus seinen Ringen. Der Rand einer Kurve besteht aus ihren zwei Endpunkten. Eine geschlossene Kurve hat keinen Rand. Auch der Rand eines Punktes ist die leere Menge, usw. Für dieses Modell werden nur geschlossene Objekte in Betracht gezogen, sodass immer das Innere diejenige Punktmenge ist, die übrig bleibt, wenn man die Randpunkte entfernt. Das Äußere sind alle Punkte, die sich weder im Inneren noch im Rand des Objektes befinden. Nicht alle Fälle von Überschneidung, die im 9-intersection-Modell darzustellen möglich sind, entsprechen einem Fall, für den es auch in der natürlichen Sprache ein Äquivalent gibt. Egenhofer (1991) hebt daher zum Beispiel acht grundsätzliche topologische Verhältnisse hervor. In den Spezifikationen vom OGC (2006) werden fünf solcher Beziehungen näher definiert, die für GIS-Anwender besonders wichtig sind: Innerhalb, Berührung, Überlappung, Überkreuzung und Getrenntheit.

Die fünf bzw. acht topologischen Beziehungen des 9-intersection-Modells auf dreidimensionale Körper ist aber nicht ohne Weiteres möglich, da es sich bei 3D-Figuren, zumal in der Geologie, meist um Festkörper handelt. Diese können sich also von Natur aus gar nicht überschneiden und sind immer unverbunden (Kurata 2010). Obwohl sie also getrennt vorkommen, kann es aber sein, dass die Objekte sich berühren, also in Kontakt miteinander stehen, und dann kann es von Interesse sein, auf welche Weise sie sich berühren, das heißt zum Beispiel, wo sie sich berühren. Im Prinzip wird für die Erstellung der Kontaktverbindungen einfach die Überschneidung beim 9-intersection-Modell durch Kontaktverbindungen ersetzt, da ein starker Zusammenhang zwischen dem topologischen und dem Kontakt-Modell besteht. Die Anzahl der so abgeleiteten Beziehungen zwischen zwei Objekten wird allerdings schnell unübersichtlich (1051 Möglichkeiten für Fläche-Fläche-Beziehungen) wenn man sich nicht auf bloß konvexe Formen beschränkt, das heißt solche Formen die nicht im Raum gekrümmt sind. Kurata (2010) stellt zudem heraus, dass noch zu erforschen sei, ob überhaupt alle Möglichkeiten von räumlichen Beziehungen zwischen zwei Objekten betrachtet wurden. Er stellt außerdem die Vermutung auf, dass Schlussfolgerungen im Kontaktmodell ganz anders verlaufen werden als im topologischen Modell. Folgendes Beispiel wird aufgeführt, um diese Vermutung zu untermauern. Im topologischen Modell *kann gesagt werden: Wenn Objekt X das Objekt Y gänzlich enthält, und das Objekt Y das Objekt Z gänzlich enthält, dann enthält X auch Z.* Diese Folgerung ist nicht einfach auf das Kontaktmodell übertragbar. Denn wenn X das Objekt Y gänzlich berührt, kann Y nicht mehr Z berühren.

Poupeau & Bonin (2006) hingegen gehen einen anderen Weg, um nicht auf das 9-intersection-Modell zurückgreifen zu müssen. Sie entwerfen ein neues Datenmodell, Cristage, dem sie neben den geometrischen Informationen über die Objekte noch einen logischen Layer hinzufügen, der semantische Informationen enthält und unter anderem dazu dient, Berührungen zwischen Objekten zu beschreiben. Topologische Modelle wie das 9-intersection-Modell sind nach Ansicht von Poupeau & Bonin (2006) nicht ausreichend um geologische Strukturen sinnvoll zu beschreiben. Aussagen über mehr als zwei Objekte sind mit diesem Modell sehr schwer zu machen, werden in der Geologie aber häufig benötigt, wie beispielsweise beim

Einsturz einer Höhle, der einen „Krater“ durch mehrere geologische Schichten reißt. Außerdem sind in die rein topologischen Modelle keine semantischen Informationen einfügbar. Diese sind aber notwendig, beispielsweise zur Unterscheidung von Hangendem und Liegendem, also dem oberen und dem unteren Teil einer geologischen Formation. Die Idee hinter dem Datenmodell Cristage (Poupeau & Bonin 2006) liegt nun darin, einen logischen Layer zu konstruieren, der die Berührungen von Flächen anhand eines kristallographischen Netzes beschreibt.

Dieses neue Datenmodell ist also aus zwei Schichten aufgebaut. Die erste Schicht umfasst die Beschreibung der Geometrie. Dies soll durch so genannte B-reps (Boundary representations) geschehen. Diese eignen sich sehr gut für die Visualisierung der 3D-Körper, sowie für die Datenspeicherung. Sie sorgen für Kohärenz und die Objekte bleiben beweglich. Andererseits soll das Innere der 3D-Objekte durch Voxel (volume element) beschrieben werden, um Volumenberechnungen und Ähnliches durchführen zu können. Dabei handelt es sich, ähnlich der Zerlegung einer Oberfläche in ein Dreiecksmosaik, um die Zerlegung eines Körpers in Tetraeder (siehe 3 Modellieren). Der zweite Layer ist ein rein logischer Layer, der mit dem Geometrielayer verlinkt ist und semantische Informationen über die Objekte enthält. Er wird durch eine Vereinfachung der geographischen Objekte zu kristallographischen Formen abgeleitet. Dann werden die einzelnen Seiten der Kristalle indiziert. So können zum Beispiel in der kristallographischen Form für ein Haus verschiedene Indizes für Dach und Wand vergeben werden. Die Indizes ermöglichen also eine Einbindung von semantischen Informationen. Auf diese Weise wird eine Charakterisierung der Hauptteile von geographischen Objekten möglich. Dies erlaubt nun eine Beschreibung, welche Teile eines Objektes mit welchen anderen in Berührung stehen. Dazu werden nun Kontaktobjekte eingeführt. Diese ebenfalls rein logischen, unsichtbaren Objekte werden zwischen die sich berührenden Objekte „geschoben“, um dem System zu vermitteln, dass sich die beiden Objekte berühren. Enthalten die Kontaktobjekte semantische Funktionen, wie beispielsweise die Schichtgrenzen zwischen zwei geologischen Formationen, so werden sie Interface-Objekte genannt. Die explizite Ergänzung der Relationen unter den 3D-Objekten hat den Vorteil, dass selbst bei einer ungenauen Datenlage die Berührungen klar geregelt werden können. So kann es

beispielsweise bei der Modellierung von Häusern dazu kommen, dass sie nicht direkt an den Untergrund anschließen. Sie „schweben“. Dies kann an Inkohärenzen bei der Modellierung oder in der Datengrundlage selbst liegen (Poupeau 2008, S. 75). Solche Probleme sollen mithilfe von Crustage ausgeglichen werden können. Es lassen sich auf diese Weise auch komplexe geologische Formen wie Störungen darstellen, ohne Gefahr zu laufen, die Kohärenz zwischen den Daten zu verlieren. Geologische Schichten, die zusammengehören, sind auch durch Störungen hindurch als ein Volumenkörper definierbar.

Es ist zusammenfassend festzuhalten, dass Fragen beispielsweise nach der Topologie, die in der zweidimensionalen Welt schon weitgehend gelöst wurden, sich im 3D-Bereich erst stellen und alte Modelle in Frage stellen könnten. Um bereits entwickelte Modelle weiterhin verwenden zu können, müssen sie erheblich erweitert und auch verändert werden. Die Erforschung von Modellen, die zur Beschreibung von Beziehungen zwischen geologischen Körpern gebraucht werden können, befindet sich noch am Anfang. Solche grundlegenden Probleme sind zu lösen, will man eines Tages Antworten auf Fragen finden wie: „Befindet sich A auf B?“. Diese Fragen betreffen die Anordnung von Objekten im Raum (Kurata 2010), also auch insbesondere die Geologie, wie in Bereichen der Reihenfolge von Schichten, die Berührungseigenschaften der jeweiligen Schichtgrenzen usw. Trotz dieser ungelösten Problemstellungen finden sich einige Ansätze der 3D-Modellierung mit geologischen Fragestellungen. Einige Beispiele davon werden im nächsten Abschnitt näher beleuchtet.

2.3 Stand der Technik: wissenschaftliche 3D Software in der Geologie

Es gibt viele verschiedene Umsetzungen der bisher erläuterten Theorien zur Darstellung von 3D-Objekten im geologischen Raum. Neben kommerziellen Anbietern haben auch Forschungsinstitute eigene Lösungen in Betracht gezogen. Im Folgenden werden besonders weit verbreitete Programme sowie für die Geologie interessante Ansätze vorgestellt, um den derzeitigen Stand der Technik zu umreißen, und für die Geologie wichtige Funktionen zu identifizieren. Die meisten GIS und

andere Programme, die heute für die geologische Modellierung verwendet werden, bedienen sich der B-rep-Methode zur Darstellung dreidimensionaler Körper.

2.2.1 Plug-Ins für GIS-Programme

Weit verbreitet sind Programmiererweiterungen, so genannte Plug-ins, die auf eine bereits bestehende Software aufbauen, und diese um Funktionen bereichern, die speziell im geologischen Kontext von Bedeutung sind. Aufgrund der weiten Verbreitung der Esri-Produkte, wird häufig auf die 3D-Funktionalität von Esris ArcScene zurückgegriffen. ArcScene speichert die 3D-Geometrien, die aus Punkten, Linien und Polygonen bestehen, nicht in einer herkömmlichen Geodatenbank, sondern als Multipatch. Multipatches verhalten sich zwar wie die anderen Features, können aber auch echte 3D-Situationen widerspiegeln, indem sie beispielsweise senkrechte Wände oder Überhänge darstellen können. Prinzipiell unterstützt ArcScene die drei Datentypen TIN, Grid und dreidimensionale Vektordaten. TINs sind aus Dreiecken (2D) zusammengesetzte Oberflächen. Die Eckpunkte der Dreiecke beinhalten z-Werte für die Höhenangabe. So können dreidimensionale Flächen ohne Mächtigkeit (auch 2,5D genannt) dargestellt werden. Grids hingegen sind in Zeilen und Spalten geordnete Zellen, denen jeweils ein Höhenwert zugeordnet wird. Auch auf diese Weise entstehen 2,5D-Flächen im Vektorformat. Die Kanten und Knoten der Vektorgraphik bleiben dem Anwender aber verborgen und werden stets mithilfe von graphischen Overlays (Texturen) dargestellt, die von ArcMap selbst erzeugt werden, wie zum Beispiel ein Graustufenbild, das die Höhenwerte symbolisiert. Dreidimensionale Vektordaten erlauben die Darstellung von Mächtigkeiten oder Volumenkörpern. Der Aufbau entspricht eigenen Vermutungen zufolge den Richtlinien des OGC, die in den SimpleFeatures-Spezifikationen festgelegt wurden, da diese laut Esri in seinen GIS-Produkten voll umgesetzt sind. Der Import von 3D-Objekten aus anderen Formaten (3ds, wrl, skp und Weitere) ist möglich. ArcScene verfügt auch über 3D-Generatoren wie der Extrusion. Die Werkzeuge *Extrude* und *Extrude between* erlauben die Erstellung von echten Volumenkörpern aus 2,5D-Oberflächen im Vektorformat.

In diesem Abschnitt soll ein Spektrum an Möglichkeiten aufgezeigt werden, wie Plug-ins umgesetzt werden können. Als erstes wird das Programm BoreIS vorgestellt, dass

ohne viel Aufwand die 3D-Darstellung von ArcScene für sich nutzt, indem hauptsächlich neue Dialogfenster mit speziellen Funktionalitäten unter der ArcScene-Oberfläche genutzt werden. Am anderen Ende des Spektrums ist GeoRover anzusiedeln, das für die Anwender wie eine eigenständige Software aussieht, aber für Darstellung und Datenmanagement auch auf ArcScene zurückgreift. Da die Erweiterungen sich in das ArcGIS von Esri nahtlos einfügen und dem Benutzer erlauben, unter der ArcGIS Oberfläche zu arbeiten und somit auch die anderen Funktionen des GIS zu nutzen, sind die Plug-Ins in dieser Arbeit der Modellierungssoftware zugeordnet, obwohl sie teilweise auch mächtige Visualisierungskonzepte beinhalten.

BoreIS

McCarthy & Graniero (2006) stellen ein GIS-basiertes System für das Management von Bohrerndaten und deren Visualisierung in 3D vor. Borehole Information System (BoreIS) ist eine Erweiterung zu Esris ArcScene. Den Geologen soll ermöglicht werden, in möglichst bekannten Datenformaten zu arbeiten, um aufwendige und zeitraubende Datenaufbereitungen zu vermeiden, was einen Zeitgewinn für die Analyse der Daten bedeutet. Außerdem wird Wert darauf gelegt, dass die Bedienung des Programms möglichst einfach ist und auch Geologen, die keine GIS-Experten sind, sich der mächtigen Tools eines solchen bedienen können. BoreIS ist ein High-level-Programm, das möglichst viel automatisieren soll. Es befähigt ArcScene darin, Daten einzulesen, die in unter Geologen weit verbreiteten Formaten vorliegen (z.B. Excel-Tabellen), von der Standardversion des GIS jedoch nicht unbedingt so unterstützt werden, dass sie eine direkte Interpretation ermöglichen. BoreIS stellt darüber hinaus noch einige ausgewählte Funktionen für die geologische Analyse zur Verfügung, während viele der komplexen GIS-Funktionen automatisiert ablaufen. Das Programm richtet sich vornehmlich an kleine Unternehmen, die sich andere, bereits existierende Programme in dieser Sparte nicht leisten können, deren umfangreiche Funktionen sie aber auch nicht in ihrer Gesamtheit gebrauchen können. McCarthy & Graniero (2006) zufolge wurde BoreIS als Plug-In für die Esri-Software entwickelt, weil diese für viele Anwender zur Verfügung steht und auf eine gewohnte Benutzeroberfläche aufbauen kann. Das Lernen der Bedienung des Plug-ins wird erleichtert und dem User stehen alle Funktionen von ArcScene zur Verfügung. Zudem

brauchen die Entwickler sich nur auf die neuen Funktionen zu konzentrieren, um die ArcScene erweitert werden sollte. Der erste Schritt in der Benutzung von BoreIS besteht darin, durch einen interaktiven „Wizzard“ ein Shapefile aus den Excel-Daten zu erstellen. Dieses Shapefile wird sodann automatisch in ArcScene geladen und kann mit den vorhandenen Tools oder den speziellen BorIS-Funktionen bearbeitet werden. Diese umfassen vor allem SQL-Abfrage-Werkzeuge, die entweder nach Kategorien (Zeichenketten) oder nach Schwellenwerten (Numerische Werte) bestimmte Features auswählen und entsprechende Farben (für „ausgewählt“ oder „nicht ausgewählt“) zuordnen. Durch die Integration der Symbolauswahl in das Abfragefenster soll dem User erspart werden, in ein anderes Menu zu wechseln um Farbe etc. festzulegen. Darin liegt ein Vorteil zum schon in der Grundversion von ArcScene *Select by Attribute*-Befehl. In einem Query-Builder können mehrere dieser einfachen Abfragen kombiniert werden und so erhält man komplexe Abfragen. Mithilfe der eingebauten Symbol-/Farbauswahl können Bohrlochdaten schnell visualisiert werden. In der 3D-Ansicht von ArcScene können dann Muster durch die räumliche Lage der eingefärbten Features identifiziert werden. Des Weiteren werden einige Tools von ArcScene einfacher zugänglich gemacht: Die Einstellung der Überhöhung und die Interpolation eines Rasters aus den ausgewählten Daten einer vorhergehenden Abfrage werden auf die Toolbar gesetzt.

BoreIS ist also ein Programm, das sehr eng mit den Funktionen, Dateiformaten und Visualisierungstechniken von ArcScene verknüpft ist. Ein typisches Motiv bei der Entwicklung von geologischen Modellierungsprogrammen, wie auch BoreIS, ist die einfache Bedienbarkeit und Eingrenzung der Kosten. BoreIS erkennt auch die Wichtigkeit der Darstellung von geologischen Daten im dreidimensionalen Raum an, und unterstützt diese, wenn auch nur stark eingeschränkt, anhand von Punktdaten. BoreIS verwendet als Dateiformat das zwar sehr weit verbreitete aber doch proprietäre Shape-Format von Esri. Dadurch ist die Interoperabilität nur zu einem gewissen Grad gegeben, und der Datenaustausch auf die von Esri in der Exportfunktion von ArcScene vorgegebenen Formate beschränkt.

GeoRover

Das Plug-in GeoRover basiert, ähnlich BoreIS, auf der Idee, die umfangreichen GIS-Funktionen von ArcMap auch für geologische Daten nutzbar zu machen. Auch wird das Datenmanagement in Formaten, die speziell in der Geologie vorkommen, so gestaltet, dass die Daten in Esri-Softwarepaketen verwendet werden können.

GeoRover unterstützt besonders das SEG Y-Format für seismische Profile und den Log ASCII Standard (LAS), der üblicherweise in der Öl- und Gasindustrie für Bohrlochdaten verwendet wird (Gocke et al. 2010). Hinzu kommen umfangreiche Visualisierungsfunktionen für die 3D-Darstellung der Daten. Das Display kann in bis zu vier Ansichten aufgeteilt werden. Ein Fenster zeigt die perspektivische Darstellung, die anderen drei zeigen die orthographische Sicht parallel zu den drei räumlichen Achsen. In die perspektivische Ansicht kann ein Heads-up-Display eingeblendet werden, unter anderem mit einem Kompass, der auch Neigungen anzeigt. Die Darstellung der 3D-Daten beruht auf dem Prinzip, dass sich bei Perspektivänderungen die Kamera bewegt und nicht etwa das Objekt rotiert wird. So können Blickwinkel leicht aus allen erdenklichen Positionen angenommen werden (Gocke et al. 2010). Für die 2,5D Höhenmodelle, die die einzelnen Schichtgrenzen darstellen, können Farbverläufe errechnet werden. Diese Methode spart gegenüber Overlays Speicherkapazitäten (Gocke et al. 2010). Nach der gleichen Methode können auch Höhenlinien berechnet und dargestellt werden. In die Höhenmodelle können seismische Profile und andere Profilschnitte „eingehängt“ werden. Die Lage der Profile ist in dieser Sichtweise sehr gut zu erkennen. Bohrlochdaten werden als senkrechte Bänder dargestellt, die bis zu drei verschiedene Datenreihen parallel darstellen können. Geologische Störungen werden als senkrechte „Zäune“ dargestellt, die nach unten fortgesetzt werden. Ein spezielles Tool ermöglicht die Erstellung von Volumenkörpern, indem zwei oder mehrere, die Schichtgrenzen darstellende, Flächen zusammengefasst werden. Jetzt können auch Volumensberechnungen für die so erstellten Objekte durchgeführt werden. Da als Grenzen auch Schnittflächen oder Ähnliches verwendet werden können, ist die Erzeugung von Blockbildern und Profilen möglich. Komplexe geologische Strukturen können mithilfe von einfachen Objekten, wie Würfeln oder Kugeln, realisiert werden, die immer weiter verfeinert werden. Dazu greift der Nutzer direkt die Vertices des Dreiecksnetzes an, verschiebt sie und fügt neue hinzu.

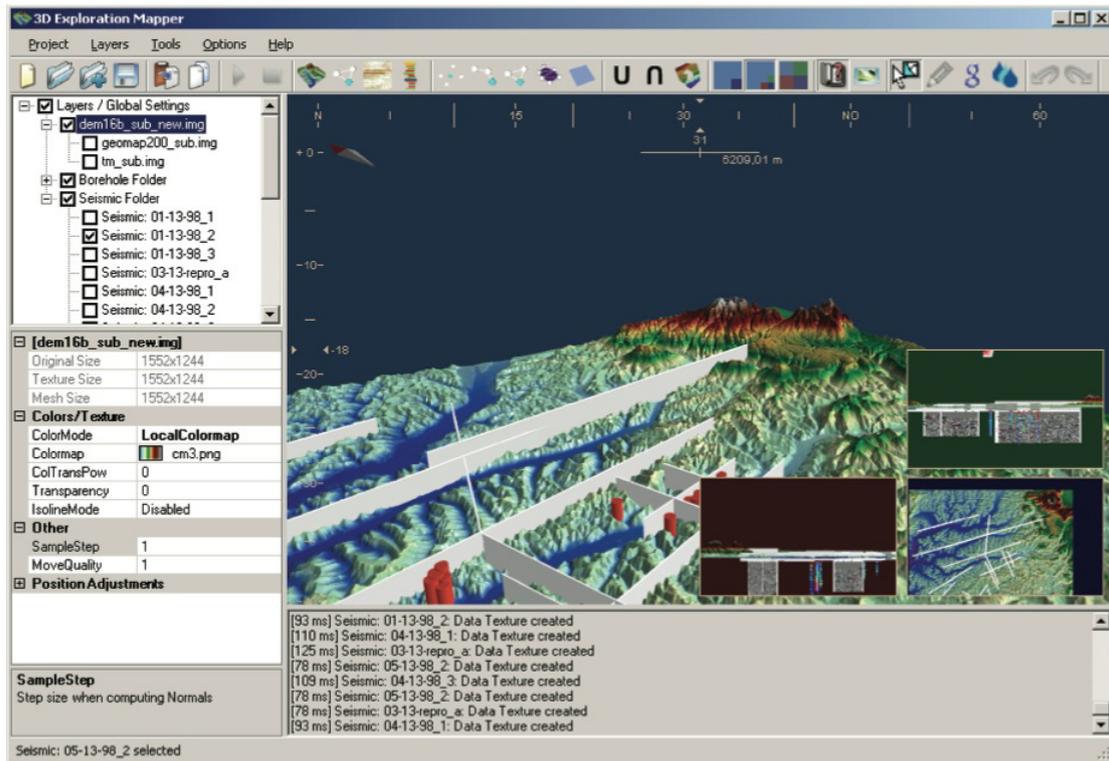


Abbildung 12: Die Benutzeroberfläche in GeoRover. Quelle: Gocke et al. 2010

GeoRover erweitert die Funktionen der Esri-Software erheblich. Da eine eigene Benutzeroberfläche verwendet wird, scheint GeoRover fast wie ein eigenständiges Programm (Abbildung 12), und die Nutzung des 3D-Renderers von Shape-Dateien verwaltet, dem Esri-eigenen Format. Die Bindung an Esri-Software wird allein hierdurch schon augenscheinlich.

2.2.2 Eigenständige Softwarelösungen zur 3D-Modellierung geologischer Formen

Viele geologische Institute setzen heute 3D-Software ein. Zu den leistungsfähigsten aber auch kostspieligsten Varianten zählen Vulcan, Gocad und Geomodeller. Eine ausführliche Liste ist beispielsweise auf den Internet-Seiten der Martin-Luther-Universität Halle-Wittenberg zu finden¹. In dieser Arbeit soll vor allem das Programm Gocad stellvertretend für kommerzielle Software, die speziell auf die Bedürfnisse der Geologie ausgelegt ist, näher beleuchtet werden. Bei diesem Typ von Informationssystem handelt es sich um eigenständige Software, die eine sehr speziell

¹ http://www.3d-geology.de/software/geology_and_mining/?lang=en

auf geologische Fragestellungen zugeschnittene Funktionalität besitzt, und daher oft auch in Bereichen der Öl- und Gasindustrie eingesetzt wird. Gocad und Geomodeller sind besonders in Australien, Europa und Nordamerika beliebt (Kessler et al. 2008). Aber staatliche und geologische Forschungsinstitutionen arbeiten zum Teil auch an eigenen Lösungen und entwickeln Programme, die ihren Aufgaben noch besser entsprechen. Auf eine jahrelange Geschichte kann in diesem Fall das British Geological Survey zurückblicken. Deswegen wird sein Programm GSI3D in dieser Arbeit ebenfalls vorgestellt, um den Blickwinkel einer Einrichtung näher kennen zu lernen, die mehr dem Umfeld dieser Masterarbeit entspricht, als die Öl- und Gasindustrie.

Gocad

Das Akronym Gocad steht für *Geological Object Computer Aided Design*. Das Projekt für die Softwareentwicklung wurde im Jahre 1989 an der School of Geology der Universität de Nancy begonnen. Die Erfahrungen der Forscher flossen in die Software ein, die nun kommerziell vermarktet wird. Das Forschungsprojekt wird fortgeführt und von einem Konsortium unterstützt, das sowohl Universitäten und Forschungsinstituten als auch Firmen offen steht. Zu den Konsortiumsmitgliedern zählen daher einerseits Geosciences Australia, BGR Hannover, der British Geological Survey, die Universität Wien und viele mehr, als auch namhafte Unternehmen aus dem Rohstoffbereich wie BHP Billiton, BP, Shell oder Exxon-Mobil. Da der 3D-Ansatz von vornherein im Fokus des Forscherteams lag, gehört schon zur Grundausstattung des Softwarepakets ein 3D-Viewer. Das Programm ist modular aufgebaut, sodass viele weitere Funktionen je nach Bedarf hinzu gezogen werden können. Die Software stellt höchste Anforderungen an die Hardware und verwendet OpenGL. Die Spezialisierung auf geologische Fragestellungen schlägt sich auf die Technologien nieder, die in der Software angewendet werden. Beispielsweise erlaubt die so genannte Multi-Z-Oberflächenmodellierung das Konstruieren von komplexen Strukturen zum Beispiel im Zusammenhang mit Salzdiapieren, die sich oft pilzförmig vertikal durch mehrere geologische Schichten ziehen (Abbildung 13).

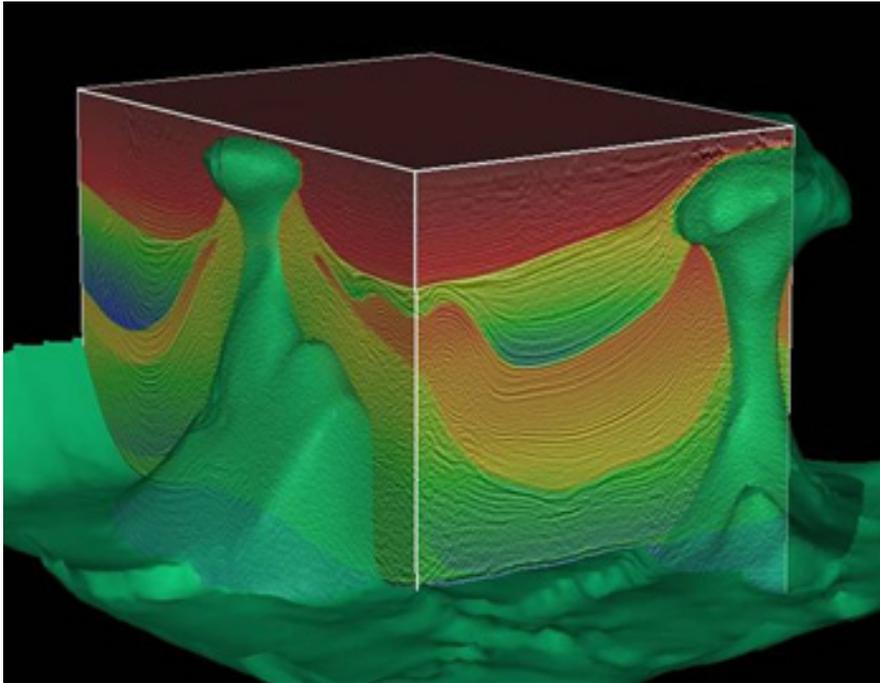


Abbildung 13: 3D-Darstellung von Salzdiapieren in Gocad. Quelle: <http://www.pdgm.com/products/seismic-processing-imaging/gocad-velocity-modeling.aspx>

In Gocad wird eine spezielle Form von B-rep verwendet, eine Datenstruktur namens G-Maps (generalized maps) (Ledoux, H. & C. Gold 2007). Jede topologische Zelle wird dabei in ihre nächst kleineren Komponenten zerlegt. Dabei werden die jeweiligen topologischen Beziehungen erhalten. Dieses Verfahren ist sehr aufwändig, da jede Zelle von geringerer Dimensionalität in Zellen von höherer Dimensionalität gespeichert werden muss. Wenn sich also fünf Kanten in einem einzigen Punkt treffen, wird dieser Punkt fünfmal, nämlich in allen Kanten, gespeichert.

GSI3D

Im Vereinigten Königreich arbeitet der British Geological Survey (BGS) schon seit mehr als 30 Jahren an einer eigenen Lösung, geologische Daten dreidimensional am Computer zu verarbeiten (vgl. Kessler et al. 2008). Zwar gibt es reichlich Software, die vor allem im Bereich der Öl- und Gasindustrie angewendet wird, doch entspricht diese oft nicht den Anforderungen von Wissenschaftlern. Speziell in Hinsicht auf akzeptierte Datentypen, Quantität aber auch Qualität der Daten, die sich im Bereich

der geologischen Forschung aus vielen verschiedenen Quellen zusammensetzen, lassen diese Programme viele Wünsche der Geologen offen.

Für den BGS war die Entwicklung von geologischen 3D-Modellen ein logischer Schritt, der über kurz oder lang die traditionelle geologische Karte ersetzen soll (Kessler et al. 2008). Dafür muss die Software aber einige Bedingungen erfüllen. Einerseits soll die Verwendbarkeit (Usability) nicht nur Computerexperten vorbehalten sein, sondern auch die Geologen selbst sollen damit umgehen können. Den Experten aus der Geologie soll es möglich gemacht werden, die Geologie des Landes effektiv modellieren zu können. Daraus folgt, dass bereits bestehende Modelle einfach um neue Daten, Updates etc. erweitert werden können. Die Modellierung soll „Wissensgesteuert“ (nach dem „knowledge-driven approach“, Kessler et al. 2008, S.1312) sein.

Andererseits sollen auch Laien in die Lage versetzt werden, einige Funktionen, vornehmlich die Visualisierung betreffend, ausführen zu können. Außerdem soll die Software kostengünstig sein, vor allem im Vergleich zu Gocad etc.

Um diesen Ansprüchen zu genügen, wurde am BGS ein eigenes Programm entwickelt: „Geological Surveying and Investigation in 3 Dimensions“ (GSI3D). Seit April 2010 wird die Software in einem Konsortium verbreitet und weiterentwickelt, dem unter anderem der United States Geological Survey (USGS), der Geological Survey of Norway oder der Hong Kong Geological Survey angehören.

GSI3D wird heute routinemäßig von den Geologen des British Geological Survey eingesetzt, besonders um Modelle für andere Institutionen wie Lokalbehörden oder Kommunalverwaltungen zu erstellen. Einen Erfolg sehen die Entwickler des Programms darin, dass dieselbe Datengrundlage verwendet wird, die Geologen seit 200 Jahren verwenden. Selbst wenn reichlich Daten über den Untergrund durch in der Vergangenheit durchgeführte Bohrprojekte usw. vorhanden sind, stammen diese oft aus sehr heterogenen, analogen Quellen (Neber et al. 2006). Das Wissen aus geologischen (Papier-)Karten, Geländemodellen, Borhlochkernen usw. kann in GSI3D eingespeist und daraufhin digital weiterverwendet werden. Daher erfreut sich die Software einer großen Akzeptanz unter den Geologen (Kessler 2008). Ein Dateninput aus vielen verschiedenen Quellen ist also möglich. Topographische Karten können als JPG-Dateien, Digitale Geländemodelle als ASCII-Grid-Dateien eingelesen werden. Bohrlochdaten werden als ASCII-„Tabellen“ (Tabulator als Trennzeichen)

eingebunden. Bereits digitalisierte geologische Karten können als Shape-Dateien eingespielt werden. Das Programm ist offen für Zusatzdaten wie Profile, Codes für die Legenden, chemische Analysen etc.

Ein Nachteil von GSI3D ist, dass bislang keine komplexen geologischen Formationen wie Faltungen und Bruchzonen, Störungen oder umgekehrte Schichtfolgen modelliert werden können. Dies soll nach Kessler et al. (2008) nach einem auf drei Jahre angelegten Forschungsprogramm (also voraussichtlich im Jahre 2011) behoben werden. Aber der große Vorteil des Systems liegt darin, dass auch das stillschweigende Wissen („tacit knowledge“ Kessler et al. (2008) S. 1314) der Geologen in den Modellierungsprozess einfließen kann. Bei der Definition von geologischen Strukturen spielen nicht nur morphologische Regeln eine große Rolle sondern auch die Vorstellung der Geologen, basierend auf deren Erkenntnissen (Neber et al. 2006). Das ist besonders beim modellieren von geologischen Abfolgen in Hinsicht auf die gestaltenden morphologischen Prozesse von Bedeutung. Von einem Computer kann diese Art von Wissen nicht simuliert werden (Kessler et al. 2008).

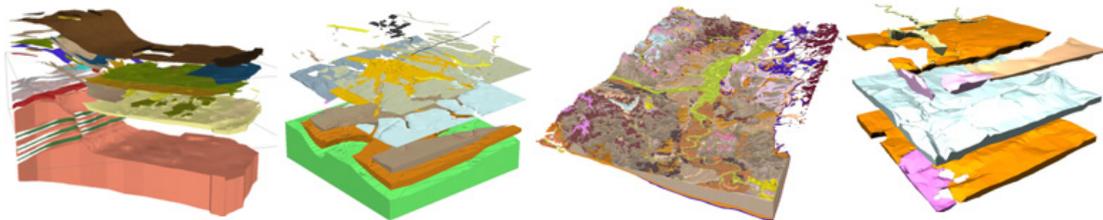


Abbildung 14: Beispiele für Modelle aus GSI3D. Quelle: <http://www.gsi3d.org/index.html>

Die 3D-Modellierung wird in GSI3D wie folgt umgesetzt. Durch Interpolation zwischen den x,y,z-Koordinaten einzelner Knotenpunkte entsteht eine Reihe von übereinander liegenden Triangulated Irregular Networks (TIN, nach Delaunay), die jeweils die Unterseite einer geologischen Einheit repräsentieren. Um Volumenkörper zu erstellen, werden die unmittelbar übereinander liegenden TINs erfasst. An den Grenzen des Projekts (eines Blockbildes zum Beispiel) werden die Zwischenräume der TINs durch senkrechte „Wände“ geschlossen. Es entsteht also eine geschlossene Polygonoberfläche, die die Begrenzung des 3D-Volumens beschreibt. Das entspricht

dem B-rep-Prinzip. GSI3D fasst also die beiden oftmals getrennten Arbeitsabläufe der Interpretation der geologischen Daten und deren Modellierung in einem zusammen (vgl. Neber et al. 2006, S. 63).

GRASS und VoxelGeo als Beispiele für voxelbasierte Modellierungssoftware

Nachdem nun wichtige Vertreter von geologischer Software erläutert wurden, die nach dem B-rep-Prinzip funktionieren, soll noch erwähnt werden, dass sich auch einige Ansätze finden lassen, die eher der Cell Decomposition bzw. der Spatial Occupancy Enumeration zuzurechnen sind. Darunter fällt beispielsweise VoxelGeo von Paradigm. Die Firma verspricht mithilfe dieser Methode, selbst kleinste Details in geologischen Faltungen, Störungen und ähnlichem gut interpretieren zu können. Auch das Open Source GIS GRASS hält neben den Vektorbasierten 3D-Techniken auch Voxel-Funktionen bereit, um auch das Innere von Volumenkörpern mit Attributen füllen zu können. Allerdings führen diese Anwendungen zumindest momentan (nach eigener Einschätzung) eher ein Nischendasein.

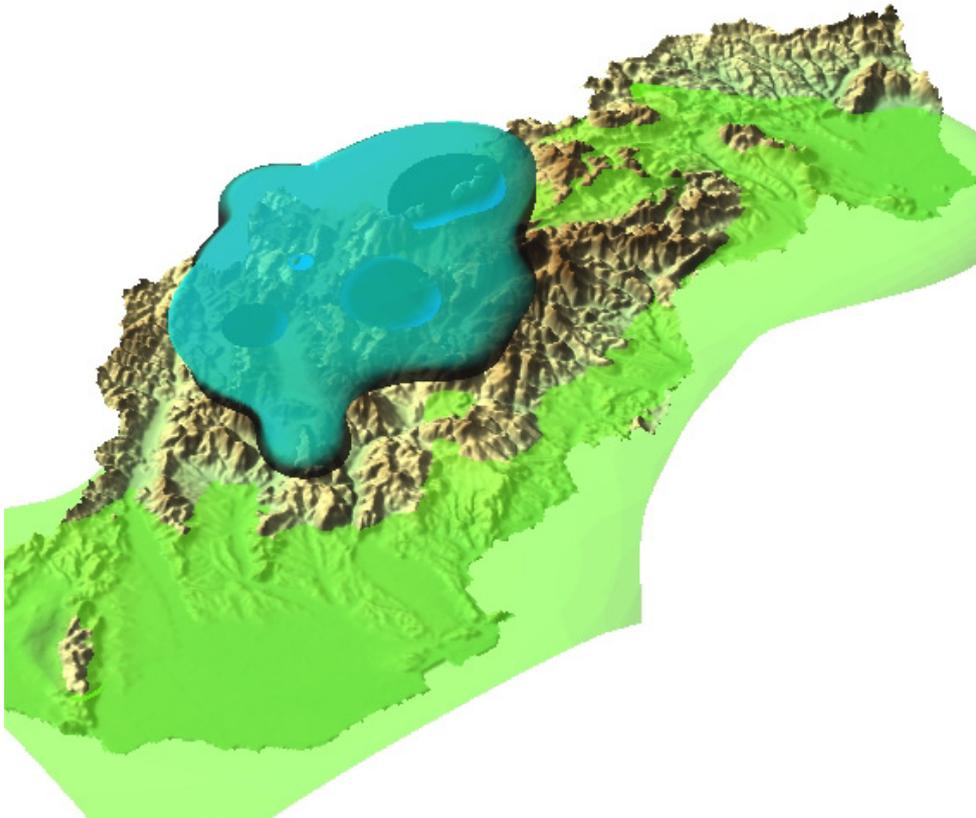


Abbildung 15: Voxel-Darstellung eines Niederschlags-Voxelvolumenkörpers aus GRASS. Quelle: <http://grass.fbk.eu/images/slovakiavol.jpg>

3 Visualisierungsmöglichkeiten mit Web-Technologien

Unter Geovisualisierung wird oft die Verwendung von graphischen Anzeigen verstanden, die es ermöglichen, Daten interaktiv zu erkunden. Durch diese Erkundung können dann Hypothesen aufgestellt, Problemlösungen entwickelt und neues Wissen konstruiert werden (MacEachren 2001, Kraak 2003). Doellner et al. (2003) definieren eine 3D-Karte „als kartenverwandte Darstellung, die raumbezogene Daten und Prozesse auf der Grundlage eines digitalen Geländemodells computergraphisch dreidimensional visualisiert, die Interaktion mit dem Dargestellten ermöglicht und den Aufbau und die Gestaltung der Karte dynamisch festlegt“.

MacEachren (2001) betont explizit die Interaktivität der neuen Medien im Gegensatz zur traditionellen Kartographie und stellt heraus, dass die Kartennutzer dadurch in besonderem Maße dazu befähigt werden, neues Wissen aus Karten aufzubauen. Daneben bleiben aber auch bei interaktiven Webkarten die traditionellen Funktionen der Karte erhalten, wie zum Beispiel Routenplanung, Terraindarstellung, Planung etc. Gut gemachte Karten verbinden ihre Nutzer mit einem Teil der Realität. Ihre abstrakten Symbole geben Einsicht in die georäumlichen Muster und Bezüge, egal ob die Karten im Internet erscheinen oder auf dem Papier (Kraak 2004). Kraak (2004) unterteilt die Funktionen der „neuen“ Karten in Präsentation und Exploration. Die Präsentation teilt er den traditionellen Funktionen der Karte zu. Die Kartographen erstellen mit Hilfe von vorhandenem Wissen eine möglichst aussagekräftige Karte. Dabei muss allerdings beachtet werden, dass die traditionellen Konzepte eines Kartenentwurfs zumindest in Teilen an die neue Situation angepasst werden muss.

Die von Bertin (1967) in seinem Buch *Semiologie Graphique* identifizierten graphischen Variablen für kartographische Darstellungen – Größe, Helligkeit („valeur ... entre le blanc et le noir“ S. 60), Textur, Farbe, Ausrichtung und Form – werden von MacEachren (2001) für die heutige Computerkartographie und die damit einhergehende Darstellung am Bildschirm erweitert, zum Beispiel um Werte für Transparenz oder Schärfe/Unschärfe. Im Gegensatz zur Präsentation verhilft die Exploration dagegen Expertennutzern zu neuem Wissen. Hierbei können die Interaktionsmöglichkeiten sehr gut genutzt werden, im Kontext des Internets werden sie sogar von den Nutzern erwartet, und mangelnde Interaktivität führt dort zum Abbruch der Nutzung einer Karte (Kraak 2004, S.86f.). Keim et al. (2005) betonen im

Gegensatz zu Kraak den Zusammenhang zwischen Präsentation und Erforschung: Es ist die Visualisierung, die den Menschen in besonderem Maße in die Lage versetzt, große Datenmengen zu „erforschen“. Die Präsentation von Daten in einer sichtbaren Form ermöglicht es erst, „Einblicke in die Daten zu gewinnen, Schlussfolgerungen zu ziehen und direkt mit den Daten zu interagieren“ (Keim et al. 2005). So wird auch die hohe Nachfrage an Visualisierungen damit erklärt, dass sie zu besseren Ergebnissen führen können, als allein die automatische Analyse durch Algorithmen, und dass sie eine deutlich höhere Sicherheit der Ergebnisse erzeugt. Die Kombination von automatischer Auswertung großer Datenmengen durch Algorithmen mit der „intuitiven Kraft des menschlichen Geistes“ (Keim et al. 2005) kann sowohl die Qualität als auch die Geschwindigkeit des Datamining erhöhen. Interaktion ist ein wesentlicher Bestandteil von Visualisierungsmethoden im Internet und insbesondere für dreidimensionale Karten. Keim et al. (2005) teilen die verschiedenen Interaktionen nach Einfluss auf das Display ein:

- Navigation
Verändert die Projektion der Daten auf den Bildschirm
- Sichterweiterung
Erlaubt die Veränderung des Detailgrads oder des thematischen Schwerpunkts der Daten
- Auswahl
Zum isolieren von Daten für bestimmte Operationen, wie Hervorheben, Filtern, quantitative Analysen etc.

Im Internet kann frei auf Geodaten und Karten von einer unbegrenzten Anzahl an Nutzern zugegriffen werden. Kraak (2004) nennt deswegen *Interoperabilität* einen der Schlüsselbegriffe dieser Diskussion. Auch MacEachren et al. (2004) sehen eine der größten Herausforderungen in den Austauschmöglichkeiten visualisierter Geodaten über Plattformgrenzen, Hard- und Software, sowie Leute und Orte hinweg. Sie unterstreichen die Notwendigkeit der Bemühungen des OGC und seiner Partner, einheitliche Standards aufzustellen, um den freien Datenaustausch zu ermöglichen. Eine weitere Besonderheit von digitaler und Internetkartographie liegt in der strikten Trennung zwischen der Organisation von Daten und der Nutzung derselben. So kann

zum Beispiel Geographic Markup Language (GML) verwendet werden, um Daten anzubieten. Ein völlig anderes Format, wie Scalable Vector Graphics (SVG), könnte dann für die (2D-)Visualisierung genutzt werden (Kraak 2004). Wie auch in Teil II dieser Arbeit über die Umsetzung eines konkreten Falls deutlich werden wird, spielt die Organisation der Daten eine immense Rolle. Es lagen Geometriedaten als ASCII-Dateien und Shape-Dateien vor, während Overlays und Texturen im JPG-Format vorhanden waren oder erst als PNG-Dateien erstellt werden mussten. Alle Daten sollten in einer einzigen Visualisierung zusammengeführt werden, die im Internet betrachtet und exploriert werden sollte.

MacEachren et al. (2004) warnen vor zwei besonderen Herausforderungen: Die Bedienung der 3D-Karten erfordert eine große Konzentration vor allem während der Eingabe über Tastatur und Maus. Dies kann Nutzer von den Inhalten der Karte ablenken. Des Weiteren sind überladene Darstellungen nur mit extrem hohem, kognitivem Aufwand zu lesen. Werden diese dann noch mit allzu komplexen interaktiven Funktionen ausgestattet, kann das Verstehen der dargestellten Szene schnell erschwert werden (MacEachren et al. 2004). Zwar wird diese Feststellung im Kontext des speziellen Gebrauchs von interaktiver Geovisualisierung im Katastrophenschutz gemacht. Sie gilt dennoch nicht nur für den zeitkritischen Einsatz im Falle eines Desasters, sondern ist auch übertragbar auf Visualisierungen eines Spezialthemas, zum Beispiel in der Geologie. Dies gilt vor allem dann, wenn die Thematiken an Entscheidungsträger aus Politik und Ämtern herangetragen werden, zumal sie oft als Grundlage für Richtungsvorgaben dienen. MacEachren et al. (2004) weisen darauf hin, dass in den meisten Situationen und Anwendungsbereichen Geovisualisierungen nicht die einzigen Werkzeuge sind, die von Agierenden verwendet werden müssen. Da sie also nicht die alleinige Aufmerksamkeit werhalten werden, sollten die virtuellen Welten und 3D-Karten vor allem in der Anwendung leicht bedienbar sein und möglichst wenig verwirren.

Im Folgenden werden einige herausstechende Ansätze in der dreidimensionalen Geovisualisierung über das Internet besprochen und in Hinsicht auf die Eignung für geologische Daten untersucht. Am publikumswirksamsten treten auf diesem Gebiet die virtuellen Globen auf, meist gratis herunterladbare Viewer, die den Nutzer mithilfe einer einfachen Bedienung durch immense Sammlungen von Geodaten „browsen“ lassen. Sie sind auch unter dem Namen *Geo-Browser* bekannt. Andere

Lösungen, die den Datenaustausch via Internet ermöglichen, sich aber eher an Fachpersonal aus dem Geo-Bereich wenden, sind die Web-Dienste des OGC, die auch immer stärker in das 3D-Segment vorrücken. Letztlich werden noch einige, nicht originär für Geodaten entwickelte Lösungen vorgestellt, die sich aber durchaus in gewissen Zusammenhängen dazu eignen, Kartenmaterial online zu verbreiten und mit Interaktionsmöglichkeiten auszustatten. Darunter fallen unter anderem Rich Internet Applikations wie Microsoft Silverlight, AdobeFlex und Java3D oder auch 3D-Dateiformate wie 3D-PDF und X3D.

3.1 Geo-Browser und virtuelle Globen

Dass sich kartographische 3D-Darstellungen im Internet mittlerweile auch bei einem breiten Publikum durchgesetzt haben, zeigt das vergleichsweise alte Beispiel der Geo-Browser, die nicht nur die Erdoberfläche als virtuellen Globus darstellen, sondern auch Zusatzdaten wie 3D-Stadtmodelle sichtbar machen. Der erste dieser virtuellen Globen war NASAs World Wind, der mittlerweile bekannteste dürfte Google Earth sein.

Es gibt heute eine Vielzahl von so genannten Geo-Browsern, auch virtuelle Globen oder Earth-Browser genannt. Bei Kolar (2006) finden sich weitere Begriffe für die gleiche Technologie: *digitale Erde*, *virtuelle Erde*, *Geographic Exploration Systems* etc. Darunter versteht man Computerprogramme, die zur dreidimensionalen Navigation durch verschiedene Geodaten auf dem Bildschirm dienen, seien es Satelliten- und Luftbilder oder thematische *Overlays*. Die virtuellen Globen bedienen sich der aussagekräftigen Metapher des *Planetes*: Geoinformationen werden auf einen virtuellen Erdball projiziert. Sie bieten den Menschen die Möglichkeit, mithilfe dieser intuitiv verständlichen Metapher und einem guten Navigationsinterface Geodaten zu verstehen (Scharl 2007). So gesehen erleichtert also die 3D-Darstellung mit der intuitiv verständlichen und allgemein bekannten Metapher die Kognition von Geodaten. Die Daten, die den Geo-Browsern zugrunde liegen, werden entweder aus dem Internet bezogen oder sind, in geringerer Auflösung, auf der Festplatte des jeweiligen Computers gespeichert. Sie umfassen meist den gesamten Erdball und mithilfe eines Interfaces könne die Nutzer den Globus drehen, Ansichten vergrößern und verkleinern und beliebig die Perspektive wechseln, sodass der Eindruck entsteht, man fliege über die Erdoberfläche. Was zur Popularität der virtuellen Globen

beigetragen hat, ist die Möglichkeit für Nutzer, eigene Daten in so genannten *Mashups* in eigene Plattformen einzufügen (Scharl 2007) und sie über Internet zu veröffentlichen oder per e-Mail anderen Personen verfügbar zu machen. Virtuelle Globen sind aber mehr als nur Verteiler oder Zugangspunkte zu Geodaten; sie geben den Daten Bedeutung und machen sie nutzbar auch für Nicht-Experten. Die Dreidimensionalität unterstützt die Intuition dieser Nutzer, da sie die Welt nah an deren alltäglicher Umgebung rekonstruiert (Bodum & Jaegly 2006). Mithilfe von Application Programming Interfaces (API) oder dem Java Software Development Kit (JavaSDK) kann jeder die Funktionen des eigenen virtuellen Globus' erweitern oder eigene Inhalte darauf platzieren. Von Urlaubsfotos bis zu wissenschaftlich erstellten Rasterbildern, die einen thematischen Hintergrund beleuchten: in vielen Bereichen sind die Möglichkeiten extrem vielfältig, solche Overlays zu erstellen. Da hierdurch der virtuelle Globus immer mehr als Plattform für andere kartographische Ausdrucksformen genutzt wird und nur mehr ein Feature von vielen anderen ist, soll in dieser Arbeit primär der Begriff *Geo-Browser* verwendet werden (vgl. Riedl & Kristen 2010).

Die Geo-Browser bieten nicht nur die Möglichkeit, zweidimensionale Daten auf der Erdoberfläche anzuzeigen. World Wind erlaubt es seinen Nutzer auch, ein eigenes Digitales Höhenmodell (DHM) für die dreidimensionale Darstellung des Geländes zu verwenden. Außerdem können 3D-Modelle von Gebäuden, die in anderen Programmen erstellt wurden, in die virtuellen Globen importiert werden. Auch 3D-Vektorgraphiken können individuell gestaltet und dargestellt werden. Beispielsweise können Lufträume über der Erdoberfläche in farbigen Quadern visualisiert werden.

Die Modifizierbarkeit der Darstellung oder die Verwendung eigener Daten kennt aber auch Grenzen. Sehr große Rasterbilder sind nicht mehr einfach ohne gewisse Kenntnisse einer Scripting-Sprache einzubauen, die Möglichkeit GPS-Daten zu verwenden kann kostenpflichtig sein, oder geologische und andere Daten, die sich *unter* der Erdoberfläche befinden, können (derzeit) gar nicht sichtbar gemacht werden (Nadeau et al. 2009).

Essentiell für die Funktionstüchtigkeit von Geo-Browsern im Speziellen und 3D-Anwendungen im Allgemeinen ist die Navigationsmöglichkeit für den Benutzer. Je einfacher das Programm zu handhaben ist, desto schneller und leichter erfolgt der Zugriff auf die für den Betrachter relevanten Daten, das heißt, desto effektiver ist die Anwendung zu gebrauchen. Die Navigation durch 3D-Daten unterscheidet sich wesentlich von der zweidimensionalen Navigation. Im Fall einer 2D-Karte ist eine Navigation nicht essentiell, vor allem wenn die gesamte Information auf dem Bildschirm Platz hat. Dreidimensionale Daten werden dagegen ohne Navigationsmöglichkeiten schnell nutzlos (Kolar 2006). Außerdem sind die Navigationsmöglichkeiten in 3D vielfältiger, verglichen mit den 2D-Funktionen „Verschieben“ (panning) und Zoom.

Es gibt mittlerweile ein großes Angebot von diesen virtuellen Globen. Oft kann man sie einfach über das Internet herunterladen und auf dem heimischen PC installieren. Prominente Vertreter dieser Klasse sind Google Earth und Virtual Earth. Die Client-Programme fragen die betrachteten Daten von einem Server über das Internet ab, während die Betrachter durch die virtuelle Welt navigieren. Einige der Globen sind auch kostenpflichtig oder in kostenpflichtigen Versionen erhältlich. Von ESRI gibt es beispielsweise ArcGlobe, das eine Erweiterung des Spatial Analysts ist, und somit letzteren voraussetzt, was wiederum das Programm ArcGIS voraussetzt. Diese Software ist also nicht gratis nutzbar.

Im Folgenden werden prominente Vertreter der Earth Browser genauer vorgestellt und daraufhin untersucht, welche Möglichkeiten sie bieten, auch eigene kartographische Daten vor allem im geologischen Kontext darzustellen.

NASA World Wind

NASA World Wind ist als gratis Download in der Version 1.4 im Internet erhältlich.² Als Systemvoraussetzungen werden dort unter anderem eine Grafikkarte mit 3D-Beschleunigung und eine schnelle Internetverbindung (mindestens DSL) angeführt. Schon diese Voraussetzungen lassen Rückschlüsse auf die Funktionsweise zu. Die Daten werden je nach Ausschnitt, der gerade am PC angeschaut wird, aus dem

² <http://worldwind.arc.nasa.gov/download.html>

Internet bezogen. Die Server von NASA versorgen die Nutzer im Moment des Anschauens und somit ist für eine flüssige Darstellung der Fluss der Daten selbst unentbehrlich. Damit die Daten dann auch aus allen Lagen, Blickwinkeln und Distanzen erfahrbar sind, muss der Computer über leistungsfähige Darstellungsmethoden verfügen, die nicht allein vom Hauptprozessor geleistet werden können. Seit etwa 2003 werden ausreichend schnelle 3D-Beschleuniger aber serienmäßig in PCs eingebaut (NASA 2005).

Die zur Verfügung stehenden Daten für die Grundkarte sind Satellitenbilddaten, die mit den Höhendaten der Shuttle Radar Topography Mission (SRTM) kombiniert werden. Weitere Daten im Vektorformat, wie Gitternetze, Staatsgrenzen und Ortsnamen stehen bereit. Es können auch von den Nutzern eigene Daten integriert und dargestellt werden. Beim Entwurf der Navigationsfunktionen wurde darauf geachtet, sie so einfach zu entwerfen, dass alle Altersgruppen das Programm bedienen können.

Die wichtigsten World Wind Grundlegendaten (NASA 2005) im Überblick:

- **Blue Marble**

Blue Marble-Daten sind eine fotografische Darstellung der Erdoberfläche, die aus verschiedenen Quellen, wie zum Beispiel Daten unterschiedlicher Satelliten, zusammengestellt wurden. Die Auflösung beträgt etwa 1 km.

- **Landsat7**

Die Daten des Landsat7-Satelliten für NASA World Wind haben eine Auflösung von 15m pro Pixel. Sie sind zusammengesetzt aus Bildern der Jahre 1999-2003. Neben den Aufnahmen aus dem sichtbaren Spektrum sollen auch die anderen Spektralbereiche des Scanners wie Infrarot zur Verfügung gestellt werden. Da es sich hierbei um einen sehr großen Datensatz handelt, wird nur eine komprimierte Version des im jeweiligen Moment ausgegebenen Ausschnitts auf dem eigenen Computer gespeichert.

- **SRTM**

Die SRTM-Daten sind Höhendaten aus der Shuttle Radar Topography Mission, die fast für den gesamten Globus zur Verfügung stehen. Das Gebiet

zwischen 54°N und 60°S wird mit einer Auflösung von 1-3 arcsec (Bodensekunden) abgedeckt (DLR 2009).

- **MODIS**

Die MODIS³-Daten haben zwar nur eine Auflösung von etwa 250m pro Pixel, werden aber täglich aktualisiert und in World Wind integriert. Spezielle farbcodierte Symbole zeigen aktuelle Informationen wie Feuer, Überflutungen, Rauch und sogar vulkanische Aktivitäten an. Durch einen Klick kann das dazugehörige Bild geladen werden.

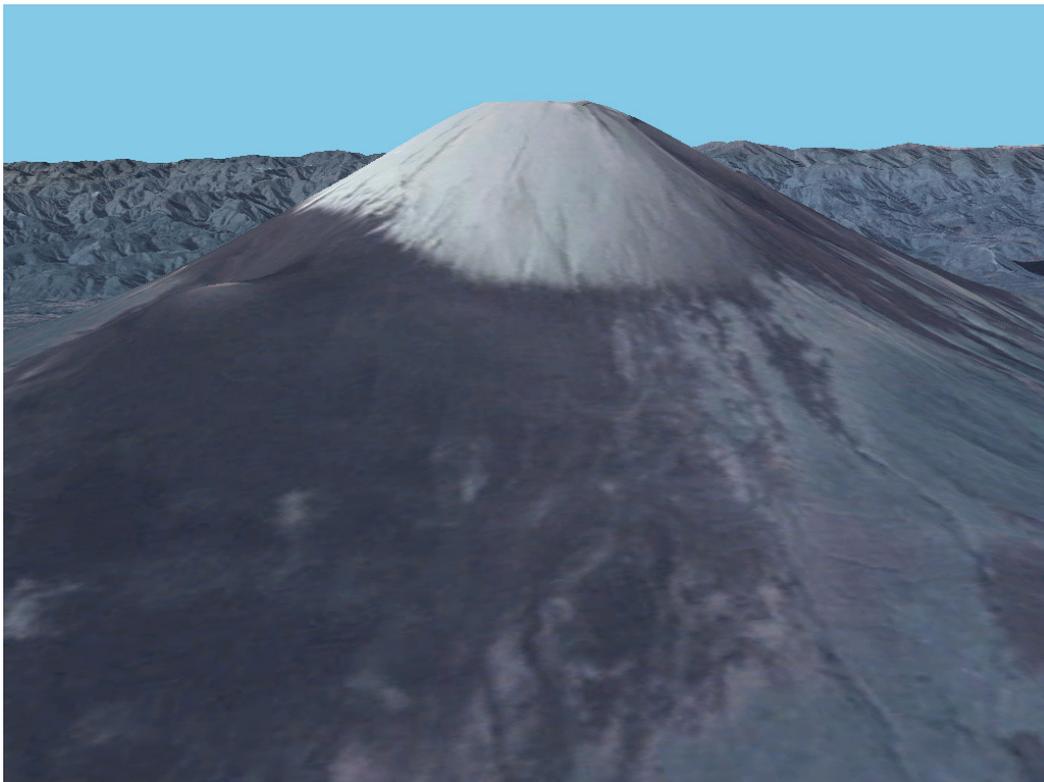


Abbildung 16: World Wind Screenshot vom Berg Fuji, Japan mit SRTM-Daten.
Quelle: <http://worldwind.arc.nasa.gov/screenshots-sl.html>

World Wind ist bei der NASA in der Abteilung „Learning Technologies“ angesiedelt. Die verfügbaren Daten und die Einbindung des Wissensnetzwerks „Global Learning and Observation to Benefit the Environment“ (GLOBE) machen den wissenschaftlichen und pädagogischen Anspruch der Applikation deutlich. Dieser

³ MODIS – Moderate Resolution Imaging Spectroradiometer. Der MODIS-Sensor ist auf den Terra (EOS AM) und Aqua (EOS PM) Satelliten montiert. Für weitere Informationen: <http://modis.gsfc.nasa.gov/about/>

Eindruck wird dadurch unterstützt, dass auch Daten zur Visualisierung der Marsoberfläche und des Mondes betrachtet werden können.

NASA World Wind unterstützt das KML (Keyhole Markup Language)-Dateiformat. Formuliert nach den allgemeinen XML-Regeln können KML-Dateien dazu verwendet werden, die virtuellen Browser mit eigenen Daten zu bereichern. Zum Beispiel können Rasterdaten als Overlays oder Fotos als Billboards angezeigt, anklickbare Marker eingebaut oder Vektorgraphiken erstellt werden. Mit dem JavaSDK können Entwickler kleine Applikationen für World Wind schreiben und so auch die Funktionen des Globus‘ erweitern.

Google Earth und Microsoft Virtual Earth

Microsofts Virtual Earth und Google Earth sind sich im Prinzip recht ähnlich. Sie werden von kommerziellen Firmen betrieben. Die Viewer, mit denen die Daten betrachtet werden, stehen Interessierten unentgeltlich zur Verfügung und können über das Internet geladen werden. Die angezeigten Daten (Satelliten- und Luftbilder) können aber nur betrachtet werden und nicht etwa zum Weitergebrauch auf den heimischen PC herunter geladen werden. Google Earth ist wohl der am weitesten verbreitete Geo-Browser. Er wird ständig weiterentwickelt und ist mittlerweile in der Version 5 frei erhältlich. Neben der Gratisversion (Abbildung 18) gibt es auch die kostenpflichtigen Plus- und Pro-Versionen, die zusätzliche Funktionen wie beispielsweise das Geocodieren von Adressen erlauben.

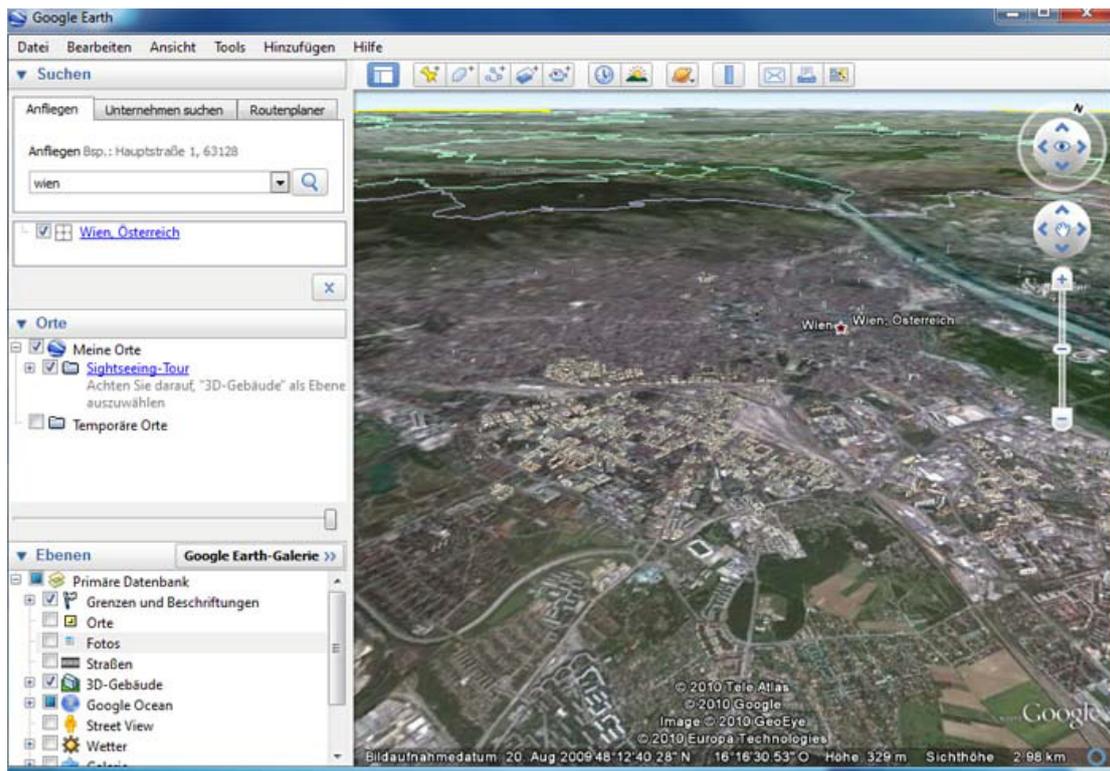


Abbildung 17: Die Gratisversion von Google Earth. Der Screenshot zeigt eine Ansicht von Wien. Quelle: Google Earth

Auch Google Earth beschränkt seine Darstellung nicht mehr ausschließlich auf die Erde: Die NASA-Daten von Mars und Mond können mittlerweile auch dargestellt werden. Historische Daten sowie die Darstellung des Meeresgrundes haben auch Einzug in die Welt von Google Earth erhalten. Google aktualisiert seine Bilddaten regelmäßig und gibt ihr Alter mit ein bis drei Jahren an. Es werden neben Satellitenbildern und 3D-Modellen auch Luftbilder und verschiedene Vektordaten verwendet (Google 2010).

Einer der augenscheinlichsten Unterschiede zwischen Google Earth und Virtual Earth ist, dass man in letzterem den Blickwinkel verstellen kann. Während bei Google Earth Satelliten- und Luftbilder die Erdoberfläche immer aus einem annähernd rechten Winkel zeigen (auch wenn die Kamera durchaus geneigt werden kann), kann bei Luftbildern von Microsoft auch in die Vogelperspektive, genannt „bird’s eye view“, gewechselt werden. So können Bereiche, für die die Schrägluftaufnahmen bereits vorliegen, aus mehreren Winkeln betrachtet werden. Leider werden keine Angaben zu Aktualisierungsraten des Bildmaterials gemacht. Dafür gibt es Angaben über die räumliche Verteilung der Datenauflösung. Für viele Länder in den Regionen Europa,

Asien/Pazifik und Nordamerika sind bis zu 90 Prozent aller Straßen erfasst. In Afrika, Zentral- und Südamerika sind zumeist wenigstens die Straßen erster und zweiter Ordnung erfasst (Via Windows Live 2009).

Die Nutzer haben aber nicht nur Zugriff auf viele Luft- und Satellitenbilder sondern können auch selbst Einfluss auf das Darzustellende nehmen, denn beide Plattformen unterstützen die Keyhole Markup Language (KML). Allerdings wird KML bei Virtual Earth noch nicht durch die API unterstützt (Via Windows Live 2009). In Virtual Earth besteht aber die Möglichkeit, durch APIs und Software Development Kits die Datengrundlage des virtuellen Globus' für eigene kleine Programme im Netz oder auf Mobilgeräten zu nutzen. Ein Beispiel für ein so genanntes App (Kurzform von Application) ist eine kleine Anwendung, die aufgrund der Routenberechnung angenäherte Taxi-Fahrtkosten berechnet (Ricky's Bing Maps Blogg 2010).

GRIFINOR

Ein weiterer Geo-Browser, der hier aufgrund seiner Besonderheiten erwähnt werden soll, ist aus dem GRIFINOR-Projekt der Universität Aalborg entstanden. Es handelt sich um ein Open Source-Projekt, das zwar als virtueller Globus durchaus vergleichbar mit Google Earth oder NASA World Wind ist. Doch kann es auch als Plattform für Geovisualisierung in einem allgemeineren Sinn verstanden werden. GRIFINOR soll das Internet mit geographischen Referenzen versorgen. Es gibt drei wesentliche Unterschiede zu den anderen Earth-Browsern. GRIFINOR ist direkt vom Webbrowser abrufbar, es ist objektorientiert und dezentralisiert (Bodum & Jaegly 2006). Alle drei Aspekte erweitern und unterstützen die Möglichkeiten der Einflussnahme des Nutzers auf die angezeigten Daten.

In den vorangegangenen Geo-Browsern werden die Daten zumindest der Kartengrundlage, wie zum Beispiel Satellitenbilder, und verschiedene Vektorgrafiken, von einem zentralen Server aus auf den Heimcomputer, den Client, geliefert. Im dezentralen System von GRIFINOR kann jeder Client auch die Rolle eines Servers übernehmen. Dann können Geodaten an viele andere Knotenpunkte im Netzwerk über das Internet gesendet werden und so Inhalte geteilt werden. Bodum & Jaegly (2006) vergleichen diesen Transfer mit einem Gnutella-Netzwerk, bei dem Anfragen indirekt

nach dem „Schneeballsystem“ im Netzwerk verteilt werden. Die Anfrage wird an mehrere Knotenpunkte gesendet, die diese wiederum an weitere Knoten versenden bis die gesuchten Daten gefunden wurden und entlang des Pfades wieder zurückgesendet werden. Es können also sowohl eigene Daten mit GRIFINOR visualisiert und veröffentlicht werden sowie fremde Daten, die auf verschiedenen Servern liegen. GRIFINOR wurde in der objektorientierten Programmiersprache Java realisiert. Jedes dargestellte Feature ist ein Java-Objekt, das mit anderen Java-Objekten interagieren kann. Das gilt nicht nur für 3D-Modelle von Gebäuden und ähnlichen Dingen, sondern auch für das Terrain in GRIFINOR. Da es ein TIN (Triangulated Irregular Network) ist, kann es mit der geschmeidigen Darstellung in Google Earth zum Beispiel nicht mithalten. Den großen Vorteil in dieser Architektur sehen Bodum & Jaegly (2006) aber darin, dass dadurch das Terrain zu einem ansprechbaren und damit beeinflussbaren Programmierobjekt wird, das viele weitere Möglichkeiten eröffnet. Dabei ist im Kontext dieser Arbeit natürlich an die Visualisierung von Daten unter der Erdoberfläche zu denken, die beispielsweise als mehrere übereinander liegende Oberflächenmodelle der Schichtenoberkanten verwirklicht werden könnte. Ein weiterer Vorteil ist, dass GRIFINOR plattformunabhängig in einer Java Runtime Umgebung läuft und auch als Webstart-Applikation einfach über einen Browser gestartet werden kann. Ein Nachteil der Java-Implementierung ist, dass alle Applikationen und Implementationen diverser Datenrepräsentationen auch in Java umgesetzt werden müssen (Kolar 2006).

Da GRIFINOR unter der LGPL (Lesser GNU Public License) kursiert, ist eine Modifikation der Software für jeden erlaubt, sowohl für freie Software als auch für proprietäre (GNU Operating Systems 2007). Das Programm kann also um Funktionen erweitert und speziellen Anforderungen angepasst werden (Kolar 2006).

Die vorhergehenden Betrachtungen zeigen, dass die virtuellen Globen und Geo-Browser durchaus Meilensteine in der Geschichte der Internetkartographie sind. Offenbar ist das Interesse der Öffentlichkeit an Geovisualisierungen ungebrochen groß und die Begeisterung ist ungebremst. Gerade die einfache Nutzbarkeit hat zur Popularität dieser Systeme beigetragen. Die Navigation durch den virtuellen Raum ist bei vielen der Globen sehr gut gelungen und zu Recht intuitiv zu nennen, insofern sie

nicht erst erlernt werden muss, sondern vom ersten Öffnen der Anwendung an schon zufrieden stellend genutzt und verstanden werden kann. Dies minimiert die Frustrationserlebnisse, die viele Nutzer von der Wiederverwendung eines solchen Produkts abhalten könnten. Außerdem spricht für die Geobrowser, dass sie individuell erweiterbar sind. Zusätzliche Informationen, ob in Text- oder Bildform, können über Pop-up-Fenster eingeblendet werden, und eigene Daten können als Overlays oder als 3D-Modelle einfach über KML-Dateien sichtbar gemacht werden. Allerdings hat diese Erweiterbarkeit ihre Grenzen: Für die Anwendungen im geologischen Bereich ist zum Zeitpunkt des Verfassens dieser Arbeit keiner der virtuellen Globen geeignet, da die Erdoberfläche nicht „durchdrungen“ werden kann. Bislang ist die Darstellung von Daten unterhalb der Erdoberfläche nicht möglich. Immerhin befasst sich bei der NASA seit Kurzem ein Team mit dieser Aufgabe, sodass vielleicht in Zukunft mithilfe von World Wind auch Daten unter der Erde sichtbar gemacht werden können (Nadeau et al. 2009). Davon ist bei Google und Microsoft noch nichts verlautbart worden. Im Falle von GRIFINOR wären die Möglichkeiten vielleicht noch am ehesten gegeben, sich den Globus selbst so anzupassen, dass er auch geologische Fragestellungen erhellen könnte. Leider ist dieses Projekt aber schon eingestellt worden und damit nicht zukunftsfähig.

Die Betrachtung der virtuellen Globen als Vorreiter der 3D-Internetvisualisierungen von Geodaten zeigt: Die einfache Navigation und die Möglichkeit, eigene Daten leicht in die Visualisierung einfließen zu lassen, sprächen sehr für den Entwurf der hier angestrebten geologischen Visualisierung im Rahmen dieser bereits vorhandenen 3D-Welten. Die Einflussmöglichkeiten des Nutzers gehen aber nicht weit genug, um geologische Daten zu verwenden.

3.2 OGC Web Services

Das Open Geospatial Consortium (OGC) bietet Spezifikationen zu vielen verschiedenen so genannten Web Services an. Web Services sollen die Verwendung des Internets für den Zugang zu Geodaten und die Generierung kartographischer Darstellungsformen ermöglichen. Das Ziel ist größtmögliche Interoperabilität, sodass zum Beispiel unterschiedliche, eigenständige Server die Daten für eine einzige Karte

liefern können. Oder andersherum, dass für die Erstellung einer Karte verschiedene Server als Quellen verwendet werden können. Den Prominentesten dieser Internetdienste stellt zweifellos der Web Map Service (WMS) dar. Dieser wird in zahlreichen Themenfeldern im Internet angewendet. Das liegt auch an seiner recht ausgereiften Funktionsweise und der Verbreitung von freier Software wie zum Beispiel MapServer⁴. WMS erlaubt also den Abruf etwa von Teilausschnitten, Layern oder Attributen einer Karte aus dem Internet und liefert ein gerendertes Bild, beispielsweise im JPG oder PNG-Format zurück an den Browser (Client) des Internetnutzers. Da WMS auf zweidimensionale Karten beschränkt ist, gibt es mehrere Ansätze, die ähnliche Funktionalitäten auch für 3D-Karten einrichten wollen. In diesem Zuge sind der Web Coverage Service (WCS), der Web Feature Service (WFS) und vor allem der Web Perspective View Service (WPVS), der Web Terrain Service (WTS) und der Web 3D Service (W3DS) zu nennen. Über WCS und WFS können Geodaten zwar auch in 3D angefordert werden, sie werden aber in ihrer originären Form an den Client ausgegeben und dort muss der weitere Prozess der Visualisierung, auch Visualisierungspipeline genannt, fortgeführt werden. Dementsprechend sind WCS und WFS den Datendiensten (data services) zuzurechnen (vgl. Kolbe 2004). Wenn der Client einen Großteil Pipeline selber bewältigt, spricht man von einem Thick Client. Dieser ist also mit vielen Extrafunktionalitäten ausgestattet. Bei der Öffentlichkeit ist aber solch ein Thick Client mit Spezialfunktionen zur Visualisierung von Geodaten nicht unbedingt voraussetzbar. Der Vorteil von WMS ist dagegen, dass auch das Erstellen der Abbildung (Rendering) auf dem Server geschieht, der selbstständig eine Bilddatei kreiert, die von jedem Browser gelesen werden kann. Diese Art von Service wird Visualisierungsdienst (portrayal service) genannt (Kolbe 2004). Es wird also nur ein Thin Client benötigt, was die Verbreitungsmöglichkeiten der Karten deutlich erhöht: mehr User können sich die Karten beispielsweise in ihren Web-Browsern anschauen. In den ausgegebenen 2D-Karten können verschiedenste thematische Layer einfach überlagert werden, wofür lediglich Transparenzen notwendig werden können, um andere Informationen nicht zu überdecken. So können auf eine Grundkarte mehr oder weniger beliebig viele Informationen „geschichtet“ werden. Das funktioniert aber nur

⁴ <http://mapserver.org/>

deshalb so gut, weil man bei 2D-Karten immer dieselbe Perspektive auf das Dargestellte hat, nämlich die orthorektifizierte Parallelperspektive rechtwinklig zur Erdoberfläche. Lediglich die Projektion der Geodaten muss gegebenenfalls in ein einheitliches System überführt werden. Bei 3D-Karten dagegen ist es geradezu das herausstechende Charakteristikum, dass die Perspektive gewechselt werden kann. Da es bereits reine Datendienste für dreidimensionale Geodaten gibt (WCS, WFS), besteht die Herausforderung darin, einen Visualisierungsdienst in 3D zu implementieren. Bisher sind zu diesem Thema verschiedene Standards in Arbeit aber vom OGC noch nicht verabschiedet worden. Man kann grundsätzlich zwei Ansätze unterscheiden. Bei WPVS bzw. WTS werden in der Anforderung von Client an den Server bereits Angaben zur Perspektive gemacht, das heißt, es wird vorher schon festgelegt, welche Sicht auf das entsprechende Gebiet oder Objekt geworfen wird. Der Server bearbeitet die Anfrage und berechnet ein Bild, das den Wünschen des Users entspricht, der ja bereits die Kameraplatzierung im virtuellen 3D-Raum festgelegt hat. Es handelt sich hierbei um ein statisches Bild in einem der gängigen Grafikformate. Damit ist die Perspektive nicht mehr veränderbar, sobald das Bild berechnet und ausgegeben wurde. Es gibt Ansätze und Ideen zur Weiterentwicklung des Services, zum Beispiel den WVPS++. Dem ausgegebenen Bild soll hierbei ein automatisch erstellter, interaktiver Layer überlagert werden, der zusätzliche Attributdaten enthält und weitere Objekte anklickbar macht. So soll dem User die Möglichkeit geboten werden, sich durch die virtuelle Welt zu navigieren (Hagedorn et al. 2010), indem immer neue Anfragen an den Server gesendet werden und dieser weitere Ansichten auswirft. Der Vorteil dieses Ansatzes ist, dass hier, genau wie bei WMS, lediglich ein Thin Client benötigt wird. Ein üblicher Web-Browser dürfte für die Betrachtung der interaktiven Bilder ausreichend sein. Die Handhabung des zugrunde liegenden 3D-Modells und die Ausgabe des fertigen Bildes werden serverseitig erledigt. WPVS/WTS sind bereits in der freien Software deegree in der Versionsnummer 2 implementiert. WPVS wird auch in der Version deegree3 dabei sein (deegree 2010).

Der andere Ansatz, verwirklicht im W3DS, versucht, dem Nutzer weit reichende Interaktionen zu ermöglichen und den vollen Nutzen aus den 3D-Daten zu ziehen. Dafür generiert der Web-Dienst 3D-Szenen im VRML-Format, wobei GeoVRML

oder X3D im Discussion Paper des OGC empfohlen werden (OGC 2005). Das OGC hat die Spezifikationen zu W3DS noch nicht offiziell verabschiedet. Sie befinden sich noch als Entwürfe in der Diskussion und sind erst unter der Versionsnummer 0.3.0 erhältlich. Dennoch gibt es schon einige Implementierungen, von denen hier ein Projekt aus Nordrhein-Westfalen, Deutschland, umrissen werden soll, mit dem alles begann.

Die Bereitstellung dieses sehr großen Datensatzes, der in CityGML vorlag, über den W3DS wird in Kulawik et al. (2009) beschrieben. Der Grunddatensatz enthielt über 6 Millionen Gebäude und war rund 34 GB groß. Die Daten wurden gekachelt, um dann vom W3DS im VRML-Format ausgegeben zu werden. VRML wird als effizientes Grafikformat bezeichnet. Das Datenvolumen von 3D-Karten ist sehr viel größer als das der 2D-Karten eines WMS, weswegen die 3D-Daten stückweise über ein Streamingverfahren an den Client übermittelt werden können. Im konkreten Projekt „Geodateninfrastruktur Nordrhein-Westfalen“ (GDI-3D 2010) wurde ein so genanntes „Pseudostreaming“-Verfahren entwickelt, das den Nutzer mit verschiedenen Levels of Detail (LOD) versorgt und große digitale Geländemodelle (DGM) in Kacheln aufteilt, die auch unterschiedliche Auflösungen aufweisen können. Dafür wurde ein eigener Viewer entwickelt, der je nach Kameraposition (nah oder fern) hoch oder niedrig aufgelöste Kacheln anzeigt. Speicherkapazitäten des Clients werden geschont, indem immer nur die gerade sichtbaren Kacheln in der 3D-Szene enthalten sind (Basanow et al. 2007). Der Viewer muss vor allem die Fähigkeiten besitzen, 3D-Szenen zu rendern, die dann auf dem 2D-Bildschirm ausgegeben werden. Dies erfordert also einen Medium Client (OGC 2005, Abbildung 18).

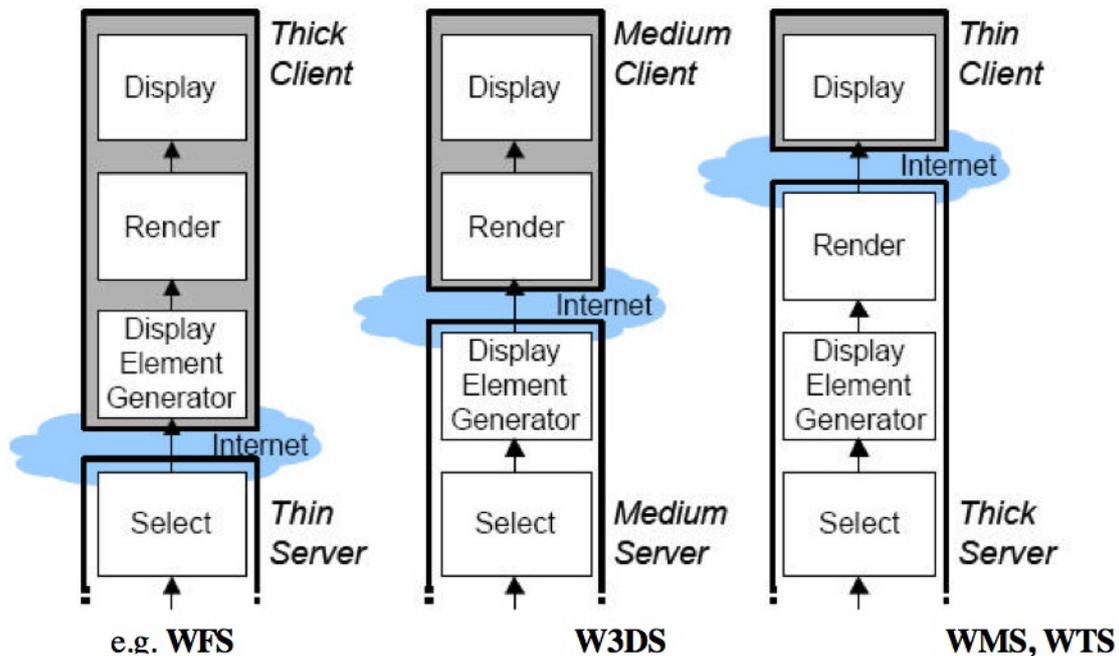


Abbildung 18: Die verschiedenen Client/Server-Konzepte und entsprechende OGC-Webservices. Quelle: OGC 2005

Weitere aktuelle Projekte betreffen die Integration der deutschlandweiten Open Street Map-Daten als 3D-Modell in den W3DS (Schilling et al. 2009) oder die Einbindung von dynamischen Sensor-Daten in die 3D-Visualisierung (Mayer & Zipf 2009).

Neben diesen Ansätzen aus Wissenschaft und Verwaltung gibt es auch schon einen kommerziellen Einsatz des W3DS, entwickelt von der Firma PiriReis unter dem Namen CitySurf (CitySurf 2009, OGC 2010).

3.3 3D-Datenformate im Netz

Generell werden unzählige Möglichkeiten angeboten, 3D-Inhalte aller Art im Internet zu veröffentlichen. Viele bekannte und weniger bekannte Firmen bieten diverse eigene Entwicklungen an. Im folgenden Abschnitt wurde eine subjektive Auswahl der wichtigsten Formate und Möglichkeiten einer näheren Untersuchung unterzogen.

3.3.1 Universal 3D/3D-PDF

Universal 3D (U3D) ist ein offenes und erweiterbares 3D-Dateiformat, das sich in Adobes Acrobat 9 in ein PDF-Dokument einbetten und im Adobe Reader anzeigen lässt. Die Spezifikationen von U3D sind in der Normbeschreibung der Ecma International (European association for standardizing information and communication systems) festgelegt (Deshmuk 2004). U3D wurde speziell für Visualisierungsanwendungen von CAD-Daten entwickelt, die in einem nicht-technischen Bereich oder einer Design-Umgebung präsentiert werden sollen, wie zum Beispiel in Trainings etc. Obwohl der Fokus auf CAD gelegt wurde, soll U3D aber durchaus ein, wie der Name sagt, universelles Datenformat sein. Der Standard ist offen und erweiterbar. Die Code-Norm kann im Internet eingesehen werden.⁵ Die Spezifikationen zu U3D sind relativ neu. Sie wurden 2003 vom 3D Industry Forum eingerichtet und stehen seit 2004 unter der Obhut des Ecma International. Die neueste Ausgabe stammt aus dem Jahr 2007. Einige namhafte Firmen sind Mitglieder des Forums, unter anderem Microsoft, Adobe Systems Inc, Intel Corporation, Cinema4D, NVIDIA und Hewlett-Packard. Der Vorteil von U3D liegt sicherlich darin, dass es in gewöhnliche PDF-Dateien eingebettet und von den gratis erhältlichen Adobe Readern (ab Version 7) angezeigt werden kann. Obwohl keine offiziellen Zahlen veröffentlicht wurden, kann mit großer Sicherheit behauptet werden, dass der Adobe Reader eine extrem hohe Marktdurchdringung erreicht hat. Es werden aber keine breit angelegten Multimedia-Anwendungsfälle im Consumer-Bereich angestrebt, sondern U3D dient der Produktvisualisierung. 3D-Informationen sollen einer breiten Öffentlichkeit auf Laptop, PC oder mobilen Geräten, sowie auf Servern zugänglich gemacht werden. Normale Internet-Browser mit gratis Plug-in sollen für die Anzeige ausreichen. Dadurch sollen die Daten übers Internet zur Verfügung gestellt werden können und somit überall zu jedem Zeitpunkt abrufbar sein. Ziel ist es, U3D als einzigen globalen Standard durchzusetzen. Ecma International führt auf der Prioritätenliste in seinem Ausblick für 2010 (Memento) U3D an zweiter Stelle (Ecma 2010).

⁵ <http://sourceforge.net/projects/u3d>

3.3.2 O3D/WebGL

Ursprünglich als neuer, offener Standard vorgesehen sollte das von Google entwickelte O3D-Format als universell genutztes 3D-Format etabliert werden. Das auf einem Plug-in basierende API wurde allerdings im Mai 2010 eingestellt und stattdessen in eine auf WebGL (zu WebGL siehe auch: Kapitel 9 Ausblick) basierende JavaScript-Library umgewandelt. Als Gründe werden die eigene Entwicklungsarbeit an WebGL und die gesteigerte Leistungsfähigkeit von JavaScript angeführt (O3D team 2010). Das zeigt den Glauben daran und den Willen von Google, WebGL zu einem allgemein anerkannten und verwendeten Standard zu machen. Da aber zum Zeitpunkt des Verfassens dieser Arbeit keine offiziellen Browser-Versionen existieren, die schon WebGL unterstützen, und auch WebGL selbst noch nicht offiziell veröffentlicht wurde, kann O3D in dieser Arbeit noch nicht erschöpfend behandelt werden und wurde der Vollständigkeit halber und mit Blick in die Zukunft (siehe auch Kapitel 9 Ausblick) an dieser Stelle erwähnt.

3.3.3 Rich Internet Applications mit Adobe Flex, JavaFX, MS Silverlight

Der Begriff Rich Internet Application (RIA) steht für Programme, die aus einem gewöhnlichen Web-Browser heraus über das Internet abrufbar sind und dem Internetsurfer weit mehr Funktionen für den dargestellten Inhalt zur Verfügung stellen, als es Web-Seiten üblicherweise tun (Abbildung 19). Typisch für solche Programme ist ein eigenes graphisches User-Interface (GUI) sowie ein hoher Grad an Interaktivität.

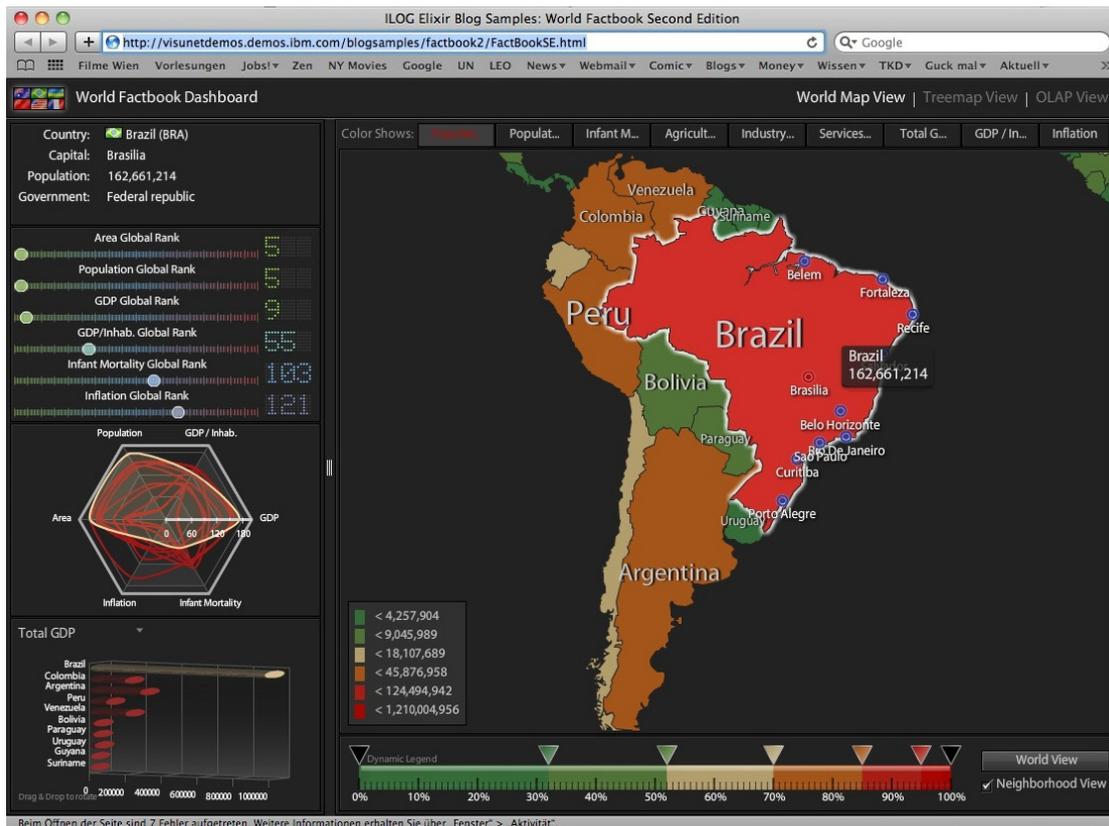


Abbildung 19: Beispiel einer RIA. Im Browserfenster ist ein GUI für das CIA Factbook geöffnet. Es sind interaktive Legenden, Charts und 3D-Graphiken zu erkennen. Quelle:

<http://visunetdemos.demos.ibm.com/blogsamples/factbook2/FactBookSE.html>

Diese Funktionen können im Backend durch Programmiersprachen wie Java, C#, Python oder ähnliche definiert sein. RIAs brauchen daher ein Framework, in dem sie laufen und das die Programmiersprachen interpretieren kann; und ein Plug-in, über das der Browser auf das Framework zugreifen kann, und das gewissermaßen die Verbindung zwischen angezeigtem Inhalt im Frontend und den im Hintergrund laufenden Funktionen des Backends herstellt. Es gibt einen großen Markt für RIAs, der von Firmen wie Microsoft, Adobe und Oracle hart umkämpft ist und die respektive die Frameworks Silverlight, Flex und JavaFX auf den Markt gebracht haben. Dabei geht es zum Großteil um die Verbreitung der Plug-ins, von denen abhängt, ob die Inhalte der Frameworks auf den Client-PCs angezeigt werden können. Jedes Framework braucht ein eigenes Plug-in oder, im Fall von JavaFX, eine aktuelle Java-Version. Die Applications sind nicht austauschbar und laufen also nur in den proprietären Umgebungen. Inhalte, die mit Silverlight erstellt wurden, können also

nicht etwa mit JavaFX oder Flex angeschaut werden usw. Alle drei Frameworks sind seit spätestens Anfang 2010 mit 3D-Funktionen ausgestattet worden. Prinzipiell ist also die Möglichkeit gegeben, interaktive 3D-Karten damit zu erstellen, auch wenn es keine expliziten „Geo-Funktionen“ gibt, wie die Unterstützungen von georeferenzierten Daten.

Silverlight

Silverlight ist in die .NET-Umgebung integriert. Um den frei erhältlichen Plug-in anzusprechen, reicht im Prinzip ein einfacher Texteditor, in den man beispielsweise XAML-Anweisungen eingeben kann, die dann vom Plug-in interpretiert werden. XAML (Extensible Application Markup Language, sprich: Semmel) ist eine auf XML basierende deklarative Sprache, die in Silverlight für das Design von User Interfaces verwendet wird. Die Funktionen des Interfaces sind in so genannten *code-behind*-Dateien organisiert, die in C# oder Visual Basic verfasst werden. Die Applikation kann dann als `<object>` in den HTML-Code einer Internetseite eingebunden werden. Microsoft stellt viel Trainingsmaterial und Tutorials online zur Verfügung. Es gibt auch Entwicklungsumgebungen, zum Beispiel Visual Studio, die teilweise gratis von den Silverlight-Seiten herunterladbar sind (Silverlight 2010).

AdobeFlex

Flex ist das von Adobe Entwickelte Framework für RIAs. Es ist im Prinzip der Nachfolger von Flash und die Flex-Applikationen können dementsprechend im Flash-Player angezeigt werden. Darin liegt auch einer der großen Vorteile von Flex: der Flash-Player genießt eine extrem hohe Verbreitung (siehe unten). Das Flex SDK wird von Adobe als Open Source Code bereitgestellt. Weiterreichende Authoring-Programme zum Erstellen von Webseiten können bei Adobe auch käuflich erworben werden.

JavaFX

JavaFX ist voll integriert in die Java Runtime und läuft in der Java Virtual Machine, einer relativ weit verbreiteten (siehe unten) und weit entwickelten Umgebung. Die Anwendungen für JavaFX werden in JavaFX Script verfasst, einer neuen Scripting-Sprache, die im Vergleich zu Java verstärkt graphische und Media-Inhalte unterstützt.

Die Entwickler können aber alle bereits bestehenden Java Libraries wieder verwenden. Dies kommt vor allem der großen Community der Java-Entwickler entgegen. JavaFX verbindet die Scripting-Sprache, die RichClient-Plattform und ein Tool-Set, um RIAs zu ermöglichen.

Die Marktdurchdringung der Plug-ins ist einer der Schlüsselfaktoren für den Erfolg eines der oben erwähnten Frameworks für RIAs. Für viele Web-Entwickler lohnt es erst dann, eine aufwändige Seite in einem der Formate zu gestalten, wenn ausreichend Internetnutzer auch über entsprechende Plug-ins verfügen, um in der Lage zu sein, solche RIAs auch tatsächlich zu verwenden. Die Statistiken unterscheiden sich im Einzelnen je nach Quelle, zeichnen aber ein identisches Gesamtbild. Nach Angaben von Adobe hat der Flash-Player eine Marktdurchdringung von weltweit 99 Prozent und selbst die neueste Version 10.1, die im Juni dieses Jahres (2010) herausgekommen ist, weist im September schon Penetrationswerte zwischen 66 und 76 Prozent in den Märkten der Industrieländer auf (Adobe 2010a). Das würde bedeuten, dass praktisch jeder Heim-PC mit einer mehr oder weniger aktuellen Version des Plug-ins ausgestattet ist und die Flex-Inhalte anzeigen kann. Die Marktdurchdringung bei Java beträgt laut Adobe (2010b) weltweit 79 Prozent. Laut DreamingWell.com (2010) liegt die Marktdurchdringung der Hauptversionen des Flex/Flash-Players bei 96 Prozent, von Java (als Voraussetzung für JavaFX) bei 72 Prozent und von Silverlight bei 65 Prozent (DreamingWell.com 2010). Aus diesen und anderen Quellen ist jedenfalls zu schließen, dass dies die drei wichtigsten Frameworks für RIAs sind. Der Flash-Player ist am weitesten verbreitet, gefolgt von Java. Auf dem dritten Platz liegt Silverlight, das bereits eine rasante Aufholjagd hinter sich hat.

Für die Erstellung des geologischen Modells im Rahmen dieser Arbeit als RIA sprechen die weite Verbreitung, relativ stabilen Laufzeiten und vor allem die immensen Möglichkeiten der Interaktivität, die mit den Frameworks umgesetzt werden können. Es gibt auch zahlreiche Beispiele für kartographische Anwendungen und Mash-Ups (Abbildungen 19 und 20). Auch selbst erstellte 2D-Karten und kartenverwandte Ansichten können sicher sehr gut in RIAs integriert werden. Jedoch steht in keiner der drei oben beschriebenen Plattformen die dritte Dimension im

Vordergrund und keins der Showcase-Beispiele macht exzessiven Gebrauch von echten 3D-Grafiken. Das lässt Fragen offen, in wie weit die für 3D essenziellen Funktionen der Grafikkarten unterstützt werden, wie beispielsweise Shader, der Zugriff auf die Hardwarebeschleunigung, wie die Vertices und Gitternetze umgesetzt werden etc. Es handelt sich nicht um spezialisierte Echtzeit-3D-Anwendungen und somit bleibt fraglich, ob die Leistungen der Systeme für komplexe und große 3D-Modelle ausreichen.

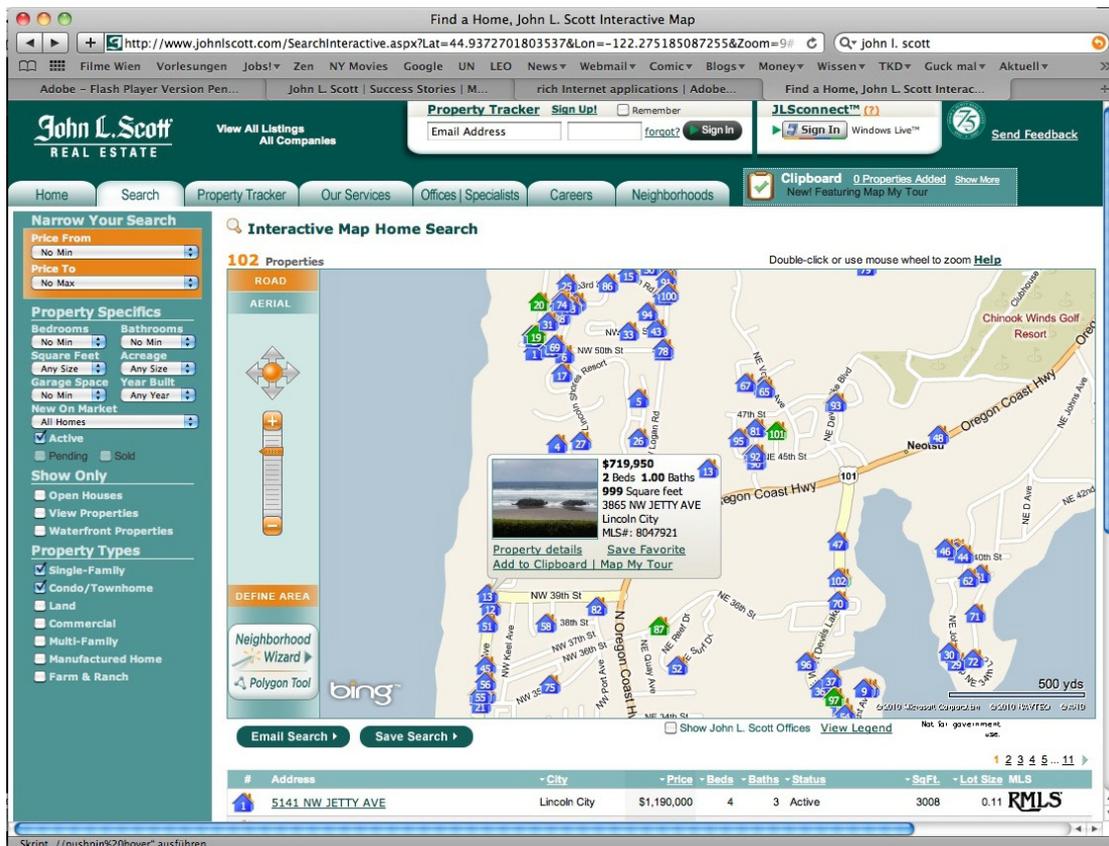


Abbildung 20: Kartographische RIA der Firma John L. Scott Real Estate als Mash-Up mit bing-maps. Quelle: www.johnlscott.com

3.3.4 eXtensible 3D

eXtensible 3D (X3D) ist ein lizenzfreies Dateiformat in Form eines offenen ISO-Standards, der vom Web3D-Konsortium gepflegt und weiterentwickelt wird. X3D ist der Nachfolger der relativ weit verbreiteten und von vielen 3D-Programmen unterstützten Virtual Reality Markup Language (VRML). Um die Web-Kompatibilität

und Interoperabilität zu steigern wurde VRML in ein auf XML (eXtensible Markup Language) basierendes Format umgestellt und auf X3D getauft. X3D ist genau wie VRML eine Sprache, in der Aufbau und Inhalte einer 3D-Szene (Szenengraph) beschrieben werden können (Listing 1).

```
<Scene>
  <Transform translation="1.5 2 -1">
    <Shape>
      <Appearance>
        <Material diffuseColor="1 1 0" />
      </Appearance>
      <Cylinder height="2" radius="1" />
    </Shape>
  </Transform>
</Scene>
```

Listing 1: Aufbau einer einfachen Szene in X3D. Die Farben zur besseren Kennzeichnung der Knoten und Felder basieren auf X3D-Edit

Eine Szenenbeschreibung wird immer von den <Scene>-Tags umschlossen. Daraufhin wird beschrieben, wo sich das Objekt befinden soll. Dazu dient der Transform-Knoten, der seinerseits die Geometrie umschließt. In das Feld *translation* des Transform-Knotens kann die Position des Objektes mit den Koordinatenwerten für die drei Raumachsen x, y und z eingetragen werden. Die Geometrie des Objekts wird zwischen den <Shape>-Tags definiert. Hier befinden sich Angaben über das Erscheinungsbild (Appearance) und die Form (hier: Cylinder). In den Feldern der entsprechenden Knoten können Angaben zu Farbe, Größe und Proportionen des Objektes gemacht werden. Das einfache Beispiel aus Listing 1 zeigt den prinzipiellen Aufbau eines Szenengrafen. Die Beschreibung des Szenengraphen kann jetzt von einem X3D-Player interpretiert werden, vorausgesetzt die Datei ist XML-konform, entspricht also der allgemeinen Grammatik von XML. Ist dies der Fall, so wird ein gelber Zylinder angezeigt (Abbildung 21).

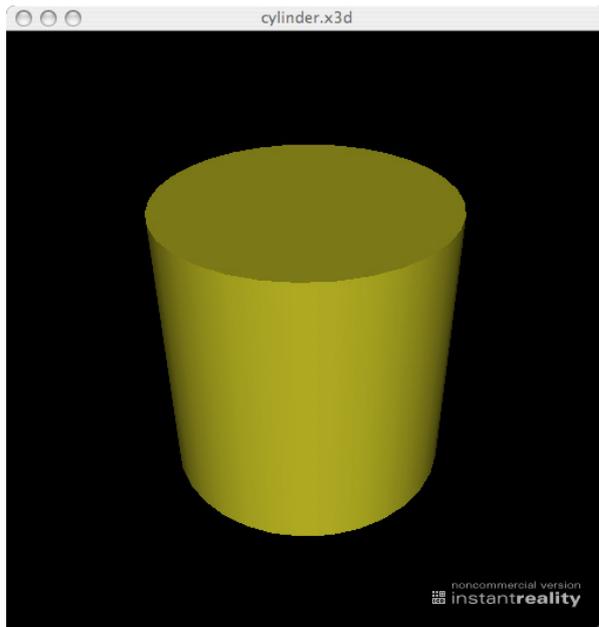


Abbildung 21: Das Ergebnis aus Listing 1, wie es von einem X3D-Player interpretiert wird: ein gelber Zylinder. Quelle: Eigene Abbildung

Schon VRML stellt sehr viel mehr Knoten zur Beschreibung von Geometrien zur Verfügung, als das Beispiel aus Listing 1 vermuten lässt. X3D wurde noch um einige bedeutende Grafikfunktionen wie NURBS, Partikelsysteme und Shader ergänzt. X3D ist flexibler als VRML, da verschiedene Profile gewählt werden können, die jeweils eigene Komponenten enthalten, um die Funktionen der Sprache so gut wie möglich an die Verwendung anpassen zu können. X3D ist rückwärts kompatibel zu vorigen Standards, sodass die VRML-Dateien immer noch interpretiert werden können. Gleichzeitig soll X3D durch seine Erweiterbarkeit, für die das X steht, für die Zukunft gewappnet sein. X3D hat den Anspruch ein Format zu sein, dass es ermöglicht einmal verfasste 3D-Szenen plattformunabhängig, also überall und jederzeit zu rendern. X3D ist auch ohne tiefere Programmierkenntnisse, ähnlich wie HTML, leicht zu erlernen. Beim Design wurde Wert gelegt auf Performanz, Geräteunabhängigkeit, Interaktivität und Internationalität (Web3D Consortium 2010). Dadurch erhofft sich das Web3D-Konsortium, dass die breite Masse der Web-Benutzer diese Technologie annehmen wird. Das Web3D-Konsortium betreibt eine breit angelegte Zusammenarbeit sowohl mit Industrie und Wissenschaft als auch normgebenden Institutionen wie beispielsweise der International Organization for Standardization (ISO) oder dem Open Geospatial Consortium (OGC). Der X3D-Standard ist bereits in

einer weltweit gültigen ISO-Norm festgelegt, und die Spezifikationen sind frei zugänglich.⁶

In der X3D-Earth-Workinggroup wird in enger Verbindung mit dem OGC an virtuellen Modellen der Erde gearbeitet, die Open Source und frei verfügbar sein sollen. Schon im Jahr 1997 beobachtete das Carto-Project der ACM SIGGRAPH (Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques) die Annäherungen der Kartographen-Community an die Virtuellen Welten von VRML (SIGGRAPH 2007), und es wurden Funktionen in den Standard integriert, die es erlauben, georeferenzierte Daten in VRML zu verwenden und GeoVRML entstand, woraus sich in Folge X3D-Earth entwickelte. Ab 1999 arbeitete sogar die Kommission für Visualisierung und Virtuelle Welten (Commission on Visualization and Virtual Environments) der Internationalen Kartographischen Vereinigung (ICA) in diesem Projekt mit (MacEachren et al. 2004).

Tabelle 1: Die verschiedenen 3D-fähigen Internet Formate und Anwendungen im Vergleich

	Inter-aktivität	Open Source	Geo-Funktionalitäten	Verbreitung von Plug-ins	GUI für Erstellung v. Inhalt	Interoperabilität/Austauschbarkeit
U3D	o	x	--	x	--	o
X3D	x	x	x	--	--	x
O3D	x	x	--	--	--	--
JavaFX	x	x	--	x	--	o
Silverlight	x	x	--	o	x	--
AdobeFlex	x	x	--	x	x	--
x: ausgeprägt o: mäßig ausgeprägt --: nicht vorhanden						

3.4 Beispiele aus der Wissenschaft

In diesem Abschnitt werden einige Beispiele für die Visualisierung von wissenschaftlichen Inhalten durch und über das Internet vorgestellt, um einen kleinen

⁶ <http://www.web3d.org/x3d/specifications/#x3d-spec>

Eindruck davon zu vermitteln, welche von den oben aufgeführten Methoden denn nun im wissenschaftlichen Bereich verwendet werden, welche Anforderungen an die Visualisierungen bestehen, und welche Ansprüche erfüllt werden können.

Subsurface Viewer/LithoFrame Viewer

Mithilfe von GSI3D (siehe Kapitel 2 Modellierung) sollen flächendeckende Modelle vom gesamten Vereinigten Königreich in verschiedenen Maßstäben (1:1Mio, 1:250 000, 1:50 000, 1:10 000) erstellt werden, die die Funktionen der traditionellen geologischen Karte vollständig übernehmen sollen. Die Modelle können dann auf unterschiedliche Weise an den Kunden gelangen. Sie können als Flash-Animation oder 3D-PDF aus dem Internet geladen werden, um eine Vorschau zu erhalten, die im Fall von 3D-PDF auch schon über einige Funktionen wie Navigation, Vergrößern/Verkleinern etc. verfügt. Zur interaktiven Betrachtung wurde aber ein extra Viewer in Java entwickelt, genannt *Litho Frame Viewer*. Es handelt sich um einen rich-Client, der 3D-Objekte mit OpenGL in Echtzeit darstellen kann. Er ist plattformunabhängig und erlaubt den Nutzern die Optimierung der Daten für das jeweilige Anwendungsgebiet. Weder ein GIS noch irgendeine andere Spezialsoftware ist vonnöten (Neber et al. 2006). Die Daten können als x,y,z-Punkte, als ASCII-Grids, als Esri SHP-Dateien, als Grids und als VRML-Oberflächen herunter geladen und in LithoFrame visualisiert werden. Auf dem heimischen PC können die Daten dann manipuliert werden. Synthetische Bohrlöcher und Profile können interpoliert werden, Höhenlinien berechnet und eine Explosionsdarstellung erstellt werden. Mit dem auch *Subsurface Viewer* genannten Programm können Datenbankabfragen durchgeführt werden und animierte GIF-Dateien angezeigt werden (Neber et al. 2006). In Zukunft soll es auch möglich sein, die Daten aus GSI3D in einem XML-Format zu exportieren.

Beispiel für eine einfache Visualisierungspipeline

Ein Ansatz, der sich stark auf die Visualisierung konzentriert und die Analyse mehr oder weniger außen vor lässt, wird von Baojun et al. (2009) vorgestellt. Sie verfolgen einen möglichst einfachen Ansatz um Kosten und Mühen zu sparen. Ihr verfahren besteht aus drei Komponenten. Die Daten werden in einem GIS aufbereitet. Danach werden sie mit einer 3D-Modellierungssoftware visualisiert und im Anschluss im

VRML-Format internetfähig veröffentlicht. Von allen Komponenten sollen nur die einfach zu bedienenden Funktionen verwendet werden.

Der erste Schritt betrifft vor allem die Interpolation von Bohrlochdaten. Die Daten werden als ASCII-Dateien in das GIS importiert. Die Datei enthält die x- und y-Koordinaten des Bohrlochs (2D), sowie die Höhe jeder Schichtgrenze. Mit Hilfe eines Interpolationsverfahrens werden die punkthaften Daten über die Höhenlagen der Schichtliniengrenzen in Raster umgewandelt. Es entstehen Graustufenbilder, die den Höhenverlauf der Grenzen als einen Grauwert zwischen 0 und 255 anzeigen.

Zusammen mit dem höchsten und dem niedrigsten absoluten Wert einer jeden Schicht, können die Raster im Raum positioniert werden. Damit wäre der erste Schritt vervollständigt. Für diese Etappe empfehlen Baojun et al. (2009) die Software ArcGIS von Esri aufgrund der vielfältigen Interpolationsverfahren, die dort zur Verfügung stehen. Außerdem sehen sie das Programm sowieso als eines der am weitesten verbreiteten Programme in diesem Bereich an.

Als nächstes findet der Modellierungsprozess statt. Dafür verwenden Baojun et al. (2009) die 3D-Software 3D Studio Max (3DSMax). Die Graustufenbilder, die den jeweiligen Schichtgrenzenverlauf repräsentieren, werden dazu herangezogen, die geologischen Körper zu modellieren. Dafür wird jeweils dasjenige Graustufenbild als Oberfläche des geologischen Körpers genommen, welches die obere Schichtgrenze darstellt. Und dementsprechend wird die untere Schichtgrenze für die Modellierung der Unterseite des Körpers herangezogen. Für alle im Modell vorhandenen Schichten wird diese Prozedur durchlaufen. Auf diese Weise werden einige Schichtgrenzen zweimal verwendet: als Unterseite einer der relativen Lage nach oberen Schicht, sowie als Oberseite der jeweils darunter liegenden Schicht. Jetzt können die Modelle der Schichtkörper lückenlos übereinander „gestapelt“ werden. Im Anschluss können in 3DSMax auch komplexere Vorkommnisse wie Linsen, Störungen, Faltungen etc. modelliert werden. Dafür können relativ einfach nutzbare Funktionen und Tools wie Boolesche Operatoren genutzt werden. Ist das gesamte Modell fertig gestellt, können Bilder verschiedener Ansichten oder Animationen abgespeichert werden. 3DSMax hält vielfältige Funktionen für Profilschnitte oder animierte Explosionszeichnungen und Blockbilder bereit. Die Interaktivität geht aber beim Exportieren verloren. Das interaktive Beschauen der Modelle ist nur in der Software selbst möglich. Um die

Interoperabilität zu steigern, empfehlen Baojun et al. (2009) den Datelexport ins VRML-Format (.wrl), der ohne zusätzlichen Aufwand direkt aus 3DSMax geschehen kann. Den Vorteil von VRML sehen Baojun et al. (2009) in mehreren Aspekten. VRML-Dateien können direkt in Internetbrowsern wie Internet Explorer angezeigt werden. Sie können in PowerPoint-Folien integriert werden, wo es auch möglich ist, das Modell interaktiv zu betrachten (Zoomen, drehen etc.). Außerdem können nicht nur statische Modelle ins VRML-Format exportiert werden, sondern die Animationen, die in 3DSMax erstellt wurden, können auch während der interaktiven Betrachtung der WRL-Datei ablaufen. Baojun et al. (2009) geben allerdings zu bedenken, dass VRML-Dateien nicht so gut für sehr große Datenmengen geeignet sind, finden aber die Darstellung von VRML-Modellen in 3D leicht und sehr effektiv für kleinere Regionen.

GA und X3D

Das australische Amt für Geologie (Australia's National Geological Survey, Geosciences Australia (GA)) veröffentlicht komplexe geologische Modelle seit dem Jahr 2001 im Internet (Beard et al. 2005). Das Ziel war die Veröffentlichung von 3D-Informationen in einem für eine breite Öffentlichkeit frei zugänglichen Format (Abbildung 22).

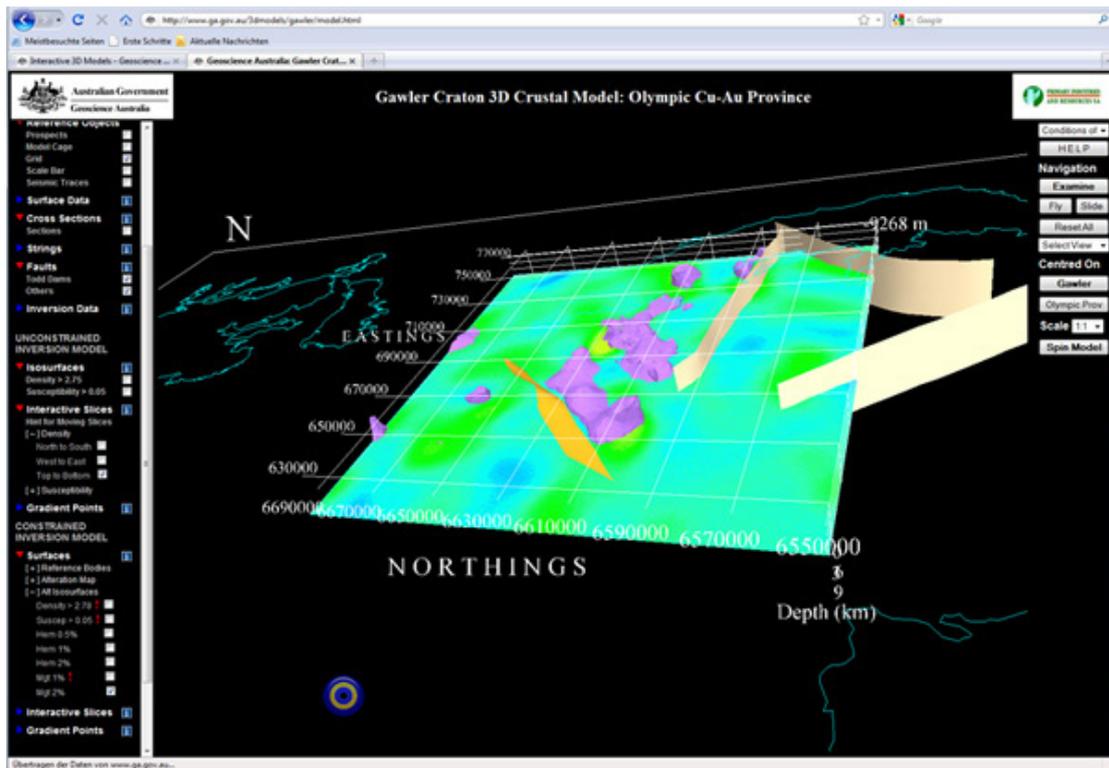


Abbildung 22: Ein online verfügbares 3D-Modell von GA, hier die Region Gawler Craton. Verschiedene geologische Informationen werden im Plug-in visualisiert, während die Steuerelemente als JavaScript-Schaltflächen im HTML-Bereich der Webseite definiert werden. Quelle: <http://www.ga.gov.au/resources/multimedia/3dmodels.jsp>

GA ist verpflichtet, der Regierung und den Behörden räumliche Informationen und Wissen als qualifizierte Entscheidungshilfe zur Verfügung zu stellen (Beard 2006). Dabei fordert eine 3D-Interpretation eine strengere, genauere Auslegung der Daten, da sich die 2D-Theorien auch in der dritten Dimension bewähren müssen. Das verlangt nach einer robusten zusammenhängen Interpretation (Beard et al. 2005). In den späten 90ern begann man bei GA mit VRML zu experimentieren. Die Datengrundlage waren exportierte Dateien aus Gocad oder damals auch ArcInfo7 und ähnlichen Experten-Programmen. In den Anfängen konnten mit VRML in Hinsicht auf die damals verbreiteten Hardware-Standards noch keine allzu komplexen Modelle erstellt werden (Beard 2005). Erst im Jahre 2001 wurde dann das erste komplexe Modell auf einer Konferenz vorgestellt, und seitdem von GA verwendet, um 3D-Modelle im Internet zu veröffentlichen. Das erste Modell deckte bereits ein sehr großes Areal des Offshore-Bereiches rund um Australiens Küsten ab. Das Modell bestand aus drei VRML-Dateien, die in HTML integriert waren. Obwohl noch einfach

im Aufbau, ermöglichte es schon erste Interaktivitäten (Zoomen, drehen) und mehrere Durchflüge. Weitere, vielschichtigere Modelle wurden von kleineren Regionen gemacht. Hier konnten die Betrachter bereits einzelne Layer ein- und ausschalten. Dies wurde durch die Integration mehrerer VRML-Dateien in ein einziges Interface ermöglicht (Beard et al. 2005). Im Jahre 2002 wurde das Modell im Internet veröffentlicht und bis 2005 folgten drei weitere (Beard et al. 2005). Mittlerweile ist ein hoher Grad an Interaktivität erreicht worden (Coffey et al. 2007). Der Inhalt der VRML-Dateien ist in HTML eingebettet und mit JavaScript-Funktionen angereichert worden (Coffey et al. 2007). Der HTML-Bereich enthält den Rahmen, in dem das 3D-Modell angezeigt wird, sowie eine Liste der Layer, den Titel und die Navigationsfunktionen. In den HTML-Teil sind die JavaScript-Funktionen für die Interaktivität und die Beeinflussung der VRML-Datei eingebettet. Einige Funktionen für die Interaktivität erfordern nämlich die Veränderung der VRML-Datei selbst (Beard 2006). Die VRML-Datei selbst besteht aus einer Hauptdatei der mehrere andere VRML-Dateien für jedes Geo-Objekt (Feature) zugeordnet sind. Die Hauptdatei, auch „Elterndatei“ genannt, bestimmt die Beleuchtung für das Modell und enthält Navigationseinstellungen, Blickwinkel und Informationen über die Animation. Sie leitet auch die JavaScript-Befehle an die untergeordneten Dateien weiter, die über die VRML-Knoten „Switch“ und „Inline“ dazugeschaltet und wieder ausgeschaltet werden können. Die Objektdateien werden meist aus Programmen wie Gocad oder Geomodeller exportiert oder aus anderen (beispielsweise Text-) Dateien überwiegend automatisch erstellt, während die Elterndatei, wie auch die HTML-Datei, manuell erstellt wird, was jedoch kein großer Aufwand ist (Beard 2006). Durch die Loslösung der Geo-Objekte von der Elterndatei können diese je nach den Bedürfnissen der Nutzer herunter geladen werden. Wenn also weniger Bandbreite zur Verfügung steht, muss nur ein kleiner Teil des gesamten Modells herunter geladen werden (Beard 2007). Zum Beispiel ist das vollständige Modell des 2009 im Internet zur Verfügung gestellten Radiometrischen Atlases von Australien 163 MB groß. Das Datenvolumen beim ersten herunterladen beträgt aber nur 8,52 MB. Die übrigen Dateien werden erst geladen, wenn sie von den Nutzern angefordert werden (Geoscience Australia 2009).⁷

⁷ Der Atlas steht übrigens nicht als VRML-Datei sondern schon im X3D-Format zur Verfügung.

Die Datengrundlage ist aus äußerst heterogenen Quellen zusammengefügt. Eine große Vielzahl unterschiedlichster Daten fließen in die Modelle ein. Coffey et al. (2007) berichten von einem Modell der unterseeischen Geologie. Bathymetrische Daten aus digitalisierten Karten und Sonarmessungen flossen darin ebenso ein, wie Landsat7-Satellitenbilder und SRTM-Oberflächenmodelle. Viele der Oberflächen wurden erst mit ERMapper erstellt und dann mit Gocad in Dreiecksnetze umgewandelt. GIS-Daten wie Straßen, Bahnlinien etc. wurden als Vektordaten in das Modell integriert. Sedimentproben wurden in ihrer genauen 3D-Position eingetragen. Um einige davon zu visualisieren wurden Kuchendiagramme mithilfe einer Spezialsoftware erstellt, die tabellarische Daten in das VRML-Format umwandelt. Orte, an denen Proben entnommen wurden, Epizentren von Erdbeben und Mineralische Ablagerungen werden als Punkte dargestellt; Grenzen und seismische Interpretationen als Linien; Profilschnitte und seismische Profile als Flächen, die als 2D-Profile in das Modell „eingehängt“ werden, sodass ein realistischer Eindruck der stratigraphischen Schichten entsteht (Coffey et al. 2007); digitale Höhenmodelle und Störungsflächen als triangulierte Oberflächen; komplexe Felsformationen und dreidimensionale geophysikalische Datenwürfel als Volumenkörper; Bohrlochschächte und ähnliches als Zylinder und andere 3D-Objekte; und Rasterdaten und Satellitenfotos als Bilddateien. Auf diese Weise kann ein einziges 3D-Modell 2600 einzelne VRML-Dateien beinhalten (Beard et al. 2005). GA konnte einige sehr innovative Funktionen in die VRML-Modelle einbauen. In einigen Modellen sind Attributdaten enthalten, wie zum Beispiel Namen von Mineralien, Faltungen oder geochronologischen Abschnitten, die über eine Datenbankabfrage per Mausklick angezeigt werden können (Beard et al. 2005). Ein PHP-Befehl führt die Abfrage durch, und es werden HTML-Tabellen zurückgegeben (Beard 2006). Des Weiteren wurde eine Möglichkeit entwickelt, Profilschnitte durch interaktives *Slicing*, also schneiden durch das Modell, zu erstellen (Beard 2006).

Um eine große Datenfülle, wie sie einige sehr komplexe Modelle mit vielen Interaktionsmöglichkeiten aufweisen, über das Internet zur Verfügung stellen zu können, werden eigens aufgestellte Leitlinien befolgt. Die Modelle sollen eine bestimmte Dateigröße nicht überschreiten. Es werden Zusatzprogramme zum Komprimieren der Daten verwendet. Anstatt die einzelnen Dreiecke einer Oberfläche

einzufärben, um Karten oder Satellitenbilder anzuzeigen, sollten sie lieber als JPG, GIF oder ähnliche Dateiformate auf die Oberfläche projiziert werden. Um Ladezeiten zu minimieren, werden nur die angefragten Layer aus dem Internet übertragen. Anfangs werden nur die Grundlagen eines Modells angezeigt. Die Nutzer können dann über ein Interface einfach weitere Daten auswählen, die nachfolgend in das Modell eingefügt werden (Beard et al. 2005). Coffey et al. (2007) erstellten für einige Oberflächen drei verschiedene Genauigkeitsstufen, sodass die Nutzer zwischen einer hohen, mittleren und niedrigen Auflösung wählen konnten, je nach zur Verfügung stehender Bandbreite. Die Oberflächen wurden im Spezialprogramm Gocad als Dreiecksnetze erstellt, bei denen flache Gebiete weniger, dafür größere Dreiecke aufweisen als Gebiete mit ausgeprägter Reliefstruktur, für die kleinere aber dafür mehr Dreiecke verwendet werden. Allein das hilft schon ungemein bei der Datenreduzierung (Coffey et al. 2007). Allerdings berichten Coffey et al. (2007) auch von signifikanten Problemen, die die Mischung verschiedener Auflösungen erzeugt. Artefakte in der Höhendarstellung treten auf, die nur schwer zu beseitigen sind. Um die Datenmenge klein zu halten, können die VRML-Dateien im gzip-Verfahren komprimiert werden und so allgemein recht kleine Dateien erzeugt werden, die sich gut über das Internet verbreiten lassen, auch bei geringer Bandbreite (Coffey et al. 2007). Coffey et al. (2007) halten VRML für einen effektiven Weg, auch große Datenmengen über das Internet einer breiten Öffentlichkeit zur Verfügung zu stellen. GA stellt gerade von VRML auf X3D um, was einige Verbesserungen in den Bereichen Datenbankabfragen, Interoperabilität mit anderen XML-Formaten und breitere Unterstützung durch 3D-Viewer Plug-Ins und Software mit sich bringt. Auch die Kompressionsfähigkeit wird im X3D-Format deutlich höher eingeschätzt (Beard 2006).

Teil II

Umsetzung

4 Anforderungen an die Visualisierung

Die Wiener Gewässermanagement Gesellschaft mbH (WGM) will in einem Forschungsprojekt Wege und Mittel identifizieren, mit denen sie ihre Daten Experten und Entscheidungsträgern aus anderen Magistratsabteilungen der Stadt Wien zur Verfügung stellen kann. Es handelt sich dabei erst einmal um geologische Daten, die aber im Laufe der Zeit mit allen erdenklichen anderen Daten der Stadt Wien ergänzt und erweitert werden sollen. Nicht nur digitale Geländemodelle, sondern auch Gebäudemodelle aus Laserdaten, Infrastrukturelemente wie U-Bahnschächte oder Luft- und Satellitenbilder wären denkbar. Es sollen also die unterschiedlichsten Datenquellen und -formate nutzbar gemacht werden, die alle möglicherweise in jeweils spezialisierter Expertensoftware erstellt wurden. In Bezug auf die geologischen Daten der WGM wurde schnell deutlich, dass nur die wenigsten der potenziell interessierten Mitarbeiter des Magistrats über die notwendigen, speziellen Computerprogramm-Lizenzen verfügen, um sich die Daten auch anschauen zu können. Einer effektiven Verbreitung der Daten steht dies natürlich im Wege. Ebenso von Nachteil ist unter Umständen die fehlende Möglichkeit der Integration von Daten, die etwa aus Laservermessungen erhoben wurden, in die Modelle der Geologen. Wenn unterschiedliche Software und die damit einhergehenden proprietären Datenformate zum Einsatz kommen, sind die Informationen nicht ohne weiteres in einem einzigen Modell zusammenführbar. Es wird also ein Visualisierungskonzept benötigt, das einen geeigneten Standard für geo-3D-Daten beinhaltet, in den alle Datentypen der unterschiedlichsten Modelle möglichst ohne Informationsverlust überführt werden können. Die Verwendung dieses Standards sollte möglichst frei zugänglich und ohne große Kosten zu verwenden sein, um Konflikte bei der Anschaffung von Softwarelizenzen oder Spezialprogrammen zu vermeiden. Schließlich soll eine einfache Verbreitung der 3D-Daten ermöglicht werden in einem Netzwerk von räumlich getrennten Nutzern. Das legt eine Nutzung der ubiquitär zur Verfügung stehenden Internettechnologie nur allzu Nahe.

Nachdem die Rahmenbedingungen, die erfüllt werden müssen, geklärt sind, werden Anforderungen an die interaktive Funktionalität der bildlichen Darstellung der Daten gestellt. Sie beziehen sich im Rahmen dieser Arbeit hauptsächlich auf den Fall der

Visualisierung der geologischen Daten der WGM und betreffen die Interaktionsmöglichkeiten, die in Tabelle 2 aufgelistet sind.

Tabelle 2: Anforderungskatalog der WGM an die Visualisierung der geologischen Daten

	notwendig	zusätzlich
Darstellung der Daten und Handhabung allgemein		
Visuelle, räumliche Darstellung von 2,5d Raster-Daten in den vorliegenden Datenformaten (Erdas Img, Geotiff, Ascii Raster etc.) als auch von Vektordaten (Shape-Files) mittels Webbrowser, plattformunabhängig in einer gängigen Programmierumgebung	X	
Wählen der Beleuchtungsquelle (Sonnenstand und Richtung) (evtl. vordefinierte Werte: Beleuchtung aus N, NW, W, SW ...)		X
Ein- und ausblenden von Datenlayern – Raster als auch Vektordaten – aus einer Layertabelle	X	
freies Rotieren im Raum (x-, y- und z-Rotation)	X	
freies Heran- und Herauszoomen	X	
manuelle Bestimmung des Drehpunktes	X	
interaktive Steuerung der Transparenz einzelner Layer	X	
3d-Steuerung per Maus (z.B. Zoomfaktor mittels Mause, drehen mittels „festhalten“ an einer Ecke etc.)	X	
Überlagern von referenzierten „Bildern“ auf ein Höhenmodell (z.B. das Orthofoto auf das Geländemodell legen)	X	
Entfernungsabhängiges Ein- bzw. Ausblenden von Textinformationen (z.B. Straßennamen, Bohrbezeichnungen etc.)		X
Zuweisen von Farben für Rasterdaten – Color Ramp (rasterwertabhängige Einfärbung)		X
freie Wahl der Hintergrundfarbe des Viewers		X
Darstellung von Bohrungen als Säulen (entsprechend der Bohrtiefe)		X
Neben der Überhöhung zusätzliches „Strecken“ einzelner Schichten in z-Richtung (um besser zwischen die Schichten sehen zu können)		X
Frei wählbare Kontrasteinstellung		X
Abfragemöglichkeiten durch Mausklick auf Position		
Rasterdaten: Rasterwert (z.B. Höhenwert) eines gewählten Layers	X	
Rasterdaten: tabellarische Auflistung aller Werte der sichtbaren Layer vom Klickpunkt senkrecht nach unten		X
Vektordaten: Abfrage eines vordefinierten Attributes eines Layers (z.B. Tiefe eines Bohrpunktes)	X	
Vektordaten: Abfrage von Attributdaten (z.B. Informationen über POI)	X	

Erweiterte (Analyse-) Funktionen		
Eingabe eines Punktes mittels x-, y-Koordinaten (als Kennzeichnung eines Standortes – z.B. Lokalität eines Bauvorhabens)	X	
Abfrage von Höhendaten entlang einer interaktiv eingegebenen Linie (Profilschnitterstellung)		X
Verlinkung bestimmter Dokumente an bestimmten, dargestellten Punkten (Points of Interest – POIs) – z.B. Klick auf einen speziellen Punkt führt zum Download von PDF-Dokumenten oder öffnet ein verlinktes Foto oder eine Animation etc.)		X

Dieser Anforderungskatalog enthält viele Funktionen, die eigentlich Geographischen Informationssystemen (GIS) eigentümlich sind, wie zum Beispiel Attributabfragen, Koordinateneingaben und ähnliche Datenbank-Abfragemöglichkeiten mit Raumbezug. Wie die Erörterung der aktuell zur Verfügung stehenden technischen Mittel für Visualisierungen im Internet zeigt (siehe Kapitel 3 Visualisierungsmöglichkeiten mit Web-Technologie), sind dies sehr hohe Ansprüche an die vorhandenen Software-Pakete und Datenformate. Hier wird zudem das Spannungsfeld zwischen Expertensoftware und der Visualisierung für ein breiteres Publikum sichtbar. Zu viele und zu komplexe Funktionen können einen Laien schnell überfordern und sind damit nicht mehr effektiv (ebenfalls in Kapitel 3 erörtert). Ein großer Vorteil der 3D-Darstellung könnte verloren gehen, nämlich der der intuitiven Begreifbarkeit des Gezeigten. Diese Bedenken sind durchaus in ein umfassendes Visualisierungskonzept mit einzubeziehen.

für die es keine Daten gibt, hier NODATA-Wert genannt. Dieser Wert entspricht keinem real vorkommenden Wert: hier: -9999, was weit unterhalb des tiefsten Punktes der Erdoberfläche liegt und deshalb wohl in den meisten Anwendungen gefahrlos als NODATA-Wert verwendet werden kann. Dieser Wert soll nicht dargestellt werden. In diesem Fall können mit dem Wert Gebiete innerhalb des strikt rechteckigen Rasters beschrieben werden, in denen die entsprechende Schicht nicht vorkommt. Gefolgt werden diese sechs Zeilen von einer langen Liste mit Höhewerten, für jede Zelle ein Wert. Das entsprechende Computersystem kann nun den Raster Zeile für Zeile aufbauen, indem er die Zahlenkolonne einliest und nach jedem n-ten Wert (entsprechend der Angabe im Header, Zeile eins) eine neue Zeile anfängt, bis die entsprechende Zeilenanzahl erreicht wurde (angegeben im Header, Zeile zwei).

Die geologischen Daten

Die geologischen Schichtenmodelle wurden aufgrund der Bohrkerndaten von den Experten der WGM interpoliert und in zwei Formaten exportiert. Einerseits lagen die geologischen Schichten als Raster im ASCII-Format vor. Zusätzlich zu den ASCII-Rastern wurden die Vorkommen der (diskontinuierlichen) geologischen Oberflächen in Grafikdateien im TIF-Format beschrieben. Diese Rasterdaten hatten eine Auflösung von 15 m. Sie standen für fünf Schichten zur Verfügung (siehe Tabelle 3). Es wurden also keine geologischen Volumenkörper modelliert. Die Datenlage entspricht dem Verlauf der Oberkannten (Hangendes) der geologischen Formationen.

Tabelle 3: Die Schichten des geologischen Modells in ihrer generellen Reihenfolge.

Schicht	Darstellung im Modell
Wienerwaldschotter	Grün
Interkolluvium	Pink
Junger Donauschotter	Gelb
Grundwasser	Blau
Miozänoberkannte (Wasserstauer)	Lila

Die Lagen der Bohrungen, mit deren Hilfe die Schichten interpoliert wurden, lagen als Punktdatensatz mit ihren 2D-Koordinaten im Shape-Format vor. Zu jedem Punkt

enthielt die Attributtabelle jeweils Angaben zur Teufe (Tiefe eines Bohrkerns). Auch die Lage und Ausdehnung der Gruben lag im Shape-Format als Polygondatensatz vor, während die Höhenschichtlinien als Datensatz mit Linien im Shape-Format zur Verfügung standen, sodass die drei wichtigen Vektorformate zum Einsatz kamen.

Das Orthofoto

Beim Orthofoto handelte es sich um eine Farbaufnahme des Testgebiets, die wurde in einer Auflösung von 1,5 m pro Pixel als JPG-Bild vorlag. Das Gesamtvolumen entsprach mit 1650x5050 Pixeln 2,3 MB. Aus Auflösung und Pixelanzahl lassen sich leicht die Dimensionen des Bildes ableiten, die einer Breite von 2475 m und einer Länge von 7575 m entsprachen und damit das gesamte Testgebiet abdeckten.

Das digitale Geländemodell

Das DGM hatte eine Auflösung von 5 m und deckte ebenfalls das gesamte Testgebiet ab. Auch dieses lag im ASCII-Format vor.

SRTM-Daten

Weitere Höheninformationen über das übrige Stadtgebiet Wiens lagen ebenfalls im ASCII-Format vor. Sie wurden aus SRTM-Daten erstellt und hatten eine Auflösung von ca. 85 m.

Die OpenStreetMap-Daten

Die Daten für den Stadtplan zur besseren Lagebestimmung des Testgebiets stammen von OpenStreetMap. Sie konnten über Geofabrik bezogen werden.⁸ Die Daten werden dort fast täglich aktualisiert und liegen unter anderem im Shape-Format vor, sodass sie relativ einfach mit GIS-Software weiterverarbeitet werden können (Geofabrik 2010). Die Shape-Daten sind in verschiedenen thematischen Layern organisiert von denen in dieser Arbeit nur der Straßen-Layer verwendet wurde.

⁸ <http://download.geofabrik.de> (Download-Server)

6 Die qualifizierte Entscheidung für X3D

Aus der breiten Palette der in Kapitel 3 Visualisierungsmöglichkeiten mit Web-Technologien beschriebenen Mittel, die für diese Projekt in Frage kommen, wurde X3D für die Umsetzung ausgewählt, da es für die Erfüllung der genannten Anforderungen die beste Voraussetzungen hat. Dafür sprechen sowohl gute Rahmenbedingungen als auch praktische Gründe.

6.1 Gute Rahmenbedingungen

6.1.1 Interoperabilität

Ein wichtiger Punkt bei der Überlegung den Datenaustausch betreffend ist die Interoperabilität. Trotz der großen Flexibilität ist sie in X3D gesichert, denn die die möglichen Erweiterungen spielen sich alle im XML-Format ab, einem allgemeingültigen ISO-Standard auf dem X3D basiert. XML steht für eXtensible Markup Language und hat bereits viele Dialekte ermöglicht, die alle nach dem gleichen Prinzip erstellt wurden, wie zum Beispiel CityGML. Es ist dadurch relativ einfach, Dateien von einem ins andere Format zu verwandeln, wie es beispielsweise im W3DS gemacht wird (siehe Kapitel 3 Visualisierungsmöglichkeiten mit Web-Technologie). Etwas allgemeiner gefasst soll die Interoperabilität auch durch den Status von X3D als weltweit gültige Norm für 3D-Grafik im Internet gesichert werden. Über den Erfolg der Anerkennung von X3D als ein Standard mit solch einer breiten Unterstützung, wie sie beispielsweise das JPG-Format für Bilddateien erfährt, lässt sich allerdings bisher noch streiten (siehe unten).

6.1.2 Offenheit des Standards

Es ist jedenfalls zu betonen, dass es sich um einen offenen ISO-Standard handelt, der es jedem Entwickler erlaubt, eigene Software für X3D zu kreieren (zum Beispile eigene Viewer), ohne dabei Konsortiumsmitglied oder ähnliches sein zu müssen. Auch die von Autoren erstellten Inhalte sind und bleiben Eigentum der Autoren selbst und unterliegen keiner Lizenz oder Gebührenordnung. Dies entspricht auch den Anforderungen der WGM (siehe oben). Des Weiteren soll der offene Status des Standards zur Haltbarkeit der Produkte verhelfen. Er ist nicht an eine bestimmte

Firma gebunden, die die Weiterführung ihres 3D-Formats einstellen könnte, so wie beispielsweise mit Google und O3D geschehen (siehe Kapitel 3 Visualisierungsmöglichkeiten mit Web-Technologie Abschnitt O3D). Die noch vor zwei, drei Jahren in O3D erstellten Inhalte sind heute nicht weiter verwendbar. Die Rückwärtskompatibilität von X3D soll dagegen garantieren, dass auch die schon Mitte der 90er Jahre erstellten 3D-Szenen im VRML-Format heute noch mit den aktuellen Player-Versionen gezeigt werden können. Die Haltbarkeit beträgt also schon jetzt gut 15 Jahre, und ein Verfallsdatum ist nicht in Sicht. Da das Forschungsprojekt der WGM auf zehn Jahre ausgelegt ist, wäre es sicherlich von Vorteil, einen durchgängigen Standard zu haben, um nicht in die Situation zu geraten, dass die Daten von den Anfängen des Projekts am Ende nicht mehr verwendet werden können.

6.1.3 Breite Akzeptanz

Die hinter X3D stehende Technologie existiert in der Form des Vorläufers VRML, wie bereits erwähnt, schon seit den 90er Jahren. Trotzdem wurde bis jetzt keine übermäßig beeindruckende Verbreitung von 3D-Inhalten in diesem Format gezeitigt, was Kritiker dazu bewegt, VRML/X3D als „tote“ Technologie zu bezeichnen. Laut Anita Havele, der geschäftsführenden Direktorin des Web3D-Konsortiums, befindet sich X3D aber auf dem „Pfad der Erleuchtung“ (*Slope of Enlightenment*), einer Phase des Hype-Zyklus von Gartner⁹ (Abbildung 23). Demnach wäre mit einer weiteren

⁹ Der Pfad der Erleuchtung ist die vierte von insgesamt fünf Phasen des so genannten Hype-Zyklus. Der Zyklus beginnt mit der Einführung der Technologie. Diese wird in der zweiten Phase in einer extremen Euphorie (Überschätzung) angenommen („Gipfel der überzogenen Erwartungen“), bis sich herausstellt, dass die Erwartungen (noch) nicht erfüllt werden können und die Kurve der Aufmerksamkeit jäh abstürzt in das „Tal der Desillusionierung“ (Phase drei). In dieser Phase werden die nicht erfolgreichen Implementationen der Technologie ausgesiebt und nur die wenigen erfolgreichen verbleiben. Sie werden weiterentwickelt, und obwohl die Presse das Thema nicht weiter behandelt, bleiben ein paar Interessierte an der Technologie dran. Der vermehrte Erfolg lässt das Interesse der Öffentlichkeit wieder ansteigen und stabile Folgegenerationen kommen auf den Markt. Diese Phase wird „Pfad der Erleuchtung“ genannt, der in das Plateau der Produktivität mündet (fünfte Phase). Auf diesem Niveau pendelt sich der Erfolg in Abhängigkeit von der allgemeinen Anwendbarkeit der Technologie ein. Quelle:

<http://www.gartner.com/pages/story.php.id.8795.s.8.jsp>.

Steigerung der Beliebtheit und Verbreitung von Anwendungen von X3D zu rechnen. Die Krisenphasen hätte X3D damit auch schon durchschiffert und hinter sich gelassen.

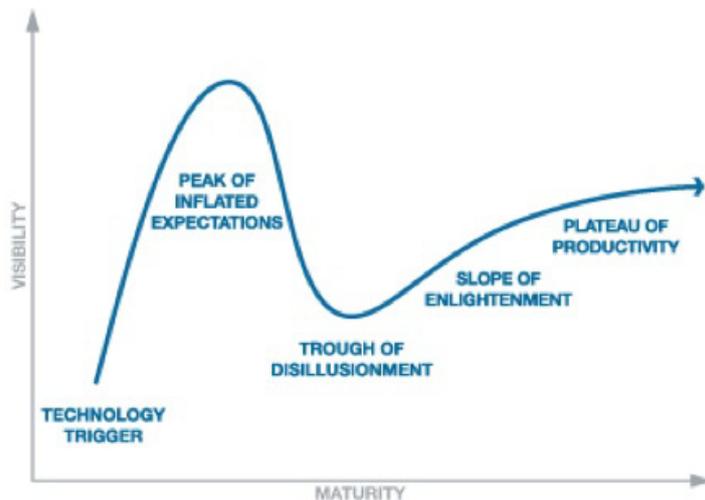


Abbildung 23: Der Hype Zyklus, entwickelt von der Beraterfirma Gartner.
Quelle: <http://www.gartner.com/pages/story.php.id.8795.s.8.jsp>

6.1.4 Verschlüsselung und Kompression

Ein weiterer positiver Aspekt, der im Rahmen des Forschungsprojektes der WGM von Interesse sein könnte, ist die Möglichkeit X3D-Daten zu verschlüsseln und zu komprimieren. Dies wird möglich aufgrund der XML-Basis, die eine Sicherung der Daten nach Empfehlungen des WorldWideWeb–Consortium (W3C) erlaubt. Sollte in Erwägung gezogen werden, die Visualisierung auch der Öffentlichkeit zugänglich zumachen, ist es oft aus rechtlichen Gründen geboten, die Weiterverwendung der Daten, die über das reine Betrachten hinausgeht, zu verhindern. Die Visualisierung darf also veröffentlicht werden, die dahinter stehenden Modelle und Daten aber nicht. Verschlüsselungsverfahren könnten hierbei eine entscheidende Rolle spielen.

Für die Bereitstellung großer Datenmengen könnten die Komprimierungsverfahren, die die X3D-Technologie zulässt, für das Datenstreaming eine bedeutende Rolle einnehmen. Um Übertragungsraten und Speicherkapazitäten nicht zu überlasten könnten die Daten großer Szenen nach Bedarf schnell aus dem Internet bezogen werden, sobald sie im Viewer angesteuert werden. Verschwinden sie wieder aus dem Sichtbereich, werden sie wieder entladen.

6.1.5 Erfahrung aus dem wissenschaftlichen Bereich

Neben diesen eher technischen Gründen, die bei der Entscheidung für X3D ins Gewicht fielen, spielt auch die Erfahrung Anderer mit diesem Format eine Rolle. X3D wird in vielen Fällen im wissenschaftlichem Kontext verwendet. Das zeigen die Beispiele der Interaktiven 3D-Modelle von Geoscience Australia (Abbildung 22) oder vom Monterey Bay Aquarium Research Institute. Auch ein Blick auf die Liste der Mitglieder und Unterstützer des Web3D-Konsortiums, das für die Weiterentwicklung und Pflege des Standards verantwortlich ist, spiegelt das große Interesse der Wissenschaft an X3D wider. Darunter finden sich nicht nur viele Universitäten, wie die Naval Postgraduate School oder die University of Suwon, Korea, sondern auch Organisationen wie NASA und das Fraunhofer Institut. In diesem Zusammenhang ist auch noch einmal auf den Web3D Service hinzuweisen, der für die Visualisierung von Geodaten ja auch auf den X3D/VRML-Standard zurückgreift und bereits Implementationen aus dem wissenschaftlichen/verwaltungstechnischen Bereich online gestellt hat.

6.1.6 Einwände

Als Nachteil ist allerdings zu sehen, dass weder Apple noch Google derzeit Mitglieder des Web3D-Konsortiums sind, obwohl beide Firmen mehrmals ein- und wieder ausgetreten sind. Zwar hat das Web3D-Konsortium nach eigenen Angaben den Dialog mit Google gesucht, aber nicht gefunden (Web3D Consortium 2010). Auch Microsoft ist kein Mitglied des Konsortiums, hat allerdings große Beiträge zu HTML5 geleistet, einem Standard des W3C, mit dem das Web3D-Konsortium wiederum eng zusammenarbeitet, um X3D besser in HTML-Seiten zu integrieren.

6.2 Praktische Gründe

6.2.1 Vielfältigkeit

Ein weiterer wichtiger Aspekt bei der Wahl des geeigneten Mittels ist die Vielfältigkeit der Anwendungsmöglichkeiten. Je mehr Möglichkeiten ein Werkzeug bietet, desto höher ist die Wahrscheinlichkeit, dass die passende Funktion für eine bestimmte Aufgabe zur Verfügung steht. X3D stellt extrem viele verschiedene Arten

der Darstellung von 3D-Geometrien zur Verfügung, die allen drei Möglichkeiten der Computer-3D-Repräsentation (Siehe Kapitel 2 Modellierung) entgegen kommen. Es besteht die Möglichkeit, ein komplexes Objekt aus mehreren primitiven Geometrien zusammenzusetzen, ähnlich den Binarytrees der Constructive Solid Geometry-Methode. Es stehen die primitiven Formen des Quaders, des Kegels und der Sphäre zur Verfügung, die beliebig zusammengesetzt und in Proportionen und Größe verändert sehr komplexe Objekte ergeben können. Damit wäre theoretisch sogar eine Cell Decomposition möglich (als Sonderfall der Constructive Solid Geometry), auch wenn sich der erhebliche Aufwand insbesondere für größere Datenmengen im X3D-Format sicher nicht rechnen würde. Grundlegend aber für die Darstellung von 3D-Objekten in X3D ist die Methode der Boundary Representation (B-rep), mit deren Hilfe auch die geometrischen Primitive konstruiert sind (Abbildung 24).

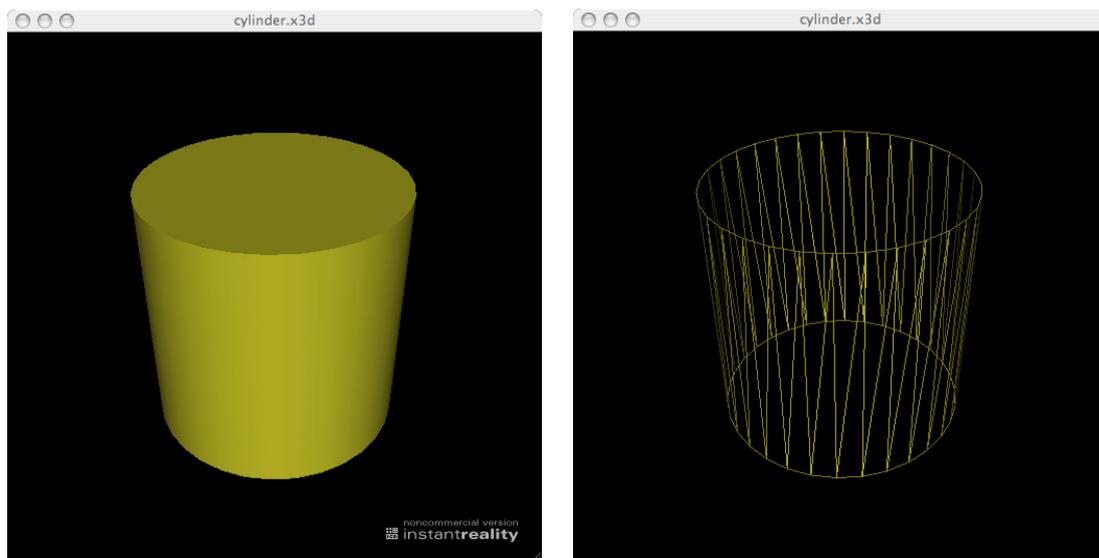


Abbildung 24: Der Aufbau des Gitternetzes des in Listing 1 beschriebenen Zylinders. Das Geometrische Primitiv wird mit Hilfe von B-reps aufgebaut, die das Innere des Objekts "leer" lassen. Quelle: Eigene Abbildung

Wie bereits erwähnt könnte für geologische Volumenkörper auch die Attributierung des Inneren eines solchen Körpers notwendig werden. Im Falle dieser Arbeit sollten aber ausdrücklich keine solchen Körper modelliert und dargestellt werden, weswegen dieser Nachteil wegfällt. Der Vorteil der B-reps ist es, dass sie der Grafikverarbeitung der Hardware entgegenkommt, die letztendlich jede Geometrie in Dreiecksnetze umwandelt, bevor sie sie in ihren virtuellen 3D-Raum projiziert. Neben den X3D-

Knoten zur Erstellung von Primitiven werden auch viele andere Funktionen unterstützt, wie beispielsweise Extrusion und NURBS, mit deren Hilfe sich wiederum sehr komplexe Oberflächen gestalten lassen. Für diese Arbeit ausschlaggebend war der ElevationGrid-Knoten von X3D. Das ElevationGrid kommt der Darstellung von Rasterformaten sehr nahe, indem eine regelhafte, rechtwinklige Anordnung von Datenpunkten eingerichtet wird. Die Anzahl der Datenpunkte, sowie ihr Abstand zueinander werden vom Entwickler, im X3D-Jargon auch *Worldbuilder* genannt, festgelegt. Jedem Punkt kann ein Höhenwert zugeordnet werden. Der Darstellung von aus TINs bestehenden Oberflächen kommen die Knoten IndexedFaceSet oder TriangleStripSet entgegen, die eine flexible Gestaltung von aus Polygonen bestehenden Oberflächen zulassen.

Reicht die Vielzahl der Knoten, die im X3D-Standard definiert sind, dennoch nicht aus, so besteht grundsätzlich die Möglichkeit einer eigenen Erweiterung der Beschreibungssprache. Dies wird durch das XML-Format ermöglicht und die Unterstützung von JavaScript und der Programmiersprache Java. Es können also sowohl eigene Knoten definiert werden, als auch die Funktionen des bestehenden Standards erweitert werden unter Wahrung der Interoperabilität.

Diese Vielfältigkeit legte bei den Überlegungen zur praktischen Umsetzung die Vermutung nahe, dass X3D die notwendigen Voraussetzungen erfüllt, die sehr unterschiedlichen Datenformate (siehe Kapitel 6 Datengrundlage) in einem Rahmen zu vereinen

6.2.2 Unterstützung von Geodaten

Viele dieser Knoten sind auch in der X3D Earth-Komponente implementiert, die die Verarbeitung von georeferenzierten Datensätzen erlaubt. Auch dies ist ein Aspekt der im Rahmen der Umsetzung des vorliegenden Modells für X3D sprach. X3D ist einer wenigen 3D-Standards, die Geofunktionalitäten aufweisen (siehe Tabelle 1).

7 Der Prozess der Visualisierung in X3D

7.1 Erstellung der Schichtenvisualisierung

Die Informationen, die in den ASCII-Dateien enthalten sind, reichen vollends aus, um mit dem X3D-Knoten ElevationGrid eine 3D-Oberfläche¹⁰ zu erzeugen. Die Informationen aus dem Dateiheder der ASCII-Dateien werden in die entsprechenden Felder des X3D-Knotens eingetragen (siehe Tabelle 4).

Tabelle 4: Die korrespondierenden Felder der ASCII-Dateien und des X3D-Knotens ElevationGrid

ASCII	ElevationGrid
ncols	xDimension
nrows	yDimension
cellsize	xSpacing
cellsize	ySpacing
z-Werte	height (y-Werte)

Die Lage des ElevationGrids im Raum, also der Rechts- und Hochwert der Eckkoordinaten (xllcorner, yllcorner), werden nicht direkt in die Felder des ElevationGrid-Knotens eingetragen, sondern können der X3D-Struktur entsprechend in dem das ElevationGrid umfassenden Transform-Knoten angegeben werden. Dies muss aber nur dann geschehen, wenn die Raster unterschiedliche Eckkoordinaten aufweisen, da durch die Default-Einstellungen in X3D, die in Kraft treten, wenn im Transform-Knoten keine Angaben gemacht werden, alle ElevationGrids im Ursprung des Koordinatensystems beginnen. Die eigentlichen Werte, die die Höhe der einzelnen Rasterzellen beschreiben, werden durch Copy-and-paste aus dem ASCII-File in das Feld „height“ des ElevationGrids eingefügt.

¹⁰ Manchmal werden diese Oberflächen auch als 2,5D-Oberflächen bezeichnet. Im Wort „Oberfläche“ ist allerdings schon die Flachheit des Objekts ausgedrückt, im Gegensatz zum Begriff (Volumen-)Körper. Da in beiden Fällen mit 3D-Koordinaten-Tupeln gearbeitet wird, wird in dieser Arbeit nicht zwischen 2,5 und 3D unterschieden.

Für durchgängige, rechteckige Rasterdaten ohne NODATA-Werte umfasst dieses simple Verfahren die vollständige Übertragung einer ASCII-Datei in ein X3D-ElevationGrid (Abbildung 25).



Abbildung 25: In der Netzstruktur sind die Rasterpunkte als Knoten zwischen den Kanten zu erkennen. Trotzdem verwendet X3D Dreiecke zur Beschreibung der Geometrie. Es handelt sich also in Wirklichkeit um ein simuliertes Raster. Quelle: Eigene Abbildung

Im vorliegenden Fall sollten aber geologische Schichten visualisiert werden, deren Eigenschaft es war, „Löcher“ aufzuweisen, deren Grundfläche also nicht das gesamte Rechteck des Rasters ausfüllen. Im Knoten ElevationGrid muss jeder Punkt im Raster einen Wert aufweisen, undefinierte Punkte gibt es nicht. Diesem Problem wurde begegnet, indem die undefinierten Stellen des Rasters mit transparenten Texturen überdeckt wurden. Ist die Textur eines Objektes transparent, wird auch die Geometrie des Objektes „unsichtbar“. Allerdings muss beachtet werden, dass die „nicht definierten“ Punkte trotzdem berechnet und gerendert werden, nur eben durch Transparenzwerte von 100 Prozent nicht zu sehen sind. Sind die Diskrepanzen zwischen den dargestellten Werten und den nicht definierten Werten sehr hoch, so kommt es in der Geometrie zu Steilstufen. Wird nun eine Textur darüber gelegt, kommt es trotz genauer Positionierung zu eiszapfenartigen Artefakten (Abbildung 26). Selbst wenn der NODATA-Wert von -9999 auf einen Bereich in der Nähe der

echten Werte ersetzt wird,¹¹ sind die Stufen immer noch deutlich zu erkennen. Das liegt unter anderem daran, dass die Struktur der Overlays dem Raster entspricht, also an den Randbereichen Rechtwinklige Begrenzungen hat. Das Netz, das dem X3D-ElevationGrid zugrunde liegt, besteht dagegen aus Dreiecken (Abbildung 25). Aufgrund des Renderings der Hardware werden alle Geometrien, bevor sie in den 3D-Raum der Grafikkarte projiziert werden, in Dreiecke als kleinste Einheiten zerlegt. So entstehen Kanten zwischen den einzelnen Vertizes, die diagonal zum rechtwinkligen Aufbau des Rasters verlaufen. An diesen Stellen „rinnt“ die Textur dann an der Steilstufe hinab und es entstehen „Eiszapfen“ (siehe Abbildungen 26 und 27).

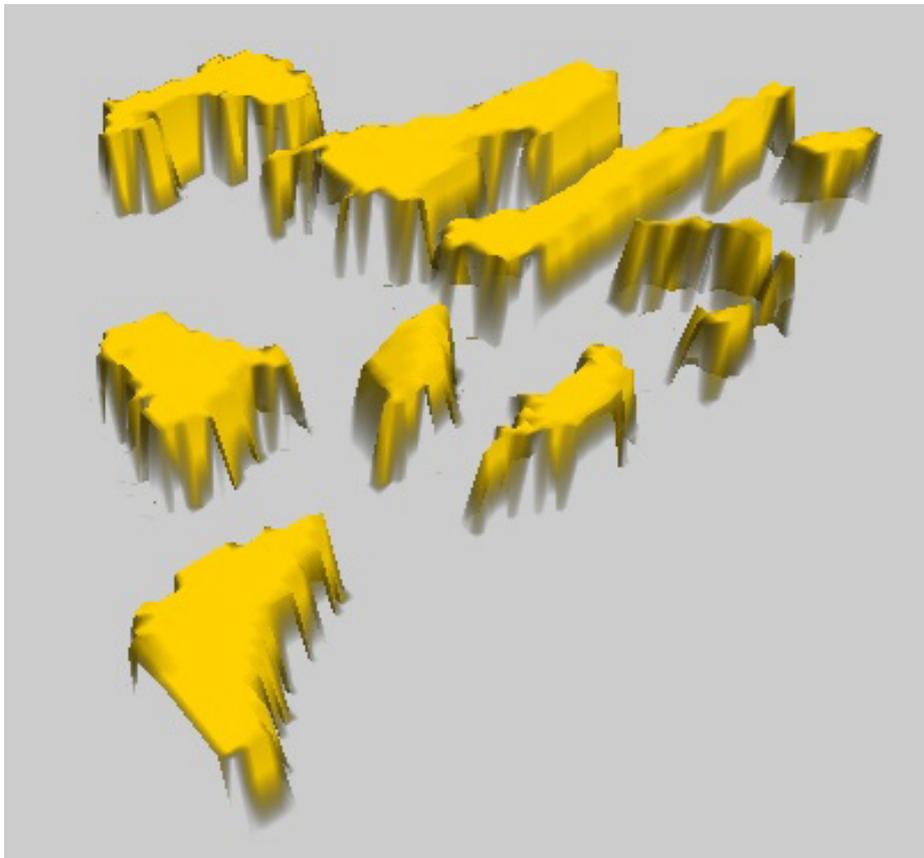


Abbildung 26: Deutlich sichtbar treten Artefakte ("Eiszapfen") an den steilen Übergängen von definierten Gebiet und nicht-definiertem. Quelle: Eigenen Abbildung

¹¹ Die Exportfunktion von ArcScene beispielsweise exportiert Rasterdaten als ElevationGrid im VRML-Format. Dabei werden die NODATA-Werte ersetzt durch den niedrigsten im Raster vorkommenden Höhenwert.

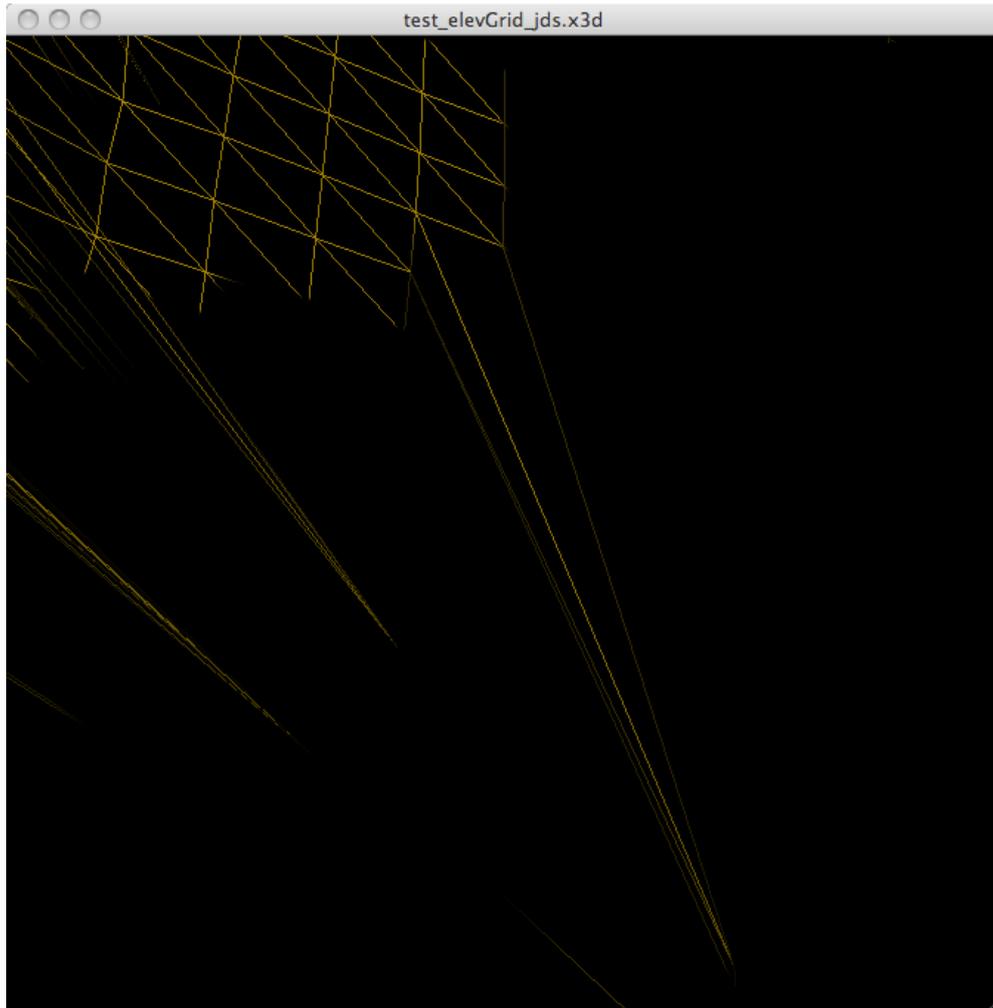


Abbildung 27: In der Gitternetz-Ansicht sind die Probleme an den Eckpunkten der definierten Oberflächen deutlich zu sehen. Quelle: Eigene Abbildung

Dieses Problem wurde umgangen, indem für den Aufbau der Schichten-Geometrie andere Grunddaten verwendet wurden, als letztendliche für die Visualisierung der Schichten. Für den Aufbau der 3D-Oberfläche wurden die Datenlayer herangezogen, die direkt nach der Interpolation der Bohrkerndaten entstanden. Im Modellierungsprozess der einzelnen Schichten werden nämlich erst nach der Interpolation die „Löcher“ aus den kontinuierlichen und regelhaften Rastern herausgeschnitten. Diese wurden wie oben beschrieben aus dem ASCII-Format in das X3D-Format überführt. Im nächsten Schritt wurden die nach Entfernen der nicht darzustellenden Bereiche entstandenen Raster als Grafik-Dateien im TIF-Format exportiert. Diese wurden benötigt um die Texturen zu erstellen, die die notwendigen

Transparenzen aufweisen, um nicht definierte Stellen „unsichtbar“ zu machen (Abbildung 28).



Abbildung 28: Ein Beispiel für eine Grafikdatei, die als Textur verwendet wurde. Die hier weiß erscheinenden Teile sind transparent. Quelle: Eigene Abbildung

Dazu wurden sie in den in Tabelle 3 aufgelisteten Farben eingefärbt, während die nicht definierten Stellen mit Transparenzen versehen wurden. Auf diese Weise konnten Artefakte vermieden werden.

Um die Texturen möglichst Rechnerfreundlich zu gestalten wurden sie zu Quadraten von 512x512 Pixeln verzerrt. In diesem optimierten Zustand wurden sie als PNG-24-Dateien gebrauchsfertig abgespeichert. Viele Grafikkarten haben Schwierigkeiten beim Aufrufen von Texturen, wenn diese nicht quadratisch sind und die Seitenlänge nicht einer Potenz von 2 entspricht. In den Spezifikationen von X3D wird die Seitenlänge von 512 Pixeln als Maximalwert deklariert, auch wenn viele Player mit etwas größeren Texturen umgehen können. Der X3D-Player bringt die Textur automatisch so auf, dass sie die gesamte Oberfläche der Geometrie bedeckt, sollten keine anderen Anweisungen gegeben werden. Die Texturen werden in diesem Fall also wieder auf ihre ursprünglichen Seitenverhältnisse „zurück verzerrt“, da sie ja exakt denen der Geometrie entsprachen. Leider wirken die Randbereiche am

Übergang zwischen Farbe und Transparenz aus bestimmten Perspektiven etwas unscharf und es können schmale weiße Ränder auftreten. Das hängt höchstwahrscheinlich mit den Shadern zusammen.

Im Fall der Schicht „Miozänoberkannte“ und des DGM lagen für die abgeschrägten Begrenzungsbereiche im Norden des Untersuchungsgebiets keine interpolierten Daten vor. Hier musste Abhilfe durch die Verwendung eines Medianfilters in einem Grafikprogramm wie Gimp oder Photoshop geschaffen werden. Die ASCII-Dateien wurden hierzu in ArcMap in Raster im TIF-Format umgewandelt. Die Exportfunktionen erzeugen ein Graustufenbild, in dem jede Graustufe für einen Höhenwert steht. Die NODATA-Werte erhalten den Graustufenwert 0 für Schwarz. Die Bilder können in einem herkömmlichen Grafikprogramm bearbeitet werden. Allerdings wird bei der Verwendung von Median und anderen Filtern der Bildmodus von *Graustufenbild* in *RGB* verändert, wobei es zu Verschiebungen in den Farbwerten kommen kann. Daher musste eine Kopie des Grauwertbildes erstellt werden, um daraufhin an dieser die Filteroperationen durchzuführen. Nachdem alle schwarzen Bereiche mit Grauwerten angefüllt waren, wurden im Originalbild durch eine Selektion nach Farbwerten alle schwarzen Bereich ausgewählt. Die Selektion wurde in einem eigenen Kanal gespeichert und auf die mit dem Filter bearbeitete Kopie angewendet. Dort erfasste sie nur diejenigen Bereiche, die mithilfe des Filters bearbeitet wurden. Die Selektion wurde dann durch Copy-and-paste in das Originalbild eingefügt. Auf diese Weise konnte umgangen werden, dass sich die Grauwerte des Originals bei der Bearbeitung durch den Filter verschieben. Allerdings waren die eingefügten Grauwerte aufgrund ebendieser Verschiebung jetzt etwas zu dunkel und mussten manuell durch den Helligkeitsregler angepasst werden. Trotz des großen Aufwandes konnten nicht alle Artefakte vollständig beseitigt werden, was aber nur in der Darstellung mit mehr als etwa siebenfacher Überhöhung zu erkennen ist (Abbildung 29).

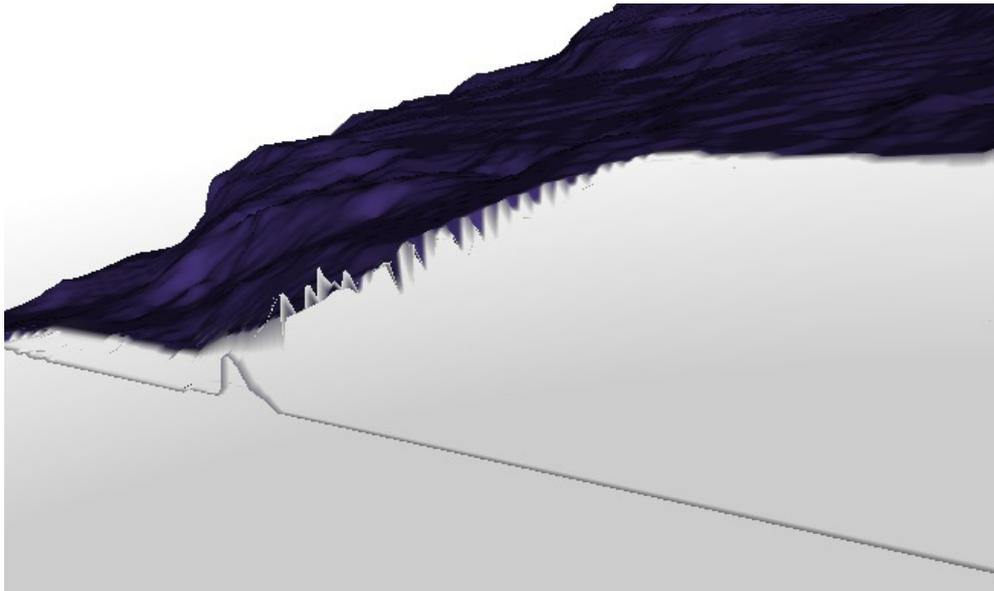


Abbildung 29: Bei extremer Überhöhung (hier: zehnfach) werden in den Schichten des Miozäns und des DGM noch vereinzelt Artefakte sichtbar. Quelle: Eigene Abbildung

Die Methode ermöglicht aber doch eine zufrieden stellende Visualisierung.

Ein Nachteil dieser Methode ist aber, dass, obwohl Bereiche erstellt werden können, die für den Betrachter nicht zu sehen sind, die unter der transparenten Textur liegende Geometrie trotzdem berechnet wird. So wird Rechenkapazität darauf verwendet, nicht sichtbares zu erstellen, was paradox klingt und nicht gut zu einem Konzept passt, dass möglichst schonend mit den vorhandenen Ressourcen umgeht. Dies ist bei Echtzeit-3D-Anwendungen, zumindest heute noch, wichtig, will man die Inhalte möglichst weit verbreiten (siehe Anmerkungen zu Echtzeit-Rendering in Kapitel 10 Fazit). Dieses Problem lässt sich theoretisch umgehen, indem anstatt im Rasterformat zu bleiben, die ASCII-Dateien in TINs umgewandelt werden. Dazu wurden sie in ArcMap in ein GIS-kompatibles Rasterformat (hier TIF) umgewandelt und von dort in ein TIN transformiert. TINs werden in der 3D-Export-Funktion von ArcScene als VRML-IndexedFaceSet exportiert. Diese bestehen aus einer Liste von allen verwendeten Datenpunkten im 3D-Raum, die dort mit ihren Koordinaten (x, y, z) aufgeführt sind. Eine weitere Liste aus Integerwerten benennt die Punkte aus der Liste, aus denen ein Polygon zusammengesetzt wird. Danach folgt der Wert -1, woraufhin das nächste Polygon beschrieben werden kann. Auf diese Weise werden

alle Polygone nacheinander beschrieben. Dadurch tauchen aber viele Punkte doppelt oder mehrmals in der Liste auf. Denn die meisten Punkte werden nicht nur in einem Polygon verwendet, sondern auch noch den benachbarten. Deswegen wird die Datenmenge schnell sehr groß. Wenn nun ein TIN ohne Verlust von Genauigkeit aus einem Raster erzeugt werden soll, werden sehr viele Dreiecke benötigt, um dies zu beschreiben. Wird dieses extrem große TIN dann als VRML-Datei exportiert, wird die Datenmenge zu groß und kann von einem üblichen Heim-PC nicht mehr bewältigt werden. Werden IndexedFaceSets anstatt ElevationGrids benutzt, müssen also in den meisten Fällen Abstriche in der Genauigkeit oder der Datendichte gemacht werden. Mit Rücksicht auf leistungsschwächere Systeme sollten jedenfalls die Ansprüche an die Rechenleistung und Arbeitsspeicher möglichst gering gehalten werden.

Bei großen Datenmengen, die in relativ hoher Auflösung in eine 3D-Szene eingebaut werden sollen, bietet sich deswegen die Verwendung von Levels of Detail (LOD) an. Die Idee hinter LOD ist die, dass bei einem Objekt, das aus der Nähe betrachtet wird, ein viel größerer Detailreichtum zu erkennen ist, als bei dem selben Objekt, betrachtet man es aus einer größeren Entfernung. Während beispielsweise eine mit Stuck verzierte Fassade die Gesichter einzelner Figuren erkennen lässt, wenn man direkt davor steht, sind aus 100 m Entfernung nur noch grobe die Körper beschreibende Strukturen zu erahnen, Details sind nicht mehr zu identifizieren, geschweige denn einzelne Gesichter. Dies wird am Computer simuliert, indem der Standpunkt der Kamera in der virtuellen Welt dazu verwendet wird, beispielsweise Texturen in geringerer Auflösung zu verwenden oder Strukturen zu vereinfachen je weiter die Kamera vom Objekt entfernt ist. LOD entsprechen damit in etwa der natürlichen Wahrnehmung des Menschen und werden, so sie richtig eingesetzt werden, vom Betrachter kaum bemerkt. Gleichzeitig schonen sie die Ressourcen eines Rechners. Wenn die Punktdichte einer Geometrie, in vorliegendem Fall der ElevationGrids, sehr hoch ist, sind die einzelnen Punkte des Rasters ab einer bestimmten Zoomstufe nicht mehr getrennt voneinander zu erkennen. Sie fallen zusammen, weil die weit entfernten Punkte perspektivisch zusammenrücken (im oberen Bereich der Abbildung 25 zu erkennen). So kommt es vor, dass der Abstand zweier oder mehrerer benachbarter Punkte auf dem Bildschirm geringer wird als ein einziger Pixel des Bildschirms. In diesem Fall würde ein Bildschirmpixel für mehrere Datenpunkte

verwendet werden. Ohne die Qualität des Bildes aber zu verschlechtern, könnte auf die Berechnung der überflüssigen Datenpunkte verzichtet werden, da ja schon ein Datenpunkt ausreicht, um einen Bildschirmpixel zum leuchten zu bringen. Die Darstellung so eng aneinander liegender Punkte überschreitet also nicht nur die Wahrnehmungsgrenze des Betrachters sondern auch die Darstellungsgrenze des Bildschirms. Trotzdem muss die 3D-Grafikkarte alle Operationen durchführen, um die Punkte in den virtuellen Raum zu projizieren, schon allein um festzustellen, welcher Punkt von welchem verdeckt wird. Das braucht Zeit und Kapazität, die eingespart werden könnte, da sie für den Betrachter ohne Nutzen ist. X3D stellt daher einen eigenen LOD-Knoten zur Verfügung. Innerhalb dieses Knotens können die Kamerapositionen angegeben werden, bei denen zum nächsten Detailgrad umgeschaltet werden soll, die Verwendung der einzelnen LOD können aber auch vom X3D-Player selbst bestimmt werden, wenn beispielsweise die Framerate unter eine gewisse Schwelle fällt (Kloss 2010).

Bei der Erstellung der geologischen Schichten für die Visualisierungsaufgabe dieser Arbeit wurden drei Auflösungsstufen verwendet. Die höchste Auflösung (LOD0) ist vorgegeben durch die Datengrundlage und betrug 15m. LOD1 und LOD2 entsprachen 30m und 60m. Für das DGM, das in einer Auflösung von 5m im LOD0 in die Visualisierung eingesetzt wurde, entsprachen die LOD1 einer Rasterweite von 10m, LOD2 wies 20m Zellgröße auf und LOD3 hatten die Datenpunkte einen Abstand von 40m. Um die Auflösungen zu verringern wurden die ASCII-Dateien in ArcMap in Raster umgewandelt und mit dem Resampling-Tool die Auflösungen eingestellt. Danach wurden sie wieder ins ASCII-Format umgewandelt. Die ASCII-Dateien mussten teilweise noch modifiziert werden, da je nach neuer Zellgröße die Verteilung der Zeilen oder Reihen des Rasters nicht genau aufging. Es werden dann automatisch Zeilen oder Reihen mit NODATA-Werten von ArcMap erstellt, die aus den ASCII-Dateien gelöscht wurden.

7.2 Erstellung des Orthofoto-Overlays

Das Orthofoto des Testgebiets sollte als so genanntes Overlay über das DGM gelegt werden und somit dessen Höheninformationen übernehmen. In X3D ist die Zuweisung von Grafikdateien in den Formaten PNG, JPG etc. zu Geometriedaten als

Textur vorgesehen. Dazu stehen verschiedene Knoten zur Verfügung. In vorliegendem Fall wurde auf den Knoten ImageTexture zurückgegriffen, der es erlaubt, die URL (Uniform Resource Locator) einer Grafikdatei anzugeben, die dann als Textur verwendet wird. Wie oben erwähnt (Abschnitt: Erstellung der Schichtenvisualisierung), sollten die Texturen ein quadratisches Format mit einer Kantenlänge die einer Potenz von 2 entspricht verfügen. Die Kantenlänge sollte außerdem nicht 512 Pixel übersteigen. Da das Orthofoto mit Maßen von 1650x5050 Pixeln diesen Anforderungen nicht entspricht, musste es in kleinere Einzelteile (Kacheln) zerlegt werden. Um die Anzahl der in einzelnen Dateien abgespeicherten Kacheln möglichst gering zu halten, wurde die maximale Kantenlänge gewählt, was 40 Kacheln von 512x512 Pixeln entsprach. Einige der Kacheln aus den Randbereichen wiesen schwarze Bereiche in denjenigen Teilen auf, in denen die Kachelgrenzen über den Bildrand hinweg wiesen. Die Kachelung wurde in ArcMap vorgenommen (Abbildung 30).

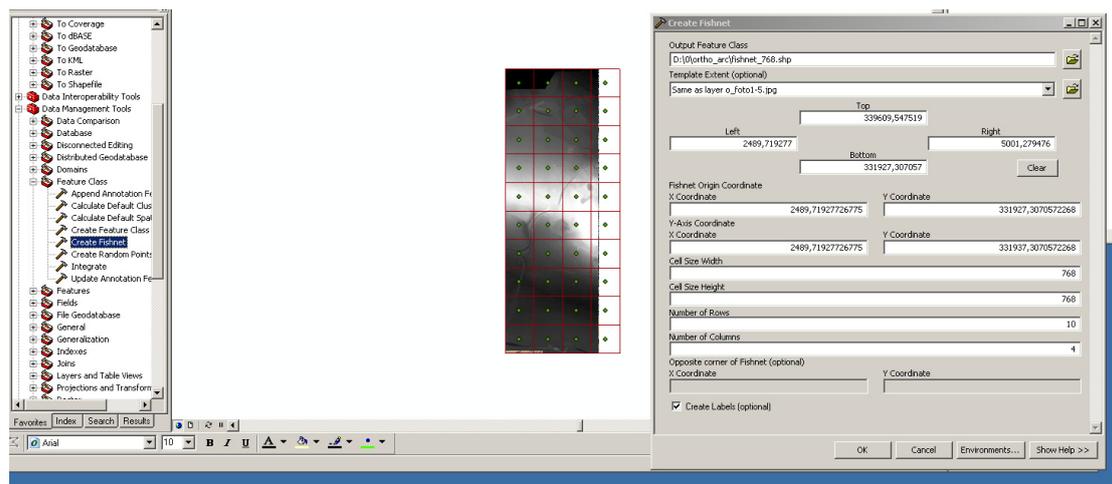


Abbildung 30: Die Ersellung der Kacheln in ArcMap. Quelle: Eigene Abbildung

Mit dem Fishnet-Tool wurde ein regelmäßiges Gitternetz von 768x768 m erstellt, das, umgewandelt in Polygone, als Schnittmaske diente. Unter Verwendung eines Python-Scripts wurde die Clipping Funktion für die vierzig Vorgänge automatisiert.

```

# Import system modules
import sys, string, os, arcgisscripting

# Create the Geoprocessor object
gp = arcgisscripting.create()
gp.overwriteoutput = 1

# Load required toolboxes...
gp.AddToolbox("C:/Program Files/ArcGIS/ArcToolbox/Toolboxes/Analysis Tools.tbx")
Input = "D:\\0\\ortho_arc\\fish_768_FeatureToPolygon.shp"
Input2 = "D:\\0\\ortho_arc\\dgm.tif"

# Local variables...
for i in range (0,40):

    j = str(i)
    Outputselect = ("D:\\0\\ortho_arc\\tiles\\temp\\XXX" + j + ".shp")
    Output2 = "D:\\0\\ortho_arc\\tiles\\dem\\tile_" + j + ".tif"
    gp.Select_analysis(Input, Outputselect, "\"FID\" = " + j)
    dsc = gp.describe(Outputselect)
    Coordinate = dsc.Extent
    gp.Clip_management(Input2, Coordinate, Output2, Outputselect, "", "NONE")

print i

```

Listing 3: Das Python-Script zur automatisierten Erstellung der Kacheln

Dieselben Polygone des Fishnet und das Script wurden auch für die Kachelung des DGM verwendet, um Geometriedaten mit identischen Ausmaßen zu erhalten. Im folgenden Schritt wurden die X3D-Dateien aus den ASCII-Dateien der einzelnen Kacheln des DGM erstellt, nach den gleichen Methoden wie weiter oben im Abschnitt „Erstellung der Schichtenvisualisierung“ beschrieben. In jeder dieser Dateien wurde der ImageTexture-Knoten eingefügt, in welchem die URL des entsprechenden Teils des Orthofotos vermerkt wurde. Diese vierzig Dateien wurden dann in einer einzigen Datei mit Hilfe des Inline-Knotens wie ein Puzzle zusammengesetzt. Die Funktion des Inline-Knotens erlaubt es, externe X3D-Dateien in eine Szene einzubauen. Die Inline-Knoten haben ein Feld „URL“, in das der Speicherort der einzufügenden X3D-Datei eingetragen wird. Die vierzig Inline-Knoten der einzelnen Kacheln wurden mit je eigenen Transform-Knoten versehen, die die Lage der Kacheln im gemeinsamen Raum der übergeordneten Szene enthielten, sodass jede Kachel exakt und nahtlos aneinander angrenzt. Diese übergeordnete Szene, die alle vierzig Inline-Knoten der einzelnen DGM-Kacheln mit den jeweiligen Texturen enthält, wurde wiederum durch einen Verweis in einem Inline-Knoten in die Datei mit den anderen geologischen Informationen und den Bedienungselementen eingefügt (siehe auch: Abbildung 33, rechts unten).

Da das Objekt in viel einzelne Kacheln aufgeteilt wurde, wäre es durchaus sinnvoll diese Sachlage dazu zu nutzen, um mit Hilfe von LOD und eventuell Streamingverfahren der Hardware weiter entgegen zu kommen und die Darstellung weiter zu optimieren. LOD könnten nämlich in diesem Fall nicht nur auf das gesamte Objekt angewendet werden, sondern diejenigen Kacheln des Objekts, die sich im Vordergrund befinden, könnten mit einem höheren Detailgrad dargestellt werden, als diejenigen aus dem Hintergrund. Dadurch würde ein einziges Objekt mit einer so großen räumlichen Ausdehnung optimal differenziert auf dem Bildschirm ausgegeben werden. Da die Erstellung einer Vielzahl von Kacheln aber sehr aufwändig ist, wären noch weiter automatisierte Abläufe wünschenswert. Yoo & Brutzman (2009) beschreiben das so genannte REZ-Tool, das hoch aufgelöste Daten automatisch kachelt und gleichzeitig eine Hierarchie verschiedener LOD kreiert. Es handelt sich dabei um eine Java-Applikation, die ein Verzeichnissystem mit X3D-Dateien erstellt, und in dem die erstellten Kacheln in verschiedenen LOD als verschachtelte Ordnerstruktur organisiert sind. Das Tool ist frei erhältlich mit genauen Beschreibungen zur Anwendung (byoo 2010, rez3d.com 2007). Aus Kompatibilitätsgründen und technischen Problemen konnten die durch das REZ-Tool erzeugten Dateien nicht für die vorliegende Arbeit verwendet werden, die Demo-Dateien aus dem Internet liefen aber problemlos. Die X3D-Earth-Knoten werden leider bisher noch nicht von demjenigen Viewer implementiert, der aus Performanzgründen für dieses Projekt ausgewählt wurde (siehe Kapitel 8 **Bewertung**). Auch die Exportfunktionen von ArcScene unterstützen eine automatische Kachelung mit Erstellung von LODs. Allerdings konnte die Auflösung des Orthofotos von 1,5 m nicht beibehalten werden, auch wenn in den Einstellungen zu Darstellung die Rasterweite explizit eingestellt wurde. Ein weiteres Problem ist, dass die Kacheln nicht gut zusammenzufügen sind und Löcher bei angrenzenden Kacheln entstehen. Das könnte daran liegen, dass in ArcScene der Rasterdatensatz vermutlich zuerst in Kacheln aufgeteilt wird, die dann erst, getrennt voneinander, in ihre unterschiedlichen Detailgrade heruntergerechnet werden. Auf diese Weise würden die einzelnen Kacheln unter lokalen Parametern in die neue Auflösung übertragen werden und so könnten Differenzen in den Grenzbereichen entstehen.

Abgesehen von diesem Problem und dem erwähnten Datenverlust funktionierte die aus ArcScene exportierte Datei aber sehr gut (Abbildung 31).

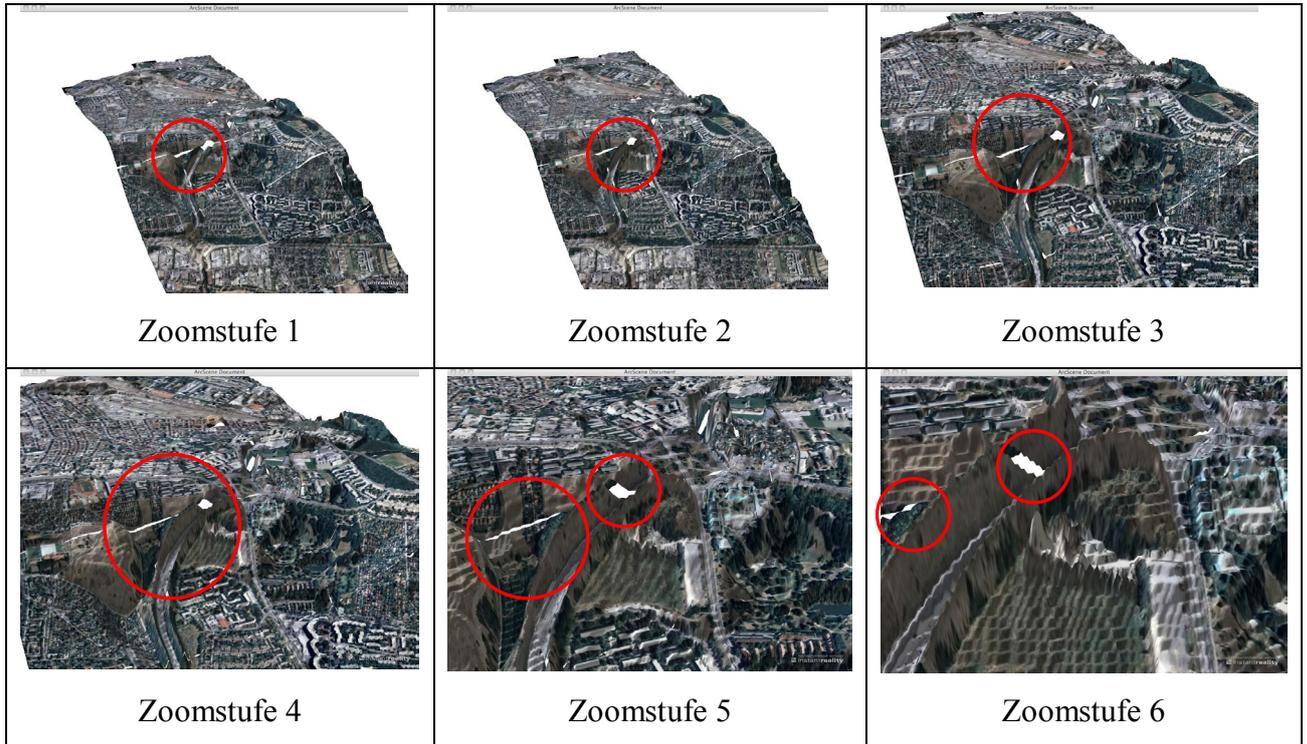


Abbildung 31: Die Lücken (rot markiert) zwischen den einzelnen Kacheln des von ArcScene ins VRML-Format exportierten Orthofoto-Overlays verändern sich je nach Zoomstufe/LOD. Quelle: Eigene Abbildung

7.3 Visualisierung der Shape-Dateien

Die Daten für die Visualisierung der Bohrkerne, Gruben, Höhenlinien und des Stadtplans lagen im Shape-Format vor. Die Polygone der Gruben und die Linienelemente der Höhenlinien konnten ohne weiteres direkt aus ArcScene als VRML-Dateien exportiert werden. Polygone werden als IndexedFaceSet ausgegeben, während Linien als IndexedLineSet in VRML umgesetzt werden. Die VRML-Dateien wurden, um Konformität zu schaffen, in der X3D-Entwicklungsumgebung *X3D Edit* in das X3D-Format umgewandelt.

Die Bohrkerndaten lagen als Punktdaten vor. Sie wurden mit der Extrusionsfunktion von ArcScene unter Verwendung der Angabe über die Teufe aus der Attributtabelle in Linienelemente transformiert, die die Bohrkerntiefe illustrieren. Auch diese konnten, wie oben für Gruben und Höhenlinien beschrieben, direkt aus ArcScene als VRML-Dateien exportiert und in X3D umgewandelt werden.

Auch die Daten für den Stadtplan von OSM lagen als Shape-Dateien vor. Deren Höheninformationen wurden aus den SRTM-Daten übernommen und anschließend ebenfalls mit ArcScene als VRML exportiert. Da die OpenStreetMap-Daten sich aus gut 32000 kleinen Einzelsegmenten von teilweise nur wenigen Metern Länge zusammensetzen, entstand eine relativ große Datei, die weder vom Online-Konverter des Fraunhofer Instituts¹² noch von X3D-Edit auf dem Desktop in das X3D-Format gebracht werden konnte. Im VRML-Format gab es keine Probleme bei der Darstellung als Inline-Element einer X3D-Szene. Die Versuche einer Generalisierung der Daten, um die Menge zu verringern, scheiterten an der Zusammensetzung der Shape-Dateien. Selbst wenn nur die kürzesten Elemente unter zwei Metern Länge ausgewählt und gelöscht werden (dies umfasste schon ca. 8600 einzelne Features), fehlen wichtige Informationen und das Straßennetz ist löchrig und zu unvollständig. Eine Generalisierung von Hand wäre wünschenswert, aber im Zeitrahmen dieser Masterarbeit nicht zu bewältigen.

Abbildung 32 zeigt den Ablauf der Arbeitsschritte in ihrer Gesamtheit.

¹² http://doc.instantreality.org/tools/x3d_encoding_converter/

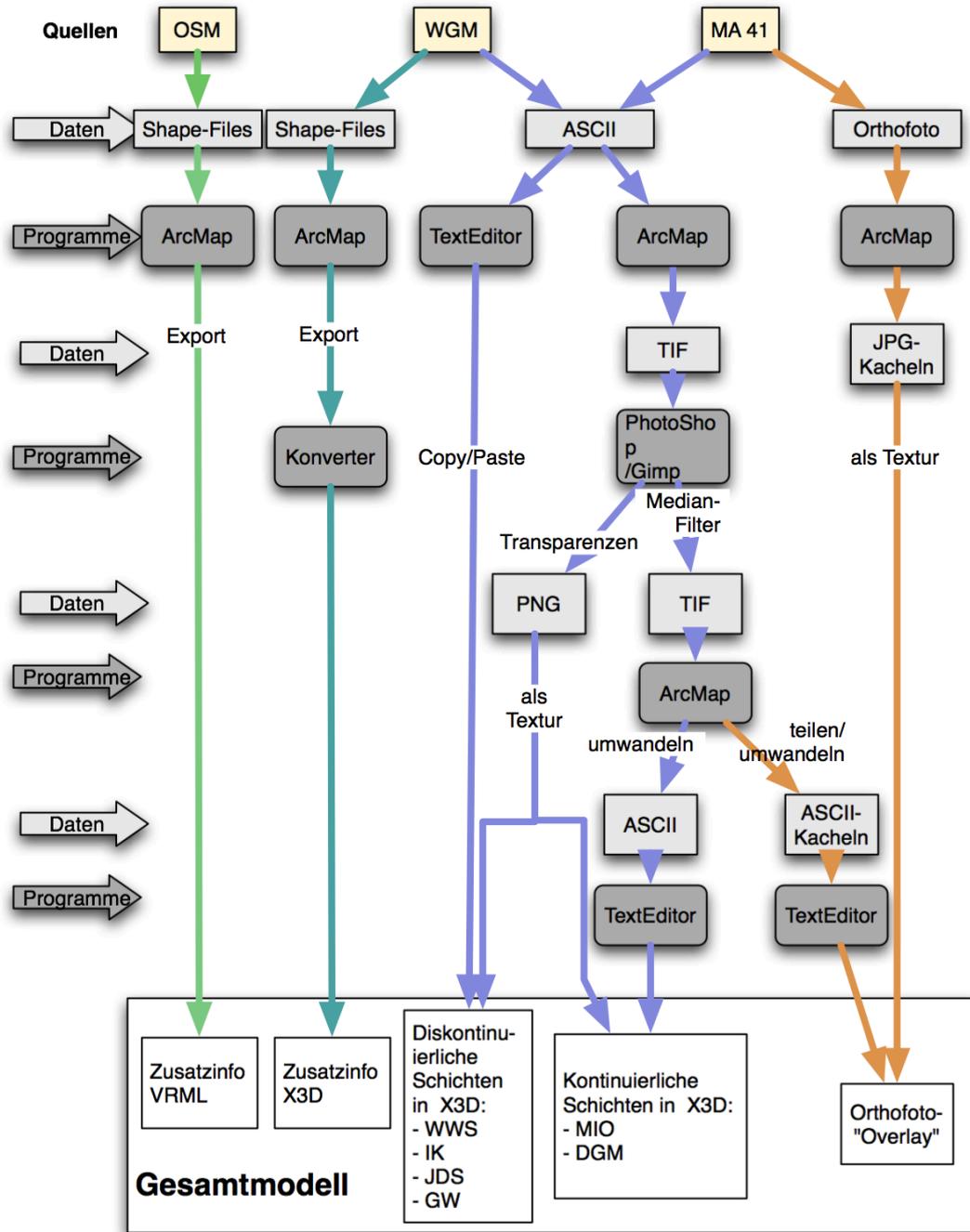


Abbildung 32: Der Workflow von der Quelle bis zum fertigen Bestandteil des Gesamtmodells. WWS: Wienerwaldschotter, IK: Interkolluvium, JDS: Junger Donauschotter, GW: Grundwasser, MIO: Miozänoberkante, DGM: Digitales Geländemodell. Quelle: Eigene Abbildung

7.4 Zusammenführung der Einzelteile und Ausstattung mit Funktionen

Nachdem die oben näher erklärten Methoden durchgeführt wurden, können die Daten nun in ein visuelles 3D-Modell integriert werden. Allerdings ist es nicht ratsam, alle Daten auf einmal in ihrem vollen Umfang ins Modell einzufügen, da die Informationsdichte zu hoch wäre, um noch angemessen vom Betrachter wahrgenommen zu werden. Das Erscheinungsbild wäre sehr überladen und unübersichtlich und würde wohl eher Verwirrung stiften als Klarheit schaffen. Außerdem würden einige Objekte von anderen Verdeckt oder überlagert, sodass gar nicht die gesamte Information zugänglich wäre. Des Weiteren würde die Datenmenge wahrscheinlich die meisten derzeitigen PCs überlasten, es würde zu einer ruckeligen Darstellung kommen oder gar zu Programmabstürzen. Um diese drei Punkte zu vermeiden, werden sowohl LOD genutzt, als auch dem Betrachter die Möglichkeit eingeräumt, Informationen oder ganze Informationsschichten (Layer) ein- und auszuschalten. Andererseits werden Detailansichten einzelner Objekte eingerichtet, um bei Bedarf deren volle Datenbandbreite einsehen zu können.

Generell wird beim Aufbau des vorliegenden Modells stark auf das Konzept der Prototypen von X3D zurückgegriffen. Durch Prototypen werden im Prinzip eigene Knoten gestaltet und somit die grundsätzliche Erweiterbarkeit (eXtensibility) des Standards genutzt. Prototypen sind Dateien, die außerhalb der X3D-Szene liegen. Sie können nicht direkt vom X3D-Player dargestellt werden. Es handelt sich vielmehr um eine Art Bibliothek, in der sich beispielsweise die Geometriedaten eines Objektes befinden. In der 3D-Szene, in der die Prototypen angezeigt werden sollen, weisen die Knoten auf die Bibliotheks-Datei hin, in der sich der Prototyp befindet. Der Vorteil von Prototypen, im Gegensatz beispielsweise zum Inline-Knoten, ist, dass man die Eigenschaften der Objekte aus der Szene heraus verändern kann. Die Prototypen werden in der Szene zuerst deklariert und dabei ihre manipulierbaren Eigenschaftsfelder angegeben. Daraufhin können sie beliebig oft in der Szene verwendet (das heißt instanziiert) werden. Eigentlich werden Prototypen für häufig verwendete Objekte verwendet, wie zum Beispiel Parkbänke in einem Park. Diese sehen alle ähnlich aus und kommen häufig vor. Da sie aber nicht alle die exakt gleichen Eigenschaften haben sollen, kann man in den Prototypen etwa die Textur

verändern, mit denen die Bänke ausgestattet sind oder auch die Länge, Höhe etc. Weiterhin können alle interaktiven Knoten wie Sensoren, Script-Knoten, etc. auf die instanziierten Prototyp-Knoten zugreifen und deren Eigenschaften dynamisch ändern. Im vorliegenden Fall wurden die Prototypen hauptsächlich verwendet, um die Deklaration der Funktionen in der Hauptdatei des interaktiven Modells von den reinen Daten zu trennen. Dadurch werden die Dateien übersichtlich gehalten, da nicht mehr die endlosen Zahlenkolonnen der Höhenwerte zwischen den Funktionen und weiterem Zubehör stehen, sondern lediglich die Deklaration der Prototypen und die jeweilige Instanziierung.

Neben den Prototypen werden in dieser Visualisierung auch eigenständige Szenen aus Dateien verwendet, zwischen denen der User während der Benutzung des Modells unmerklich wechselt. Die Dateien sind mit dem Anchor-Knoten miteinander verknüpft, dessen Funktionsweise den Hyperlinks von HTML-Webseiten ähnlich ist. Ein Objekt der Szene wird mit einem so genannten TouchSensor ausgestattet. Dieser reagiert auf die Position des Mauszeigers und wird aktiviert, sobald sich die Maus über dem Objekt befindet. Je nach Einstellung kann der TouchSensor beispielsweise einen Mausklick registrieren und dann die URL aufrufen, die im Anchor-Knoten angegeben wurde. So kann der Betrachter in X3D von einer Szene in eine andere „springen“.

Die dritte Art von Verbindung zwischen verschiedenen Dateien, die in diesem Modell genutzt wird, sind die schon mehrfach erwähnten Inline-Knoten. Mit diesen Knoten können komplette Szenen zu einer Hauptszene hinzugefügt werden. Durch die Angabe der URL im betreffenden Eigenschaftsfeld wird der Ort angegeben, an dem sich die Datei befindet, die hinzugefügt werden soll. Über das Feld *load* kann über die Veränderung des Booleschen Wertes *true* oder *false* die Datei geladen oder entladen werden.

Das Modell ist also in einer Vielzahl von Dateien organisiert (Abbildung 33 und 34). Die Dateien sind grundsätzlich alle miteinander verknüpft und man kann von jeder Datei auf alle anderen Dateien zugreifen. Deshalb hat der Nutzer auch nicht das Gefühl, zwischen identischen Szenen zu wechseln, sondern zwischen unterschiedlichen Ansichten. In diese können weitere Daten hinzugefügt oder wieder

aus ihnen entfernt werden. Der Einstieg in das Modell erfolgt über die Hauptdatei, die als erste geöffnet wird. Diese beinhaltet alle Schichten als Prototypen, allerdings nur in der geringsten Auflösung von 60 m, damit bei der Gesamtschau die PCs nicht überlastet werden. Jede Schicht kann in einer Detailansicht einzeln betrachtet werden. Dazu muss der User auf die interaktive Legende klicken und wechselt in eine andere Datei. Diese Kinddateien sehen auf den ersten Blick identisch mit der Hauptdatei aus, ihr Quellcode unterscheidet sich aber in einigen Aspekten, wie in den folgenden Abschnitten noch näher erläutert werden wird. Sie beinhalten außerdem nur Informationen über eine einzige Schicht, dafür dieses mal in allen Auflösungsstufen, organisiert in LOD, die in weiteren externen Prototyp-Dateien definiert sind. Aus jeder der Kinddateien kann direkt in die anderen Kinddateien gewechselt oder über den Reset-Knopf zurück zur Hauptdatei gelangt werden. Abbildung 33 veranschaulicht in einer graphischen Repräsentation die Struktur, in der die Prototypen mit einem oder mehreren LOD organisiert sind.

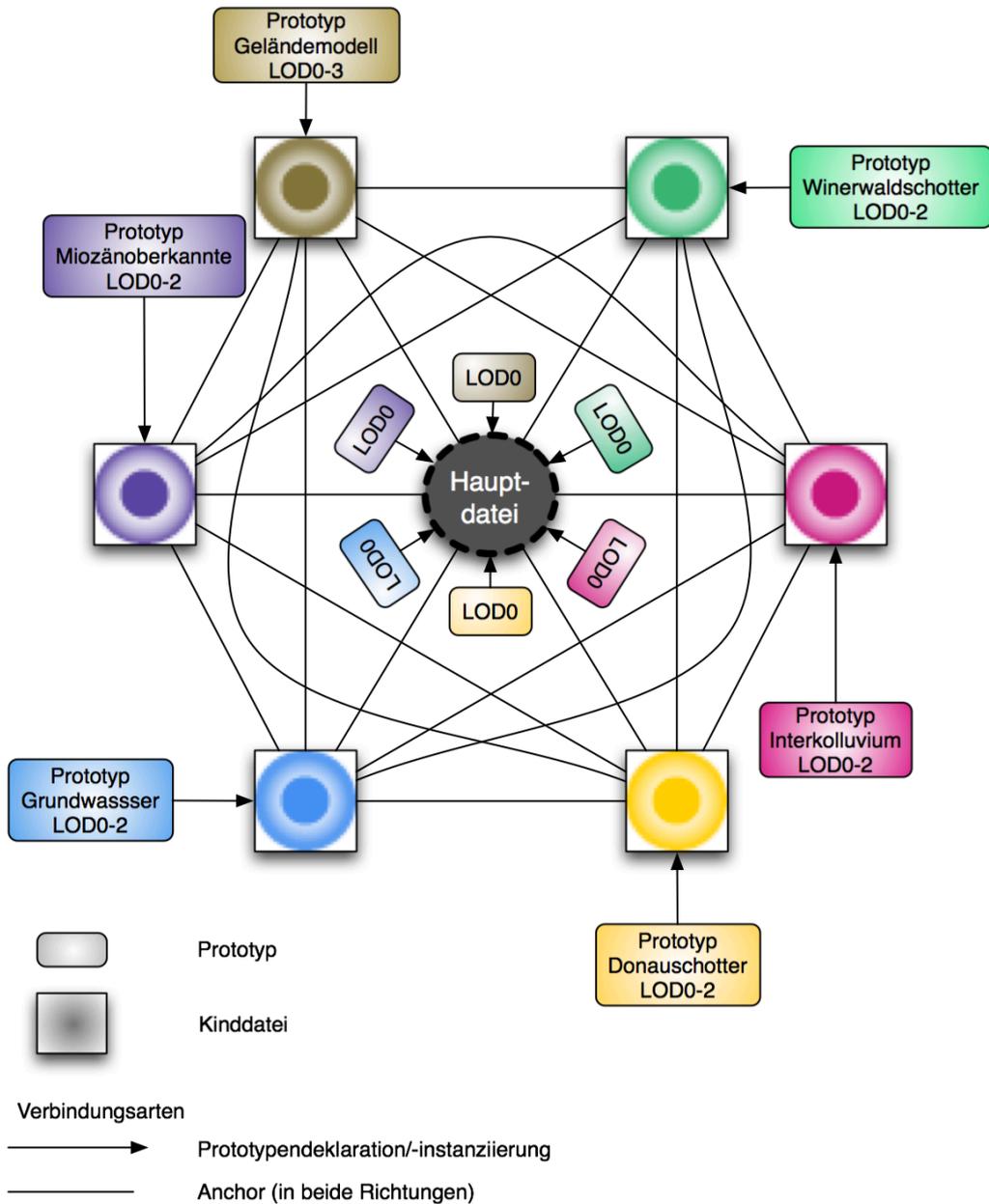


Abbildung 33: Die Dateistruktur zur Darstellung der Schichten. Die Hauptdatei verwendet Prototypen mit anderen LOD als die Kinddateien. Zwischen allen Dateien kann hin- und hergewechselt werden. Quelle: Eigene Abbildung

Sowohl die Hauptdatei als auch alle Kinddateien verfügen über Schaltelemente zum zu- oder abschalten von weiteren Dateien. Über die Schaltflächen für die Zusatzinformationen werden die Booleschen Werte im Feld *load* der betreffenden Inline-Knoten umgekehrt (Abbildung 34). So kann jede Szene mit den Informationen

über Bohrkerne, Gruben und dem Stadtplan angereichert werden. Eine Ausnahme stellt in dieser Hinsicht der Schalter für die Zusatzinformation *Orthofoto* dar. Dieser modifiziert nämlich keinen Inline-Knoten, sondern aktiviert einen Anchor-Knoten und leitet zu einer Datei weiter, deren Quellcode bis auf folgendes Detail identisch mit dem der Hauptdatei ist. Da das Orthofoto die Höheninformationen des DGM bereits enthält, ist es nicht mehr notwendig den Prototyp desselben in der Szene zu behalten. Deswegen enthält die Datei nicht mehr die Schicht des DGM in geringer Auflösung, sondern nur noch die Datei, in der die Kacheln des Orthofotos zusammengefasst sind, und zwar in einem Inline-Knoten. Dadurch kann auf einfache Weise vermieden werden, das DGM doppelt darzustellen. Da die Datei ansonsten mit der Hauptdatei identisch ist, sind selbstverständlich alle externen Dateien auch von hier aus durch die Schaltflächen zugänglich.

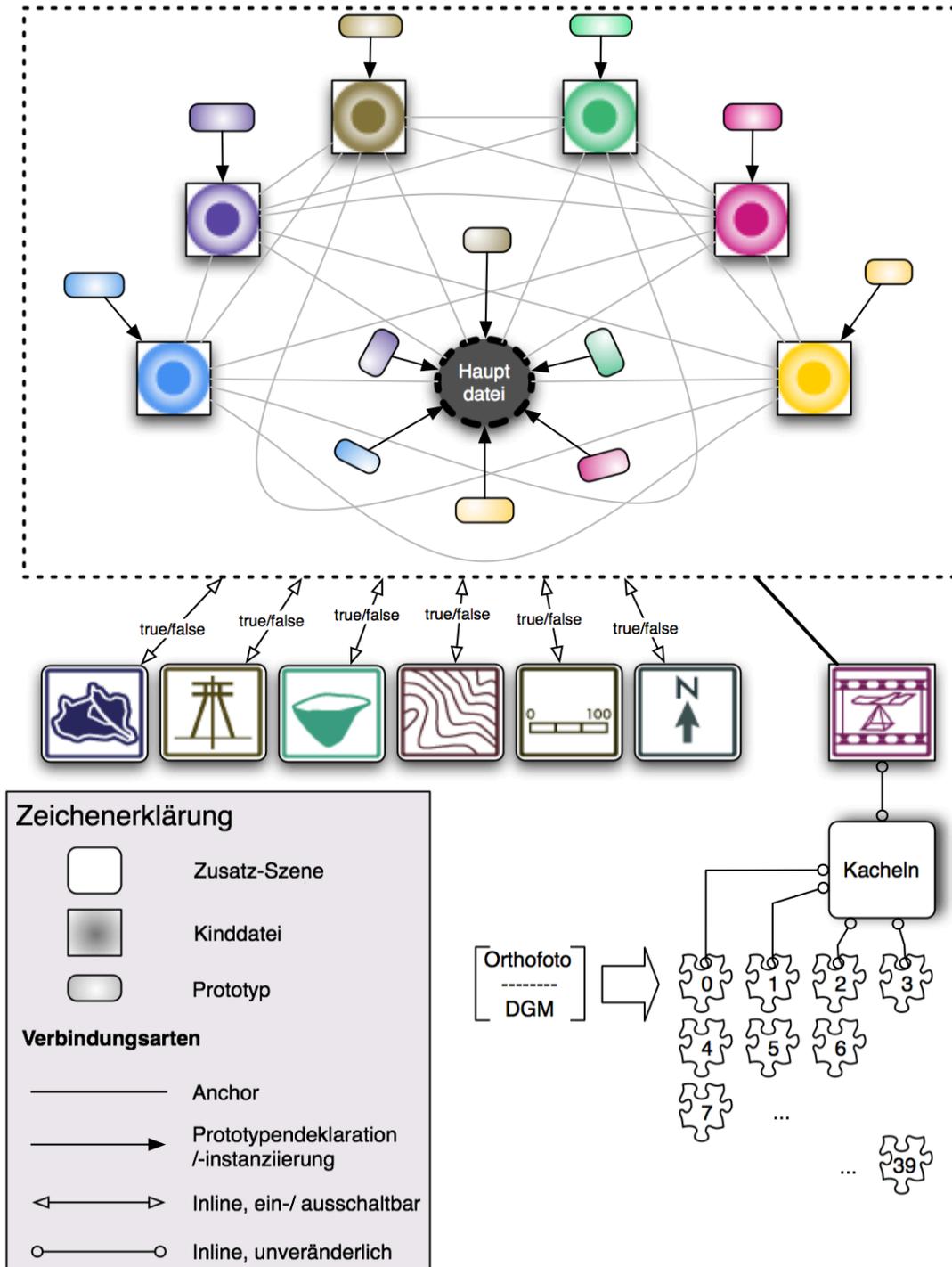


Abbildung 34: Die Struktur des gesamten Dateiensystems für die Darstellung aller Daten. Sowohl aus der Hauptdatei als auch aus allen Kinddateien können die Zusatzinformationen hinzugeschaltet werden, weshalb erstere im gestrichelten Rahmen zusammengefasst sind. Quelle: Eigene Abbildung

Die Haupt- und Kinddateien unterscheiden sich in ihrem Quellcode, wie oben beschrieben. Dies bezieht sich nicht nur auf die enthaltenen Daten, also die

unterschiedlichen Prototypen, die instanziiert werden.¹³ Es betrifft auch die Funktionen, die die interaktive Exploration des Modells zulassen.¹⁴

7.4.1 Interaktive Legende

Die Funktion der interaktiven Legende ist hinreichend im vorangehenden Abschnitt erklärt worden und betrifft das Wechseln zwischen Ansichten bzw. Dateien. Diese Funktion ist in allen Kinddateien ebenso enthalten wie in der Hauptdatei.

7.4.2 Überhöhung

Ebenso in beiden Arten von Dateien gleichermaßen enthalten ist die Funktion, die Überhöhung zu regeln. Als Überhöhung wird die übertriebene Darstellung der Höhenwerte bezeichnet, um auch subtile Geländeformen deutlicher erkennbar zu machen. Die Überhöhung wird erreicht, indem der vertikale Maßstab in Relation zum horizontalen Maßstab vergrößert wird. In X3D kann dies bewerkstelligt werden, indem man die Werte des Scale-Feldes aus dem Transform-Knoten beeinflusst. Im Scale-Feld sind die Skalierungswerte für jede der drei Raumachsen einzeln aufgeführt und können getrennt voneinander manipuliert werden. Die Default-Einstellung legt den Anfangswert in x-, y-, und z-Richtung auf 1. Für die Überhöhung müssen dementsprechend die y-Werte heraufgesetzt werden und sinnvollerweise auch wieder heruntergeregelt werden können. Dafür wurde im vorliegenden Modell ein stufenlos verstellbarer Schieberegler mit Hilfe eines PlaneSensor erstellt. Dieser Sensor zählt zur Gruppe der Drag-Sensoren. Ein mit dem Sensor ausgestattetes Objekt kann mit der Maus auf einer Ebene in x- und y-Richtung verschoben werden. Es können Grenzwerte für diese Verschiebungen definiert werden, die von dem Objekt nicht überschritten werden können. Hier wurden die Maximalwerte für die x-Verschiebung auf den Wert 0 (null) gesetzt, und so bewirkt, dass der Regler sich nur auf und ab bewegen kann. Die maximalen y-Werte wurden auf 0.09 begrenzt, sodass der Regler stufenlos von 0.00 bis zum Maximalwert bewegt werden kann. Der Wert wird bei jeder Veränderung der Position des Objekts vom PlaneSensor ausgegeben und an

¹³ Man könnte annehmen, dass es auch in der Deklaration zu unterschieden kommen sollte, denn diejenigen Prototypen, die nicht instanziiert werden, sollten eigentlich auch nicht deklariert werden müssen. Dies wird aber trotzdem getan, damit es beim Hin- und Herwechseln zwischen den Dateien nicht zu Fehlern kommt und die Prototypen vom X3D-Player immer wieder gefunden werden. (Dies betrifft vor allem InstantPlayer.)

¹⁴ Für Script-Auszüge: siehe Anhang

eine Funktion in einem Script-Knoten weitergeleitet. In dieser Funktion wird der y-Wert herausgefiltert und zur Berechnung der Überhöhung verwendet. Die Überhöhung wird einem Objekt mit dem Datentyp SFVec3f übergeben, um ihn aus dem Script-Knoten heraus an das Scale-Feld im Transform-Knoten zu übergeben, in dem sich die Schichten befinden. In dem Script-Knoten befinden sich noch weitere Anweisungen: die zweite Anweisung dient dazu, die Position der Schichten im Raum beizubehalten. Durch die Überhöhung „rutschen“ die Schichten nach oben, da ihre Werte ja rund 200 m über dem Nullpunkt des Koordinatensystems der Szene anfangen. Dies wird entsprechend ausgeglichen. Die dritte Anweisung dient der Anzeige im Display unterhalb des Schiebereglers, der dem Nutzer anzeigt, um welchen Überhöhungswert es sich handelt. Die Skala reicht von 1 (einfache bzw. keine Überhöhung) bis 10 (zehnfache Überhöhung).

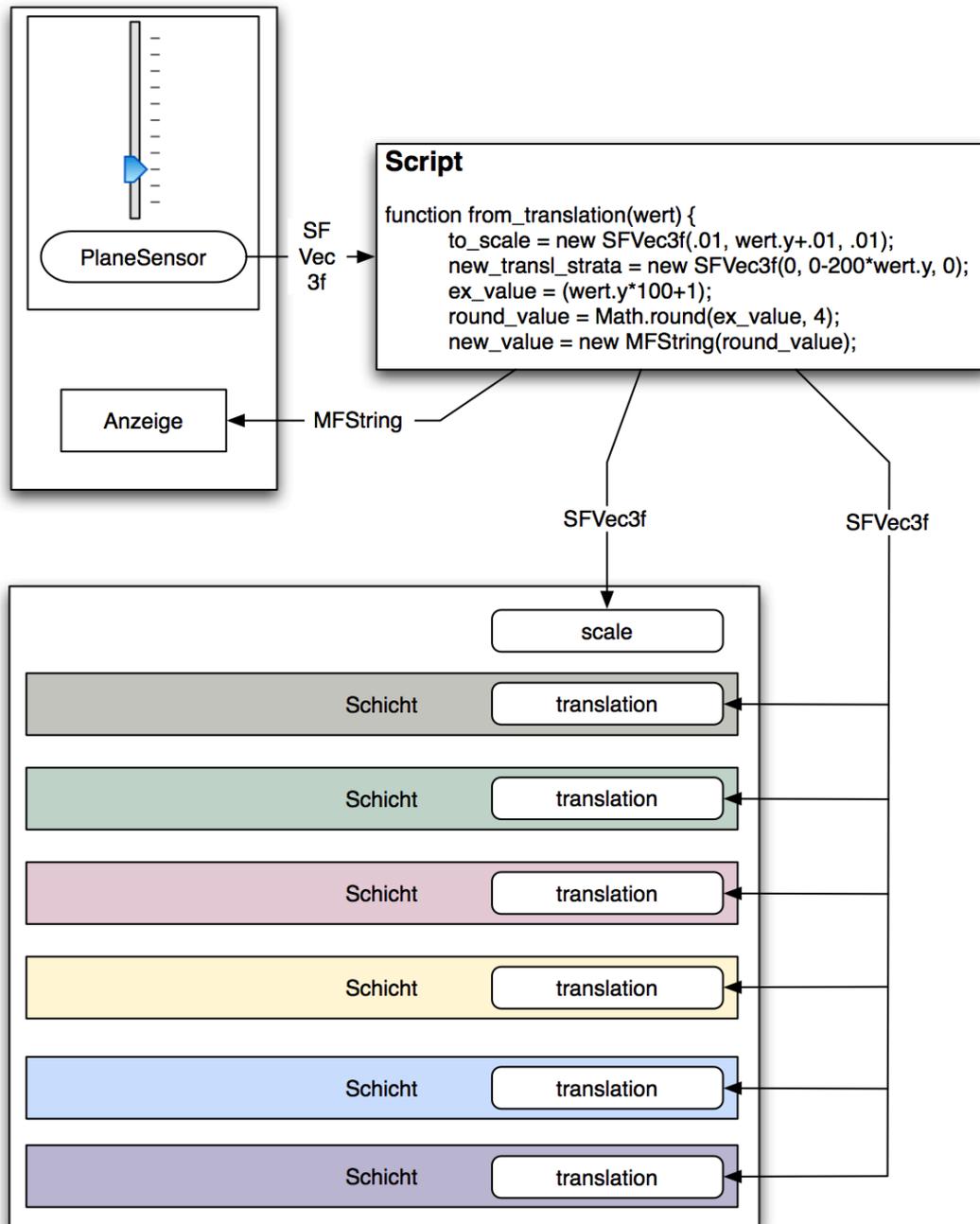


Abbildung 35: Schematische Darstellung der Funktion "Überhöhung". Quelle: Eigene Abbildung

7.4.3 Explodieren

Die Funktion, die Schichten räumlich weiter zu trennen, um sie besser analysieren zu können, ist nur in Szenen wichtig, in denen mehrere Schichten vorkommen. Sie existiert also nur in der Haupt- und der Orthofoto-Datei. Die Schaltfläche der Kontrollfunktionen, die diese Funktion auslöst, ist mit einem TouchSensor ausgestattet, der auf Mausklick ein Zeit-Event vom Datentyp **SFTime** aussendet. Das

Zeit-Event wird an einen Zeitsensor übermittelt, der es im `startTime`-Feld empfängt. Daraufhin gibt er über sein Feld `fraction_changed` einen Wert als `SFFloat` aus. Um den Datentyp `SFFloat` in einen vektoriellen Typen zu verwandeln, mit dem man Positionen im virtuellen Raum angeben kann (`SFVec3f`), wird der Zeitsensor mit einem Interpolator verknüpft. Im Interpolator-Knoten wird die Animation, hier eine Keyframe-Animation, festgelegt: Die Anfangsposition des Objekts, seine Endposition, und die in Anspruch zu nehmende Zeit, um das Objekt von der einen in die andere zu bringen. Der Interpolator wird mit einer Geometrie verbunden, an die er die neuen Positionswerte als `SFVec3f` an das Feld des Transform-Knotens übermittelt, das für die Positionsbestimmung zuständig ist, in diesem Fall das *translation*-Feld der betreffenden Schichtengeometrie. Über die so genannte Fan-out-Funktion kann ein einziger Zeitsensor die Ausgabewerte an mehrere Interpolatoren gleichzeitig versenden. Also wurde er mit den Interpolatoren aller Schichten verbunden, sodass mit einem Mausklick alle Schichten gleichzeitig animiert werden. Des Weiteren geht von dem Zeitsensor ein Befehl an einen weiteren Interpolator aus, der die Geometrie in der Schaltfläche betrifft. Diese wird nämlich umgedreht, sodass ein Icon zu sehen ist, das deutlich macht, dass die Schichten wieder in ihren Ursprungszustand zurückversetzt werden können. Diese Seite der Geometrie ist ebenfalls mit einem `TouchSensor` ausgestattet, der eine ähnliche Ereigniskaskade anstößt, nur dass dieses Mal die Interpolatoren für die andersherum laufende Animation angestoßen werden. Auch dieses Mal wird die Schaltfläche wieder umgedreht, und damit auch die ursprüngliche Form und Funktion der Schaltfläche wieder hergestellt.

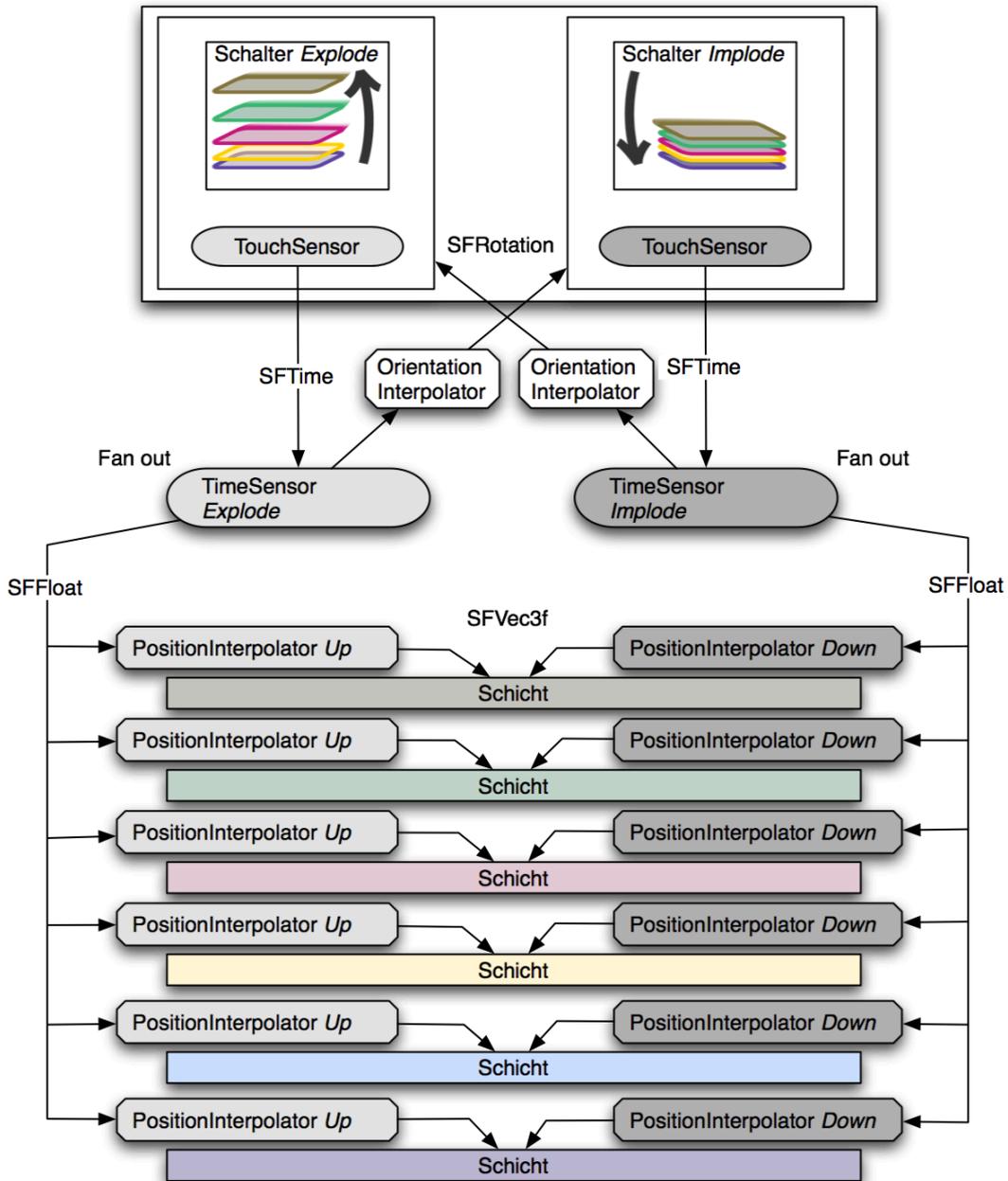


Abbildung 36: Schematische Darstellung der Funktion "Explodieren". Quelle: Eigene Abbildung

7.4.4 Profilschnitte

Eine weitere Funktion, die nur Sinn macht, wenn mehrere Schichten gleichzeitig dargestellt werden, ist die Erstellung von Profilschnitten. Daher ist auch diese nur in der Haupt- und der Orthofoto-Szene implementiert. Ziel war es, eine möglichst Nutzerdefinierte Schnittfläche durch die Schichten zu erstellen, so dass eine bessere Sicht auf die eher im Inneren liegenden Modelldaten geworfen werden können. Dafür wurden in den Kontrollfunktionen zwei Geometrien mit TouchSensor ausgestattet,

mit deren Hilfe die Position der Schnittfläche nach Norden oder Süden verschoben werden kann. Vom TouchSensor wird der Boolesche Wert *true* über sein Feld *isActive* ausgegeben und an einen BooleanFilter übermittelt, der es an einen IntegerSequencer weiterleitet. Dem IntegerSequencer kann eine Liste von Integerwerten zur Verfügung gestellt werden, die er nacheinander abrufen und ausgeben kann. Diese Werteliste kann entweder über das Feld *next* vorwärts durchlaufen werden oder über das Feld *previous* rückwärts. Wann immer jetzt also auf die entsprechende Geometrie geklickt wird, wird ein Wert der Liste in der angelegten Reihenfolge ausgegeben. Diese Integerwerte werden an einen Script-Knoten übergeben, der sie dazu verwendet, bestimmte Dateinamen zu erstellen, beispielsweise *dgm_ps0.png*, *dgm_ps1.png* oder *dgm_ps2.png* usw. Diese Dateinamen verweisen auf bereits vorhandene Texturen, die mit Transparenzen ausgestattet sind, sodass der Eindruck entsteht, sie seien an einer bestimmten Stelle abgeschnitten. Die Dateinamen werden als Datentyp *MFString* in das *ImageTexture*-Feld der Schichten-Prototypen eingetragen, die daraufhin die Textur auswechseln und einen Profilschnitt sichtbar machen.

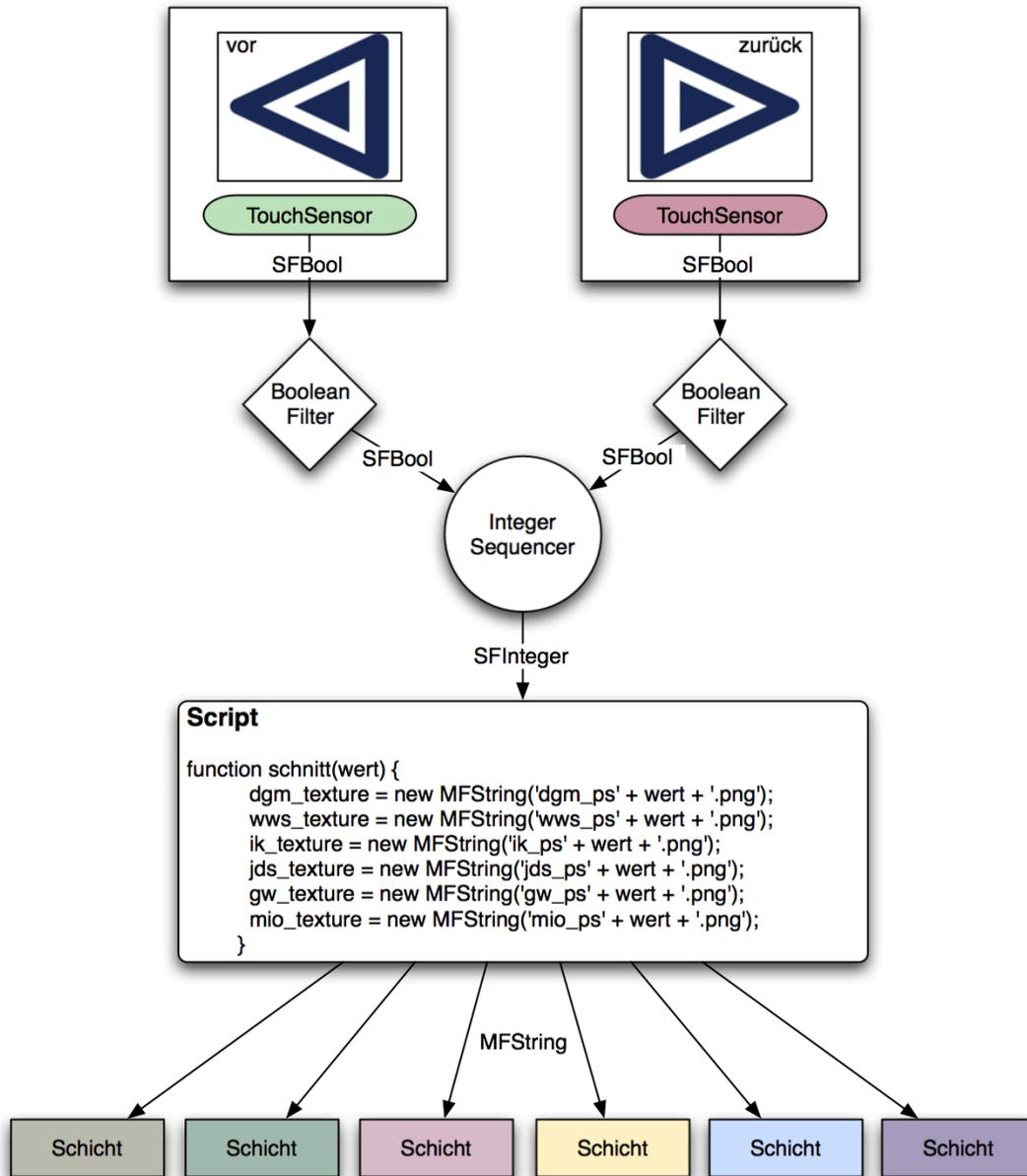


Abbildung 37: Schematische Darstellung der Funktion "Profilschnitte". Quelle: Eigene Abbildung

7.4.5 Ausgabe der Höhenwerte

Eine Funktion, die nur in den Einzelansichten der Schichten eingebaut ist, ist die Ausgabe der Höhenwerte der mit dem Mauszeiger berührten Punkte einer Geometrie. Die Geometrie, hier also das ElevationGrid der jeweiligen Schicht, wird mit einem TouchSensor ausgestattet. Der TouchSensor kann über sein Feld `hitPoint_changed` das Koordinaten-Tupel des berührten Punkts der Geometrie ausgeben. Dieses wird an einen Script-Knoten gesendet, der sich aus den drei Bestandteilen des `SFVec3f`-Wertes den `y`-Wert herausfiltert, da dieser dem Höhenwert über `Adria` in `m` entspricht.

Der Wert wird gerundet und in ein MFString-Objekt umgewandelt, sodass er an einen Text-Knoten weitergeleitet und dort in das Feld *string* eingetragen werden kann. Der Wert dieses Feldes erscheint dann im Display des User-Interface.

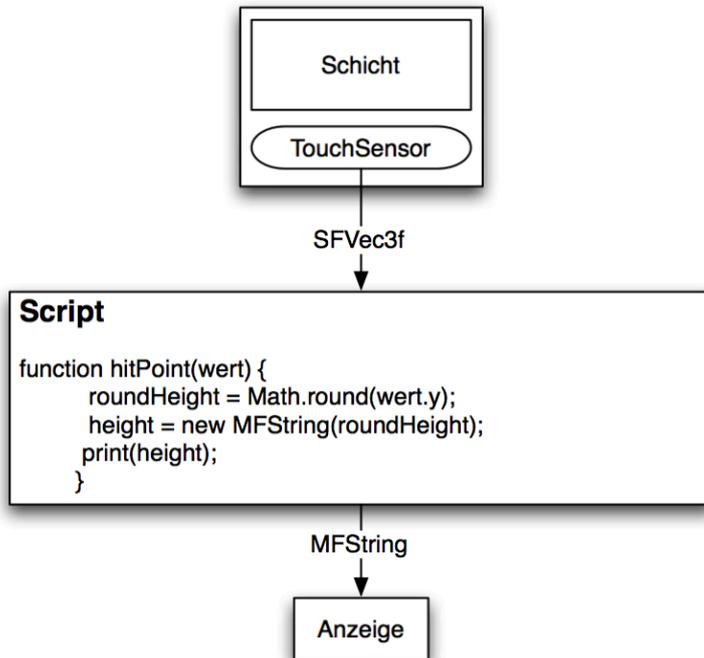


Abbildung 38: Schematische Darstellung der Funktion "Höhenwert". Quelle: Eigene Abbildung

7.5 Das Design der graphischen Benutzeroberfläche

Die graphische Benutzeroberfläche, oder das Graphical User Interface (GUI), wurde als Heads-Up-Display (HUD) entworfen. Dieser Begriff kommt aus dem militärischen Bereich und beschreibt eine Anzeige von Informationen und Steuerelementen, die dem Betrachter eines Displays direkt auf den Bildschirm eingeblendet werden, und er dementsprechend den Kopf nicht neigen muss, um Knöpfchen oder Schalter zu bedienen. Das HUD ist also von den Bewegungen der Objekte in der Szene unabhängig und bleibt für den Betrachter immer an der gleichen Stelle. Für eine X3D-Szene, die in eine HTML-Seite eingebettet ist, wird dieser Effekt ganz einfach dadurch erzielt, dass die Steuerelemente außerhalb des Szenenobjekts mit JavaScript definiert werden und sich so im Seitenbereich außerhalb des Plug-ins befinden. Die vorliegende Szene wurde aber so gestaltet, dass sie auch in Stand-alone-X3D-Playern

funktioniert, um das Verbreitungspotenzial zu erhöhen.¹⁵ Die Umsetzung des HUD beruht auf dem ProximitySensor, der Änderungen von Position und Ausrichtung der Kamera ausgibt. Diese Änderungen werden auf die entsprechenden Felder im Transform-Knoten, der das HUD umschließt, übermittelt, sodass es, wenn sich die Position oder der Winkel der Kamera ändert, auch die Position der Objekte innerhalb dieses Transform-Knoten gleichermaßen verändert. So entsteht der Eindruck, dass die Objekte immer an derselben Stelle bleiben würden, in Wahrheit aber passen sie sich der Bewegung der Kamera genau an und bleiben so nur *relativ* zur Kamera an Ort und Stelle.

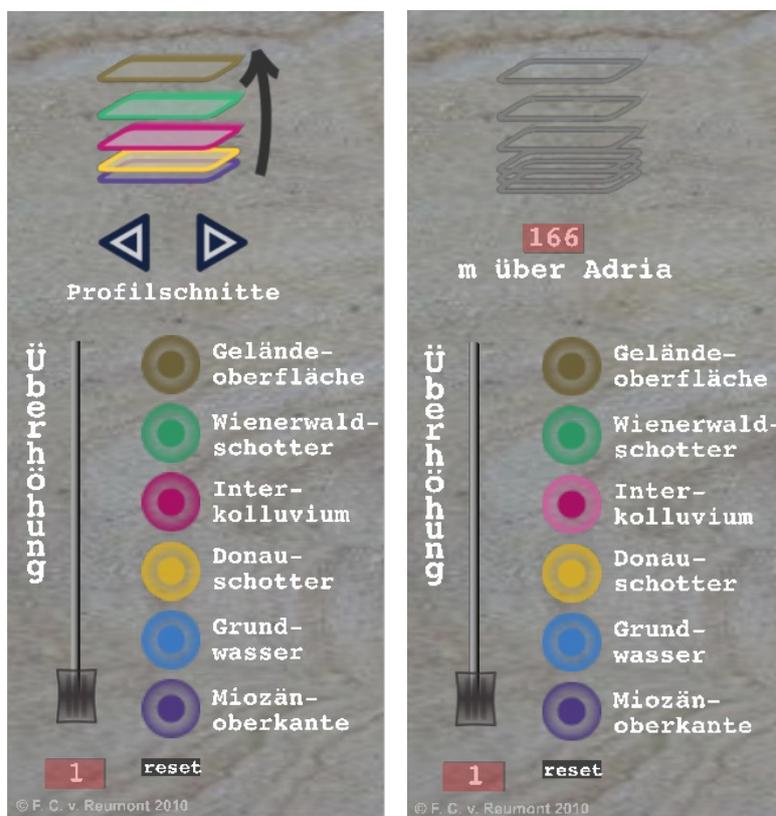


Abbildung 39: Das HUD der Steuerungsfunktionen. Rechts die Variante der Hauptdatei, links die der Kinddateien. Quelle: Eigene Abbildung

Das Design des HUD ist bewusst aus zweidimensionalen Elementen zusammengesetzt. Nicht einmal durch Schatten oder Glanzlichter wird eine

¹⁵ Für Mac-User gibt es derzeit kein vollwertiges X3D-Plug-in für den Browser, aber die Versionen des Octaga-Players, BSContact und InstantReality-Players sind als eigenständige Alternativen einsatzbereit. Der in Java programmierte Xj3D-Player funktioniert selbsterständig auch plattformübergreifend.

Erhabenheit oder Gravur vorgetäuscht. Dies unterstreicht die Andersartigkeit der Schaltfläche gegenüber des Objekts der Darstellung. Die Schaltfläche befindet sich ontologisch gesprochen auf der Metaebene über dem eigentlichen Subjekt der Darstellung, weil über sie auf das Objekt zugegriffen wird. Sie entspricht damit dem Kartenrahmen einer zweidimensionalen Karte, der ja auch Informationen *über* die Karte selbst enthält. In diesem Fall wurde allerdings bei Maßstabsleiste und Nordpfeil eine Ausnahme gemacht. Diese wurden direkt in das Kartenbild integriert, weil sich durch die Möglichkeit des Drehens und Zoomens Maßstab und Ausrichtung der 3D-Karte ständig ändern können. Es könnte durchaus in Erwägung gezogen werden, den Nordpfeil oder eine Kompassrose in das HUD zu integrieren, die sich sozusagen mitdrehen, und so zum Beispiel auch die Neigung anzeigen könnten. Aber bei der Maßstabsleiste ist es schwer möglich, ähnlich vorzugehen, weil er zu stark den perspektivischen Verzerrungen unterliegt.

Das HUD ist in zwei Teile aufgeteilt: am linken Rand befinden sich die Elemente, die die Darstellungsweise des Objektes beeinflussen, wie die Überhöhung und das Wechseln der Ansichten (Abbildung 39). Am oberen Bildrand unterhalb des Kartentitels sind die Schaltflächen für die Einblendung von weiteren kartographischen Informationen wie dem Orthofoto oder den Bohrungen angebracht (Abbildung 40).



Abbildung 40: Die Elemente zum Ein- und Ausschalten von Zusatzinformationen. Quelle: Eigene Abbildung

Zur besseren Trennung zwischen den beiden grundsätzlich verschiedenen Bedeutungen der Displaybereiche sind die Funktionsschaltflächen mit einem, dem Thema Geologie angemessenen, marmorierten Hintergrund unterlegt. Da es sich bei dem Dargestellten um die Geologie des Untergrunds einer vibrierenden, internationalen Großstadt handelt, wurden die Schaltelemente im Graffiti-Stil gestaltet, wie er von Künstlern wie Banksy und vielen Anderen geprägt wurde, sodass sie wie auf den marmornen Hintergrund aufgesprüht erscheinen. Um das Display nicht zu überladen, wurden für die Schaltflächen der Zusatzinformationen

selbsterklärende Icons entworfen, die ohne Beschriftung unter dem wechselnden Kartentitel angebracht wurden. Die Beschriftung der interaktiven Legende, der Funktionen und der Überschriften wurden in X3D-eigenen Schriftknoten erstellt, der leider nicht viel Spielraum bei der Gestaltung des Fonts übrig lässt. Der Titel wurde auf einer teiltransparenten, weiß grundierten Unterlage aufgebracht, um vor dem wechselnd mal hell mal dunklem Hintergrund zu bestehen.

7.6 Das Problem des vertikalen Maßstabs

Das breite Spektrum der Interaktionsmöglichkeiten lässt noch viel Spielraum für den weiteren Ausbau der Szene. Hier soll nur noch ein besonderes Problem bezüglich der kartographischen Grundausstattung des Modells erwähnt werden. Analog zum ein- und ausschaltbaren Maßstab für die horizontale Erstreckung des Modells, wäre ein vertikaler Maßstab ein weiteres, wünschenswertes Element der 3D-Karte. Es zeigt sich allerdings, dass dies nicht ganz einfach mit den anderen Funktionalitäten des Modells vereinbar ist, da die Höhendarstellung mit mehreren Funktionen dynamisch verändert werden kann. Die Funktionen, die die vertikale Erstreckung des Modells beeinflussen, sind namentlich die Regelung der Überhöhung und die Explosionsfunktion. Den vertikalen Maßstab auf die Überhöhung abzustimmen stellt dabei keine große Schwierigkeit dar. Die Geometrie des Maßstabs könnte einfach dem Transform-Knoten der Schichten untergeordnet werden und auf diese Weise von den Skalierungsangaben, die durch den Überhöhungs-Schieberegler generiert werden, genauso beeinflusst werden. Dadurch würde er einfach in y-Richtung des Koordinatenraumes in die Länge gezogen und entspräche in seinen absoluten Angaben immer den herrschenden Verhältnissen. Das Problem liegt eher bei der Explosionsfunktion. Bei dieser Funktion werden die Schichten um einen willkürlichen Betrag auseinander bewegt. Dieser Betrag stellt aber keineswegs ein Vielfaches des Abstandes der Datenpunkte der einzelnen Schichten dar, sodass es sich um ein proportionales Verhältnis handeln würde. Sonst könnte der Maßstab einfach entsprechend diese Wertes verzerrt werden, genau wie das bei der Überhöhung der Fall wäre. Der Betrag, um den sich die Schichten voneinander wegbewegen, wird aber einfach zu den vertikalen Koordinaten hinzuaddiert. Dies stellt eine in der Realität überhaupt nicht verankerte Situation dar, die sich auch gar nicht in den Höhenverhältnissen der einzelnen Schichten niederschlägt. Der vertikale Maßstab der

einzelnen Schicht bleibt also gleich, müsste aber sozusagen auseinander gerissen werden, um die Abstände der Schichten zueinander zu überbrücken. Das bedeutet, dass im explodierten Zustand ein einzelner vertikaler Maßstab nicht für die gesamte Erstreckung des Modells im vertikalen Raum gilt, sondern nur für die jeweiligen Abschnitte, in denen sich die Schichten befinden. Er gilt nicht in den (leeren) Räumen zwischen den Schichten, weil sie in der Natur nämlich gar nicht existieren. Damit nicht suggeriert wird, dass eine entsprechende Maßstabsleiste auch eine Aussage über diese Zwischenräume macht, wurde der vertikale Maßstab sicherheitshalber vorerst ausgelassen. Das Problem zu lösen stellt keine einfache Aufgabe dar und war im Rahmen der vorliegenden Arbeit nicht mehr zu bewältigen. Vorläufig ist der Nutzer dazu angehalten, den vertikalen Maßstab aus der Angabe des Überhöhungsdiplays abzuleiten, der ja, nach den üblichen Definitionen von „Überhöhung“, den Faktor darstellt, mit dem der horizontale Maßstab multipliziert wird.

8 Bewertung

Nachdem in Teil I dieser Arbeit die erste Forschungsfrage nach dem derzeitigen Stand der Technik eingehend geklärt wurde, soll an dieser Stelle die zweite Forschungsfrage nach der Eignung bestehender Technologie in Hinsicht auf die Umsetzung des Projektes der WGM erörtert und gegebenenfalls Erweiterungs- und Verbesserungsmöglichkeiten angeführt werden. Aufgrund von eingehenden Vorüberlegungen fiel die Entscheidung bezüglich der einzusetzenden Technologie auf X3D. Diese Überlegungen beinhalteten schon implizit einige Anforderungen an die Beschaffenheit der Mittel zur Umsetzung der Visualisierung des geologischen Modells, und viele positive Rahmenbedingungen wurden schon als Argumente für X3D herangezogen. Konnte das ausgewählte Werkzeug diesen Ansprüchen entsprechen? Außer an die Beschaffenheit des Werkzeugs wurden auch Anforderungen an die Fähigkeiten des Werkzeugs gestellt. Was kann mit der ausgewählten Technologie nun wirklich bewerkstelligt werden? Konnte die „Wunschliste“ der WGM aus Kapitel 4 (Tabelle 2) erfüllt werden? Darüber soll dieser Abschnitt Aufschluss geben.

Gesucht wurde nach einem möglichst preiswerten, möglichst allgemein gültigen und möglichst über das Internet verteilbaren Datenformat für die Visualisierung von geologischen Modellen. Für den X3D-Standard können alle wichtigen „Zutaten“ zur vollständigen Umsetzung einer 3D-Szene kostenfrei bezogen werden. Für die Erstellung einer 3D-Szene in X3D werden im Prinzip nur ein Text-Editor zum Beschreiben der Szene und ein gratis erhältlicher Player zur Betrachtung verwendet.

Bei komplexen 3D-Objekten kann es aber nötig werden, über ein Programm zu verfügen, dass die Erstellung von großen Gitternetzen mit vielen Knotenpunkten automatisiert oder für den Worldbuilder vereinfacht. Würden beispielsweise anstatt der ElevationGrids, die aufgrund ihrer einfachen Struktur bei guter Datenlage schnell manuell zu erstellen sind, IndexedFaceSets verwendet, wäre die direkte Umsetzung durch Copy-and-paste nicht mehr möglich. Dann wird Modellierungs-Software benötigt, die die Objekte ins X3D-Format exportieren kann. Aufgrund der

offensichtlichen Euphorie, mit der VRML in den 90ern empfangen wurde (siehe Hype-Zyklus in Abbildung 23), verfügen sehr viele professionelle Programme über eine entsprechende Export-Funktion. Die VRML-Dateien können anschließend sehr einfach über Konverter in X3D-Edit oder den Online-Konverter von InstantReality des Fraunhofer Instituts in das X3D-Format umgewandelt werden. Zu der am weitesten verbreiteten Software, die über VRML-Export verfügen, zählen unter anderem Studio Max 3D, Cinema 4D, und auch das Open Source-Programm Blender. Letzteres kann sowohl nach VRML als auch X3D exportieren, obwohl bei dazu offenbar leichte Modifikationen am generierten Code vorgenommen werden müssen (vgl. NPS.edu 2008). Der VRML-Export ist auch aus vielen GIS-Programmen möglich, wie ArcScene, GRASS-GIS, Gocad oder 3D Geomodeller. In dieser Arbeit wurde hauptsächlich auf die Exportfunktionen von ArcScene zurückgegriffen. Hierbei wurde deutlich, dass nicht immer die beste Form des Outputs von ArcScene verwendet wird. TINs wurden beispielsweise als IndexedFaceSets exportiert, wobei die Verwendung von für Dreiecksnetze effektiveren TriangleStripSets der Vermutung nach zu deutlich geringeren Dateigrößen geführt hätte. Sehr bedauerlich ist in diesem Zusammenhang auch, dass ArcScene weder GeoVRML noch X3D Earth in den Exportformaten unterstützt, was bei einem GIS durchaus wünschenswert wäre. Der Worldbuilder ist also in gewisser Weise abhängig von guten Exportfunktionen anderer Programme, um alle Möglichkeiten, die X3D zur Verfügung stellt, auszuschöpfen.

Eine funktionierende Infrastruktur zum Verfassen von X3D-Szenen steht im Allgemeinen also zur Verfügung. Aber wie sieht es mit der Widergabe aus? Derzeit ist eine Vielzahl an VRML-Playern als Plug-in-Versionen oder als eigenständige Viewer kostenlos im Internet erhältlich. Viele Softwarefirmen aus den unterschiedlichsten Bereichen haben ihre eigenen X3D-Player, -Browser oder -Viewer entwickelt. Es ist durchaus positiv zu bewerten, dass die Programme zur Widergabe der Szenen je nach Bedürfnissen dank der Offenheit des X3D-Standards selbst entwickelt werden können. Die so vermiedene Abhängigkeit von einer einzigen Firma wurde bereits an anderer Stelle als Vorteil beschrieben. Allerdings ist diese Freiheit mit Kosten verbunden, wie der Vergleich zwischen verschiedenen X3D-Playern aufdeckt. Zu den aktuellen Playern, die die neuesten X3D-Standards

unterstützen, zählen vor allem Xj3D¹⁶, BSContact¹⁷, InstantPlayer¹⁸ und Octaga¹⁹. Diese Player wurden für die Erstellung der 3D-Karte dieser Arbeit, so weit es möglich war, getestet, und es wurden überraschenderweise sehr große Unterschiede in der Wiedergabe der Szene festgestellt. Die Interoperabilität ist in der Theorie zwar gegeben, aber – zumindest bei einer komplexen Szenenarchitektur – offensichtlich in der Realität nicht ganz umgesetzt. Wirklich zufrieden stellende Ergebnisse wurden nur mit einem einzigen der vier Player erzielt, nämlich InstantReality. Der Octaga-Player weist in der Mac-Version in einigen Fällen Probleme auf, die über Inline-Knoten in die Hauptszene integrierten Dateien anzuzeigen, in anderen Fällen ist dies ohne Weiteres möglich. Da die Konsole keine Fehlermeldung ausgibt, ist es schwer, die Ursachen für dieses Problem festzustellen. Außerdem treten in der Mac-Version Probleme auf, wenn in andere Dateien über den Anchor-Knoten gesprungen wird. Unter Windows wird normalerweise per Dialogfenster gefragt, ob diese Datei gesichert oder geöffnet werden soll. Wird *Öffnen* gewählt, erscheint die Szene in einem neuen Fenster, was nicht unbedingt den Vorstellungen des Autors entspricht. Außerdem leidet die Qualität der Texturen, die als PNG-24 mit Transparenzen für die Schaltflächen verwendet wurden.

BSContact lief nicht stabil unter Mac OS-X. In der Windows-Version traten Probleme beim finden der Prototypen auf, die nicht in die Szene gerendert wurden, jedoch wurde keine Fehlermeldung in der Konsole ausgegeben. Es wurde nur ein einziger Prototyp angezeigt, und zwar der erste deklarierte und instanziierte, also der des DGM.

Xj3D konnte unter Mac auch nur einen Prototypen anzeigen, und zwar den kleinsten, also den des Interkolluviums. Obwohl in der Statusanzeige das Laden des Wienerwaldschotter als vollständig angezeigt wurde, erschien weder dieser noch ein anderer Layer im Display. Beim Versuch, die interaktive Legende zu bedienen, stürzte Xj3D ab.

¹⁶ <http://www.xj3d.org/>

¹⁷ <http://www.bitmanagement.com/>

¹⁸ <http://www.instantreality.org/>

¹⁹ <http://www.octaga.com/>

Die Ansprüche der X3D-Szene in ihrer Gesamtheit erfüllt als einziger Player InstantReality. Hier entsprechen sowohl die Darstellung aller Objekte, Schriften, Grafiken, Texturen als auch die Funktionalitäten innerhalb der Szene (Überhöhung, Explodieren, Profilschnitte) den angestrebten Zielen. Es treten keine Probleme beim Wechseln der Szene auf, die ohne Übergang wie gewünscht in das selbe Fenster geladen wird. Das Laden und Entladen der über den Inline-Knoten integrierten Szenen funktioniert einwandfrei. Auch die Prototypen werden richtig und vollständig angezeigt. Hierbei muss allerdings beachtet werden, dass immer alle Prototypen deklariert werden müssen, auch wenn diese nicht in allen Szenen gebraucht werden. Beispielsweise müssen alle Schichten in den Detailszenen deklariert werden, damit beim Wechseln zwischen den Ansichten die Prototypen nicht verloren gehen. Des Weiteren ist zu beachten, dass es in der Szene zu merkwürdig zerrissenen Polygonen an Stellen kam, an denen sich die Geometrien der geologischen Schichten fast berührten (Abbildung 41).

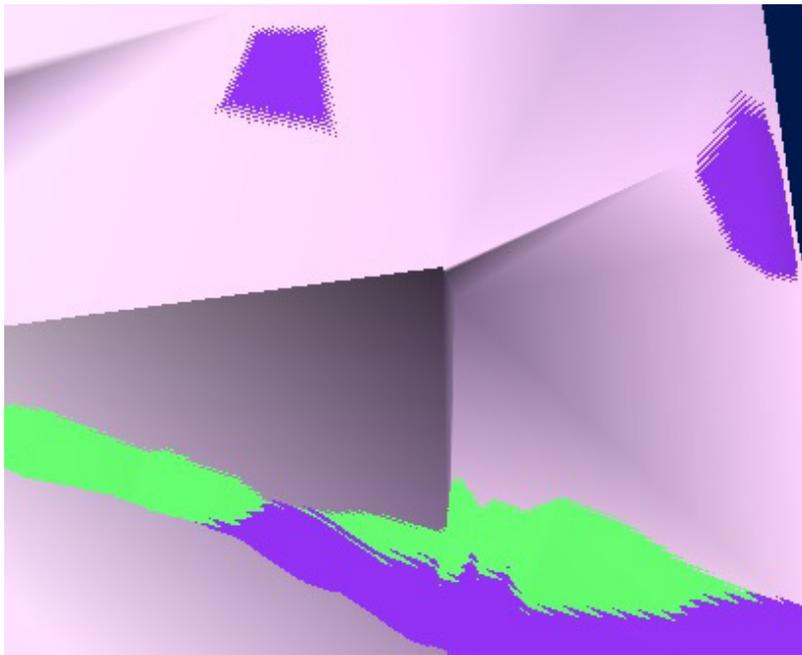


Abbildung 41: Das so genannte z-Fighting verzerrt die Polygongrenzen. Der Effekt fällt besonders bei Kamerabewegungen auf. Quelle: Eigene Abbildung

Dieses Phänomen kann auftreten, wenn der Renderer nicht genau unterscheiden kann, welche der Polygone näher an der Kamera liegen und welche von anderen verdeckt werden. Dieser Fall wird z-Fighting genannt, und muss im Viewpoint-Knoten

behandelt werden. Für dieses Problem des InstantPlayers sind speziell die beiden Felder `zNear` und `zFar` definiert worden (die nur von ihm interpretiert werden können), und mit entsprechenden Werten zu belegen. Auf diese Weise konnten eigentlich alle Probleme beseitigt werden. Nur in einem bestimmten Fall unter Windows bei der Verwendung eines Projektors traten Verschiebungen zwischen TouchSensoren und Position der Grafiken auf, wenn die Grafikkarte nicht auf 16bit Farbumfang gestellt wurde. Es war unklar, ob es an der Grafikkarte, dem Projektor oder dem Player lag.

Mit dem InstantPlayer wurden jedenfalls auch die besten Leistungen bezüglich der Framerates erzielt. Selbst die Bewegungen der größten verwendeten Dateien blieben ruckelfrei. Ein weiterer, nicht zu unterschätzender Vorteil dieses Players ist die Möglichkeit, den Rotationspunkt durch Doppelklick interaktiv aus dem Player heraus selbst zu bestimmen. Die Umsetzung der Navigationsmodi ist auch sehr gut gelungen, wobei der Einsatz der rechten Maustaste vielleicht noch mehr, beispielsweise für Zoomfunktionen, genutzt werden könnte. Die Navigation ist aber generell in X3D nicht ganz so komfortabel wie beispielsweise bei Google Earth oder Virtual Earth. Hier bestehen eventuell noch Verbesserungsmöglichkeiten.

Trotz der an dieser Stelle sowie im Abschnitt zur Umsetzung beschriebenen Schwierigkeiten konnten mit X3D 18 der 22 Punkte des Anforderungskataloges der WGM und die Überhöhungsfunktion, die in der Tabelle gar nicht als eigener Punkt aufgeführt wird, umgesetzt werden oder zumindest eindeutig als möglich identifiziert werden. Bei den weiteren vier Punkten ist es nicht auszuschließen, dass sie unter Einsatz von Java oder JavaScript auch umsetzbar sind (siehe Tabelle 5).

Tabelle 5: Umsetzungsstatus des Anforderungskataloges aus Tabelle 2

	notwendig	zusätzlich	umgesetzt: X möglich: o evtl. mögl.: ?	Kommentar
Darstellung der Daten und Handhabung allgemein				
Visuelle, räumliche Darstellung von 2,5d Raster-Daten in den vorliegenden Datenformaten	X		X	
Wählen der Beleuchtungsquelle		X	o	
Ein- und ausblenden von Datenlayern	X		X	
freies Rotieren im Raum	X		X	
freies Heran- und Herauszoomen	X		X	
manuelle Bestimmung des Drehpunktes	X		o	In InstantReality
interaktive Steuerung der Transparenz einzelner Layer	X		X	In früheren Versionen umgesetzt
3d-Steuerung per Maus	X		X	
Overlays	X		X	
Entfernungsabhängiges Ein- bzw. Ausblenden von Textinformationen		X	o	
rasterwertabhängige Einfärbung		X	?	
freie Wahl der Hintergrundfarbe		X	o	
Darstellung von Bohrungen als Säulen		X	X	
„Explodieren“		X	X	
Frei wählbare Kontrasteinstellung		X	?	
Abfragemöglichkeiten durch Mausclick auf Position				
Rasterdaten: Rasterwert (z.B. Höhenwert) eines gewählten Layers	X		X	
Rasterdaten: tabellarische Auflistung aller Werte der sichtbaren Layer vom Klickpunkt senkrecht nach unten		X	?	
Vektordaten: Abfrage eines vordefinierten Attributes eines Layers	X		o	Siehe Abschnitt... GA
Vektordaten: Abfrage von Informationen über POI	X		o	
Erweiterte (Analyse-) Funktionen				
Eingabe eines Punktes mittels x-, y-Koordinaten	X		?	
Profilschnitterstellung		X	X	Mit Workaround
Verlinkung mit Dokumenten POIs (Text, Foto, Video,...)		X	o	

9 Ausblick

Es gibt viele aktuelle Entwicklungen auf dem Gebiet von Internet 3D-Grafiken, wie im Zusammenhang mit WebGL und O3D schon kurz erwähnt wurde. Auch das Web3D Konsortium ist (mittlerweile wieder) sehr aktiv und scheint dem neuen HTML5 entgegen zu feiern, der für die 3D-Technologie große Chancen birgt.

Ein Ziel für die nähere Zukunft (Ende 2010) ist die Integration von X3D und HTML5. Es soll möglich werden, X3D Szenen in jedem HTML5-fähigen Browser zu betrachten, ohne Zusatzprogramme oder Plug-ins herunterladen zu müssen und zwar auf allen (auch mobilen) Geräten, Betriebssystemen oder Displays. Dafür wird in dem Projekt X3DOM (sprich: x-freedom) an einem JavaScript-Framework gearbeitet, das die X3D-Codeangaben interpretiert und direkt aus dem Browser heraus über das WebGL genannte Interface auf die Grafikhardware des Computers zugreift (X3DOM 2010). WebGL wird zurzeit von der Khronos Group entwickelt und vier der wichtigsten Browserhersteller (Apple, Google, Mozilla und Opera) treiben die Entwicklung voran. Auch die Entwicklungen von X3DOM laufen zurzeit: erst im Oktober dieses Jahres (2010) wurde die Version 1.1 veröffentlicht.

Die Runtime und Framework X3DOM soll also den Einsatz von X3D in normalen Webseiten ohne Plug-in oder Plug-in-Interface ermöglichen, sodass eine deklarative Definition für 3D-Szenen ohne low-Level JavaScript (WebGL) oder der OpenGL Shader Language (GLSL) möglich wird (The WebGL Cookbook 2010). Die Szenenbeschreibung in X3D wird dabei einfach in den HTML-Code zwischen den `<body>`-Tags eingebunden. Dank der WebGL-Technologie kann im neuen HTML5-Standard direkt mit der Hardware-3D-Beschleunigung kommuniziert werden und so in Echtzeit 3D-Rendering einfach aus dem Browser heraus betrieben werden. Das heißt, dass Browser, die WebGL unterstützen, (X)3D-Szenen ohne Plug-in anzeigen können. Da die Szenen unmittelbar in das Document Object Model (DOM) der HTML-Seite eingegliedert sind, können die einzelnen Objekte (also die Knoten) auch mit JavaScript manipuliert werden, und zum Beispiel mit „onclick“-Events belegt werden. Das dazu benötigte WebGL wird derzeit als offener Standard entwickelt und wird demnächst als 1.0-Version veröffentlicht werden. Die Anzeige von 3D Graphiken im Browser ohne Plug-in wird von vielen als eine der neuen HTML5-

Technologien angesehen (The WebGL Cookbook 2010). Drei der heute am weitesten verbreiteten Browser unterstützen es bereits in ihren Test-Versionen: Google Chrome, Mozilla Firefox und WebKit (Safari). Es ist noch unklar, ob der Standard von der zukünftigen Version des Internet Explorer, dem Browser mit den größten Marktanteilen, unterstützt werden wird. Aktuelle Statistiken zeigen, dass Internet Explorer einen Marktanteil von etwa 51,6 Prozent hat. Firefox, Chrome und Safari kommen zusammen auf 45,8 Prozent Marktanteil. Allerdings zeigen die Trends bei der Nutzung von Internet Explorer seit Juli 2009 zumindest stetig nach unten während gerade bei Chrome der Trend kontinuierlich aufwärts geht (Abbildung 42).

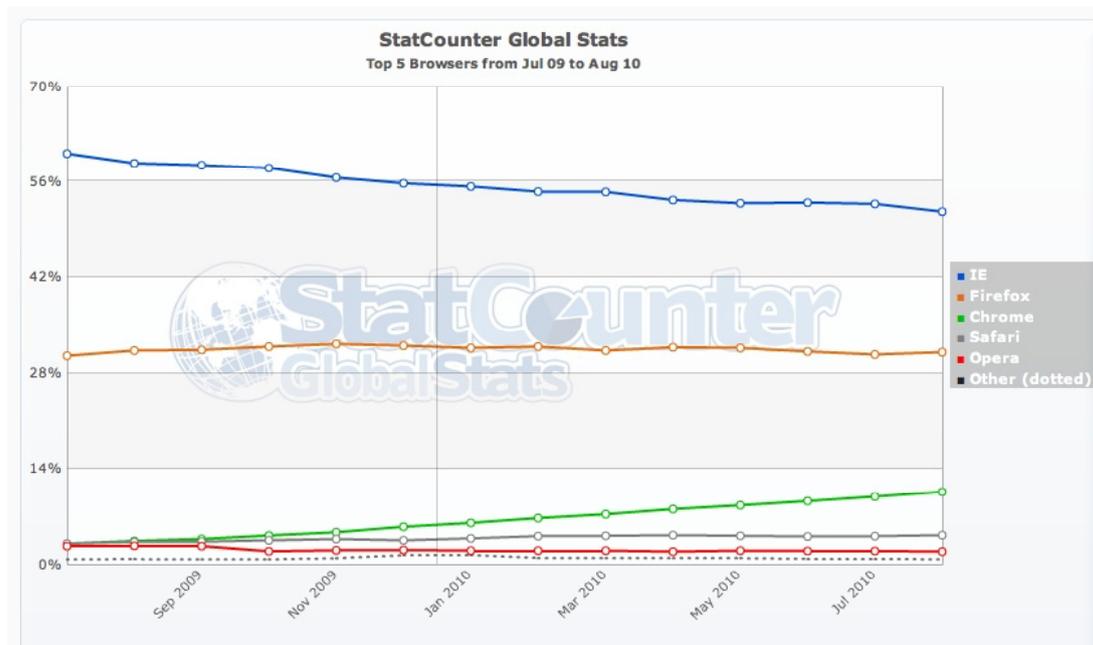


Abbildung 42: Statistik über die Verbreitung von Internetbrowser nach Marke.
Quelle: <http://gs.statcounter.com/>

Die Zahlen zeigen, dass es immer noch extrem wichtig ist, darauf zu achten, dass Webseiten sowohl im Internet Explorer funktionieren, als auch in kleinen Browsern, wie Opera (Opera ist auch ein Mitglied der WebGL-Working Group) mit Marktanteilen von um die drei Prozent, da selbst diese noch eine absolute Anzahl von schätzungsweise 50 Millionen Nutzern aufweisen (Pingdom 2010). Wenn die 3D-Fähigkeit der nächsten Ausgaben der Browser intensiv genutzt wird, kann es für IE von Nachteil sein, dies nicht zu ermöglichen. Die Hoffnung ist, dass diese Technologie bald von allen wichtigen Browsern unterstützt wird.

In Googles ANGLE-Projekt arbeiten die Entwickler an einer Lösung, WebGL-Applikationen auch auf Rechnern laufen zu lassen, die nicht über OpenGL verfügen, wie es bei vielen Windows-Maschinen der Fall ist. Googles Chromium-Browser kann ohne installierte OpenGL-Treiber 3D-Inhalte anzeigen. Ansonsten hofft man bei Google, das auch Microsoft bald WebGL implementiert.

Auf die Frage, warum WebGL erst jetzt eingeführt wird, lautet die Antwort in der Public mailing list von Khronos (2010), dass die Browser erst seit Kurzem über ausreichend Leistungsstärke verfügen. Die Leistungsfähigkeit von Browsern und Hardware sind natürlich auch für X3D von größter Bedeutung. In Anbetracht der jetzigen Entwicklungen, kann mit einem Boom von 3D-Anwendungen gerechnet werden, wenn sie überall ohne Plug-in im festen und mobilen Internet betrachtet werden können. Vielleicht hat X3D aufgrund der jahrelangen Erfahrung und des Status als ISO-Standard besonders gute Chance, den Markt anzuführen.

Im Jahr 2008 hatten erstmals mehr Menschen eine mobile Breitbandinternetverbindung als eine über das Festnetz. Laut International Telecommunication Union (ITU 2009) zeigt dies das extrem hohe Potenzial des mobilen Internets. WebGL ist auch auf mobilen Geräten anwendbar, sodass deklarative 3D-Szenensprachen auch den zukünftigen Anforderungen immer weiter entwickelter mobiler Geräte und Displays entsprechen können. Auch X3D hat den Anspruch, auf diesem Markt präsent zu sein.

Weitere Entwicklungen von X3D betreffen den Ausbau und die Verbesserung der X3D Earth Komponente. Unter anderem ist ein GPS-Knoten in Entwicklung, der dem X3D-Player die Informationen über den aktuellen Standpunkt des Beobachters aus dem GPS-Sensor mitteilen kann. Dies bietet natürlich viele Möglichkeiten für Anwendungen auf mobilen Geräten, auch für Projekte wie das dieser Arbeit. Im Feld könnte beispielsweise Mitarbeitern der Stadt Wien die Situation des Untergrunds an ihrem Standpunkt angezeigt werden.

10 Fazit

Aufgrund der oben angeführten Überlegungen scheint dem Autor dieser Arbeit die Verwendung von X3D für die Umsetzung des Projekts der WGM als beste Wahl aus der Palette der aktuell zur Verfügung stehenden technischen Möglichkeiten. Trotz einiger Herausforderungen konnte der Anforderungskatalog zufrieden stellend umgesetzt werden und aufgrund der vielfachen Erweiterungs- und Programmiermöglichkeiten stehen noch viele Interaktions- und Verbesserungsmöglichkeiten offen. Es sollte gründlich überlegt werden, den Schwierigkeiten bei der Erstellung der geologischen Schichten aus Rasterdaten entgegenzukommen, indem man schon bei der Modellierung nicht doch auf TINs anstelle von Rasterdaten zurückgreift. Dies würde das Erscheinungsbild der Objekte noch weiter verbessern und die automatische Erstellung von diskontinuierlichen Objekten vereinfachen. Oder man modelliert Volumenkörper, die als eigene Objekte in den 3D-Raum gestellt werden und nicht als Teile eines Objektes mit Lücken. Auch sollten noch weitere Programme zur Erzeugung von 3D-Netzstrukturen in Betracht gezogen werden, wie zum Beispiel Meshlab²⁰ oder der Okino Converter²¹. Da es sich dabei um Software handelt, die darauf spezialisiert ist, 3D-Netzstrukturen zu erstellen, könnten diese Programme einige Vorteile in der Erstellung von komplexen Objekten bereithalten.

Dennoch muss man sich eines bestimmten „Gesetzes“ bewusst sein: In Bezug auf Echtzeit 3D-Rendering sind Qualität und Datenmenge in den Grenzen der Hardwareleistung gefangen. Es gilt: je größer die Datenmenge, desto geringer die Qualität (hier: Auflösung) der Daten, da die Rechenleistung bei Prozessen wie Z-Buffering oder Animationen extrem beansprucht wird. In diesem Fall sind für die Visualisierung von Geo(logischen) Daten die Entwicklungen auf den Gebieten des Streamings und der Kompression sehr interessant, sowie die automatisierte Kachelung von Objekten. Aufgrund der besonders in diesem Jahr stark vorangetriebenen technologischen Entwicklungen im Bereich der 3D-Computergrafik kann damit gerechnet werden, dass die Zukunft noch Großes bereithält.

²⁰ <http://meshlab.sourceforge.net/>

²¹ http://www.okino.com/conv/exp_vrml.htm

Literatur

- BAOJUN W., BIN, S. & S. ZHEN (2009): A simple approach to 3D geological modelling and visualization. In: Bulletin of Engineering Geology and the Environment. Bd. 68. S. 559-565.
- BEARD, D.J. (2006): Using VRML to Share Large Volumes of Complex 3D Geoscientific Information via the Web. In: Web3D 2006 11th International Conference on 3D Web Technology, Columbia Maryland, 18-21 April 2006, ACM. S. 163-167. Columbia Maryland.
- BEARD, D.J., HAY, R., NICOLL, M. & D. EDGE (2005): 3D Webmapping – 3D Geoscience Information Online. In: Proceedings of SSC 2005 Spatial Intelligence, Innovation and Praxis: the national biennial Conference of the Spatial Sciences Institute. Melbourne.
- BEHR, F.-J. & H. LI (2008): SUAS Mapserver – an Open Source Framework for Extended Web Map Services. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing.
- BODE, T., BREUNIG, M. & A. CREMERS (1994): First Experiences with GEOSTORE, an Information System for Geologically Defined Geometries. In: IGIS'94: Geographic Information Systems. Nievergelt, J., Roos, T., Schek, H.-J. & P. Widmayer (Hrsg.) Nr. 84. S. 35-44. München Heidelberg.
- BODUM, L. & M. JAEGLY (2006): The Democratizing Potential of Geographic Exploration Systems (GES) Through the Example of GRIFINOR. In: Innovations in 3D Geo Information Systems. Abdul-Rahman, Zlatanova&Coors (Hrsg.), Berlin, Heidelberg. S. 235-244.
- BOGUSLAWSKI, P. & C. GOLD (2009): Construction Operators for Modelling 3D Objects and Dual Navigation Structures. In: 3D Geo-Information Sciences. Lee, J. & S. Zlatanova (Hrsg.). Berlin, Heidelberg. S. 47-60.
- BROCKHAUS (2006): Stichwort Modell, Band 18, 21. Auflage. Mannheim. S. 625f.
- COFFEY, J., BEARD, D.J. & D.A. RYAN (2007): Visualising Coastal Seabed Characteristics: Using VRML Models to Present Three Dimensional Spatial Data via the Web. In: Journal of Spatial Science. Bd. 52. Heft: 1. S. 133-143.
- EGENHOFER M. (1991): Reasoning about binary topological relations. In: Lecture Notes in Computer Science, Band 525/1991 (Advances in Spatial Databases, 2nd Symposium SSD '91 Zurich, Switzerland). Berlin, Heidelberg. S. 141-160.
- GOCKE, D., GOCKE, T. & A. LEHMANN (2010): GeoRovertm Extension: 3D Exploration Mapper. In: Beiträge zur COGeo 2010. Marschallinger, R., Wanker, W.

& F. Zobl (Hrsg.) http://www.cogeo.at/publications/cogeo2010/cogeo2010_0007.pdf. Salzburg.

GOLD, C. (2008): Modelling 3D Geo-Information. In: Advances in 3D Geoinformation Systems.. von Oosterom, P., Zlatanova, S. Penninga, F. & E. Fendel (Hrsg.). Berlin, Heidelberg. S. 429-433.

GOLD, C. & P. BOGUSLAWSKI (2010): Euler Operators and Navigation of Multi-shell Building Models. In: Developments in 3D Geo-Information Sciences. Neutens&De Maeyer (Hrsg.), Berlin Heidelberg. S. 1-16.

GREINER G. (2000): Representation and processing of surface data. In: Principles of 3D Image Analysis and Synthesis. Girod, B., Greiner, G. & H. Niermann (Hrsg.). Boston, Dordrecht, London. S. 142-152.

HAGEDORN, B., HILDEBRANDT, D. & J. DÖLLNER (2010): Towards Advanced and Interactive Web Perspective View Services. In: Developments in 3D Geo-Information Sciences. Neutens & De Maeyer (Hrsg.), Berlin Heidelberg. S. 33-51.

KEIM, D., PANSE, C. & M. SIPS (2005): Information Visualization: Scope Techniques, and Opportunities for Geovisualization. In: Exploring Geovisualization. Dykes, J., MacEachren, A. & J.M. Kraak (Hrsg.). Oxford. S. 23-52.

KESSLER H., MATHERS, S. & H.-G. SOBISCH (2008): The capture and dissemination of 3D geospatial knowledge at the British Geological Survey using GSI3D software and methodology. In: Computers & Geosciences. Heft 35 (2009). Ottawa. S. 1311-1321.

KLOSS, J. (2010): X3D. Programmierung interaktiver 3D-Anwendungen für das Internet. München.

KOLAR, J. 2006: On the Road to 3D Geographic Systems: Important Aspects of Global Model-Mapping Technology. In: Innovations in 3D Geo Information Systems. Abdul-Rahman, Zlatanova&Coors (Hrsg.), Berlin Heidelberg. S. 207-223.

KOLBE, T. (2004): Interoperable 3D-Visualisierung ("3D Web Map Server"). In: Tagungsband zum Symposium Praktische Kartographie 2004 in Königslutter, Kartographische Schriften, Bd. 9, Bonn. S. 1-14.

KRAAK, M.-J.(2002): Geovisualization illustrated. ISPRS Journal of Photogrammetry and Remote Sensing. Bd. 57. S. 390-399.

KRAAK, M.-J. (2004): The role of the map in a Web-GIS environment. In: Journal of Geographical Systems. Bd. 6. Nr. 2. S. 83-93.

KRAUS K., 2007: Photogrammetry. Geometry from Images and Laser Scans. 2. Auflage. New York, Berlin.

KULAWIK R., SCHILLING, A. & A. ZIPF (2009): Landesweite 3D-Stadtmodelle im Internetauf Basis offener Standardsdes Open Geospatial Consortiums (OGC) – das

Beispiel Nordrhein-Westfalen-3D. In: Schrenk, M., Popovich, V., Engelke, D. & P. Elisei (Hrsg.), Real Corp 2009 Proceedings/Tagungsband Sitges 22-25 April 2009. S. 1-10.

KURATA, Y. 2010: From Three-Dimensional Topological Relations to Contact Relations. In: Developments in 3D Geo-Information Sciences. Neutens&De Maeyer (Hrsg.), Berlin, Heidelberg. S. 123-142.

LEDOUX, H. & C. GOLD (2007): Simultaneous storage of primal and dual three-dimensional subdivisions. In: Computers, Environment and Urban Systems. Bd. 31 (2007). S. 393-408.

LORENZ, H., TRAPP, M. & J. DÖLLNER (2009): Interaktive, multiperspektivische Ansichten für geovirtuelle 3D-Umgebungen. In: Kartographische Nachrichten. Heft 4, Jahr 2009. S. 175-181.

MACCARTHY, J. & P. GRANIERO (2006): A GIS-based borehole data management and 3D visualization system. In: Computers & Geosciences. Heft 32 (2006). Ottawa. S. 1699-1708.

MAC EACHREN, A. (2001): An evolving cognitive-semiotic approach to geographic visualization and knowledge construction. In: Information Design Journal. Bd. 10. Nr. 1. S. 26-36.

MAC EACHREN, A., GAHEGAN, M., PIKE, W., BREWER, I., CAI, G., LINGERICH, E. & F. HARDISTRY (2004): Geovisualization for knowledge construction and decision support. In: IEEE Computer Graphics and Applications. Bd. 24. Nr. 1. S. 13-17.

MAYER, C & A. ZIPF (2009): Integration and Visualization of dynamic Sensor Data into 3D Spatial Data Infrastructure in a standardized way.
<http://www.geographie.uni-bonn.de/karto/>

NADEAU D., MORELAND, J., BARU, C. & C. CROSBY (2009): Getting Beneath the Surface with the OpenEarth Framework (OEF) Virtual Globe, American Geophysical Union, Fall Meeting 2009, abstract #IN43A-1138.

NEBER, A., CLASSON, F. & M. HOWAHR (2006): Construction and usage of geological near-surface models with GSI3D – applied (hydro-)geological information for land sites and urban areas. In: Austrian Journal of Earth Sciences, Bd. 99. Wien. S. 62-69.

NEUMANN, D., SCHÖNBERG, G. & G. Strobel (2006): 3D-modeling of ground conditions for the engineering geology map of the city of Magdeburg. In: IAEG2006 Paper number 444, The Geological Society of London. London. S. 1-7.

OPEN GEOSPATIAL CONSORTIUM OGC (2005): Web 3D Service. (Hrsg.: Quadt, U. & T. Kolbe, OGC Discussion Paper Version: 0.3.0).

- OPEN GEOSPATIAL CONSORTIUM OGC (2006): Open GIS Implementation Specification for Geographic information – Simple feature access – Part I: Common architecture. (Hrsg.: Herring, J., Version: 1.2.0).
- PFEIFER N., HOLLAUS M. & BRIESE C. (2008): Geobasisdaten aus Airborne Laserscanning. In: 3Dgeo Geoconvent 2008. Wien, Graz. S. 141-158.
- POUPEAU, B. (2008): Analyses et requêtes de données géographiques 3D. Contributions de la cristallographie Géométrique. <http://tel.archives-ouvertes.fr/tel-00481924/> Université Paris-Est. Paris.
- POUPEAU B. & O. BONIN (2006): Cristage: a 3D GIS with A Logical Crystallographic Layer To Enable Complex Analysis. In: Innovations in 3D Geo Information Systems. Abdul-Rahman, Zlatanova&Coors (Hrsg.), Berlin, Heidelberg. S. 225-234.
- REQUICHA, A. (1980): Representations for Rigid Solids: Theory, Methods, and Systems. In: Computing Surveys. Bd. 12, Nr. 4.S. 437-464.
- RIEDL A. & J. KRISTEN (2010): Der Einsatz sphärischer Displays zur Visualisierung globaler Phänomene. In: Kartographische Nachrichten, 60. Jg., Heft 3/2010, S. 129-137.
- SCHARL A. (2007): Towards the Geospatial Web:Media Platforms for Managing Geotagged Knowledge Repositories. In: The Geospatial Web. How Geo-Browsers, Social Software and the Web 2.0 are Shaping the Network Society. A. Scharl & Tochtermann K. (Hrsg.). London.
- SCHILLING A., OVER M., NEUBAUER S., NEIS, P., WALENCIAK & A. ZIPF (2009): Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. UDMS 2009. 27th Urban Data Management Symposium, Ljubljana , Slovenia. http://www.geog.uni-heidelberg.de/lehrstuehle/gis/publikationen_conference_en.html
- SIEBER, R., JELLER, P. & HURNI, L. (2009): Statistische Oberflächen in einem interaktiven 3D-Atlas – Strategien und Techniken. In: Kartographische Nachrichten. Heft 4, Jahr 2009. S. 190-196.
- WOLFF, M. (2009): Ansätze zur Entwicklung geovirtueller Lagebilder. In: Kartographische Nachrichten. Heft 4, Jahr 2009. S. 192-203.
- YOO, B. & D. BRUTZMAN (2009): X3D earth terrain-tile production chain for georeferenced simulation. In: Web3D '09 Proceedings of the 14th International Conference on 3D Web Technology. Spencer, S. (Hrsg.). New York. S. 159-166.

Quellen aus dem Internet

Adobe 2010a: Adobe – Flash Player Version Penetration.

http://www.adobe.com/products/player_census/flashplayer/version_penetration.html

Stand: 11.11.2010.

Adobe 2010b: Adobe – Flash Player Statistics.

http://www.adobe.com/products/player_census/flashplayer/ Stand: 11.11.2010

Bundesinstitut für Bau-, Stadt- und Raumforschung 2009: BBSR Informationen zur Raumentwicklung.

http://www.bbsr.bund.de/cln_016/nn_23470/BBSR/DE/Veroeffentlichungen/IzR/2009/10__11/GrafikenKarten.html#doc606860bodyText4 Stand: 16.11.2010.

byoo 2010: X3D Earth Terrain Tile Production Chain – web. <http://web.byoo.net/x3d-earth-terrain-tile-production> Stand: 16.11.2010.

CitySurf 2009: CitySurf – Server Based 3D GIS (City Surf – Sunucu Tabanlı 3B CBS). <http://www.citysurf.com.tr/en/page.asp?id=7> Stand: 15.11.2010

Deshmuk 2004: Ecma TC43 – Universal 3D (U3D). <http://www.ecma-international.org/activities/Communications/tc43-2004-018.pdf> Stand: 24.08.2010.

Ecma 2010: Memento 2010. [http://www.ecma-international.org/publications/files/ECMA-](http://www.ecma-international.org/publications/files/ECMA-MEMENTOS/Ecma%20memento%202010%20public.pdf)

[MEMENTOS/Ecma%20memento%202010%20public.pdf](http://www.ecma-international.org/publications/files/ECMA-MEMENTOS/Ecma%20memento%202010%20public.pdf) Stand:24.08.2010

Deegree 2010: Building Spatial Data Infrastructures based on Free Software.

[http://www.deegree.org/deegree/portal/media-](http://www.deegree.org/deegree/portal/media-type/html/user/anon/page/default.psml/js_pane/home)

[type/html/user/anon/page/default.psml/js_pane/home](http://www.deegree.org/deegree/portal/media-type/html/user/anon/page/default.psml/js_pane/home) Stand: 14.11.2010.

DLR 2009: SRTM FAQs. http://www.dlr.de/srtm/level1/faq_en.htm Stand:

05.06.2010.

DreamingWell.com 2010: Rich Internet Application Statistics.

<http://www.riastats.com/#> Stand: 12.11.2010.

Gartner: <http://www.gartner.com/pages/story.php.id.8795.s.8.jsp> Stand: 10.09.2010.

GDI-3D 2010: GDI-3D. <http://koenigstuhl.geog.uni-heidelberg.de/gdi-3d/> Stand:

14.11.2010.

Geofabrik 2010: EOFABRIK // Downloads.

<http://www.geofabrik.de/data/download.html> Stand: 15.11.2010.

Geoscience Australia 2009: Geoscience Australia Free Data Download. Radiometric Map of Australia – 3D Atlas.

https://www.ga.gov.au/products/servlet/controller?event=GEOCAT_DETAILS&catno=68654 Stand: 03.08.2010.

Google 2010: Bildmaterial in Google Earth – Google Earth-Hilfe.
<http://earth.google.de/support/bin/answer.py?hl=de&answer=176147> Stand:
05.06.2010.

GNU Operating Systems 2007: GNU Lesser General Public License.
<http://www.gnu.org/licenses/lgpl.html> Stand: 20.06.2010.

ITU 2009: Thw World in 2009: ICT Facts and Figures. http://www.itu.int/ITU-D/ict/material/Telecom09_flyer.pdf Stand: 25.08.2010.

Khronos 2010: Re: [Public WebGL] Why WebGL? Why now and not 10 years ago?
<http://www.khronos.org/webgl/public-mailing-list/archives/1003/msg00207.html>
Stand: 19.11.2010.

NASA 2005: NASA World Wind. <http://worldwind.arc.nasa.gov/download.html>
Stand: 04.06.2010.

NPS.edu 2008: Blender Export to X3D. <https://savage.nps.edu/X3D-Edit/BlenderExportToX3d.html> Stand: 18.11.2010.

O3D team 2010: The O3D API Blog. The future of O3D. <http://o3d.blogspot.com/>
Stand: 24.08.2010.

OGC 2010: Implementation by Specification | OGC ®
<http://www.opengeospatial.org/resource/products/byspec/?specid=119> updated:
15.11.2010 Stand: 15.11.2010

Pingdom 2010: Does Internet Explorer have more than a billion users? | Royal Pingdom. <http://royal.pingdom.com/2010/04/16/does-internet-explorer-have-more-than-a-billion-users/> Stand: 24.08.2010.

Ricky's Bing Maps Blogg 2010: Bing Maps Taxi Fare Calculator.
<http://rbrundritt.wordpress.com/2010/08/03/bing-maps-taxi-fare-calculator/> Stand:
10.11.2010.

SIGGRAPH 2007: The Carto Project. <http://www.siggraph.org/~rhyne/carto/> Stand:
18.08.2008.

Silverlight 2010: Get Started : the Official Microsoft Silverlight Website.
<http://www.silverlight.net/getstarted/> Stand: 12.11.2010.

StatCounter 2010: Top 5 Browsers from July 9 to August 10 | StatCounter Global Stats. <http://gs.statcounter.com/> Stand: 24.08.2010.

The WebGL Cookbook 2010: WebGL: Frequently Asked Questions – The WebGL Cookbook.
http://learningwebgl.com/cookbook/index.php/WebGL:_Frequently_Asked_Questions#What_about_Microsoft_and_Internet_Explorer.3F Stand: 24.08.2010.

rez3d.com 2007: RezIndex. <http://www.rez3d.com/> Stand: 16.11.2010.

Via Windows Live2009: Virtual Earth Frequently Asked Questions.
<http://www.viawindowslive.com/VirtualEarth/VirtualEarthFAQ.aspx> Stand:
09.11.2010.

Web3D Consortium 2010:
http://web3d.org/x3d/wiki/index.php/Interview_with_Research_2.0 Stand:
10.06.2010.

x3dom 2010: X3DOM 1.0 – home. http://www.x3dom.org/?page_id=2 Stand:
19.11.2010.

Anhang

I: Screenshots des visualisierten Modells

II: X3D-Code

Auszüge

- Funktion *Explodieren*
- Funktion *Profilschnitte*
- Funktion *Überhöhung*

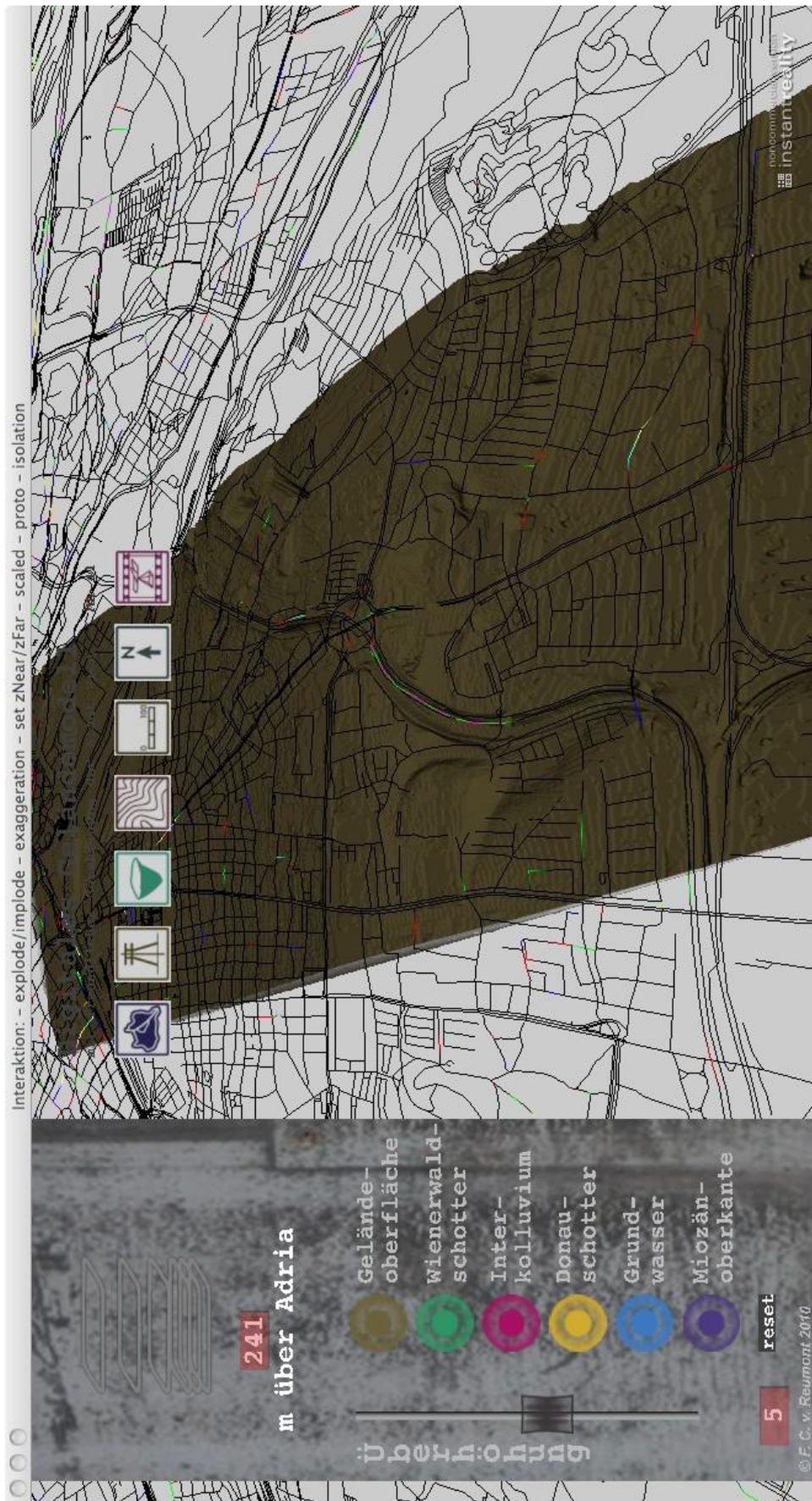
Vollständige Hauptdatei

I

Screenshots des visualisierten Modells

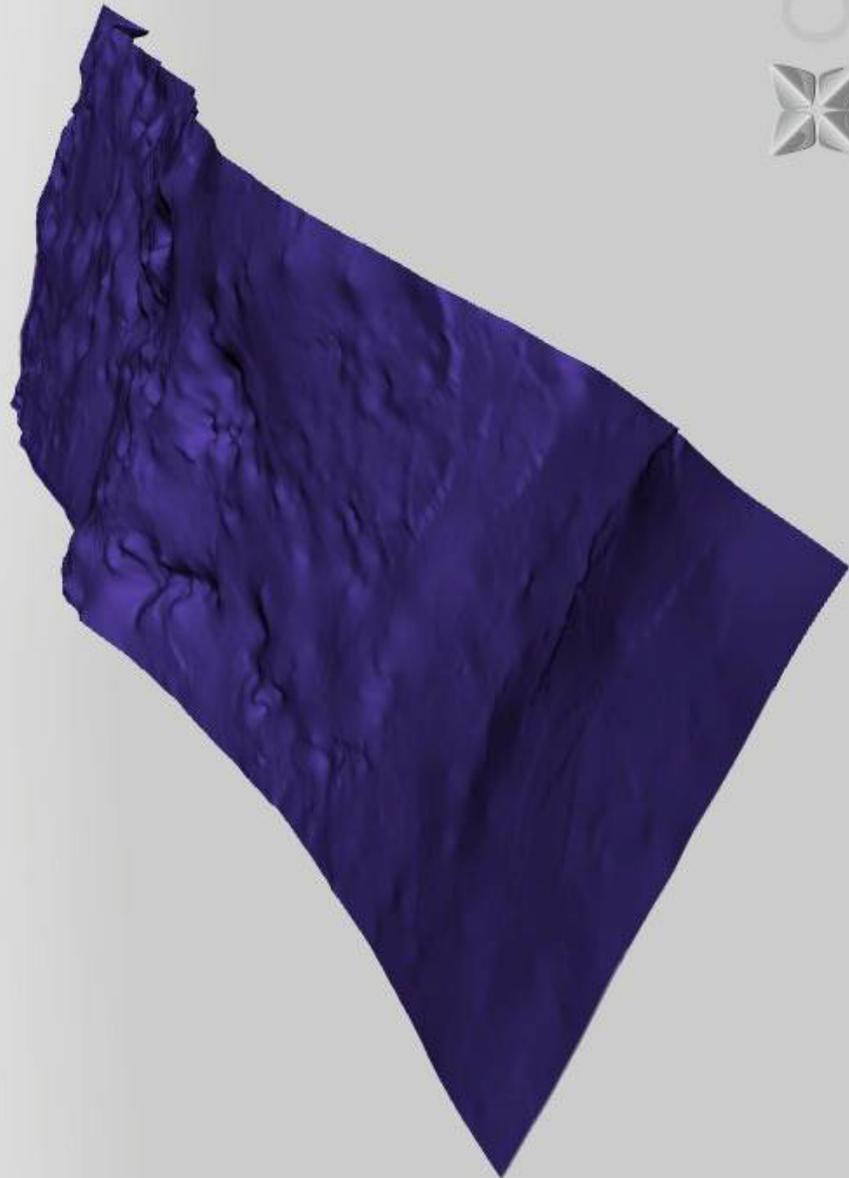


Screenshot I: Default-Ansicht des 3D-Modells



Screenshot II: Ansicht der Hauptdatei mit eingeblendetem Stadtplan

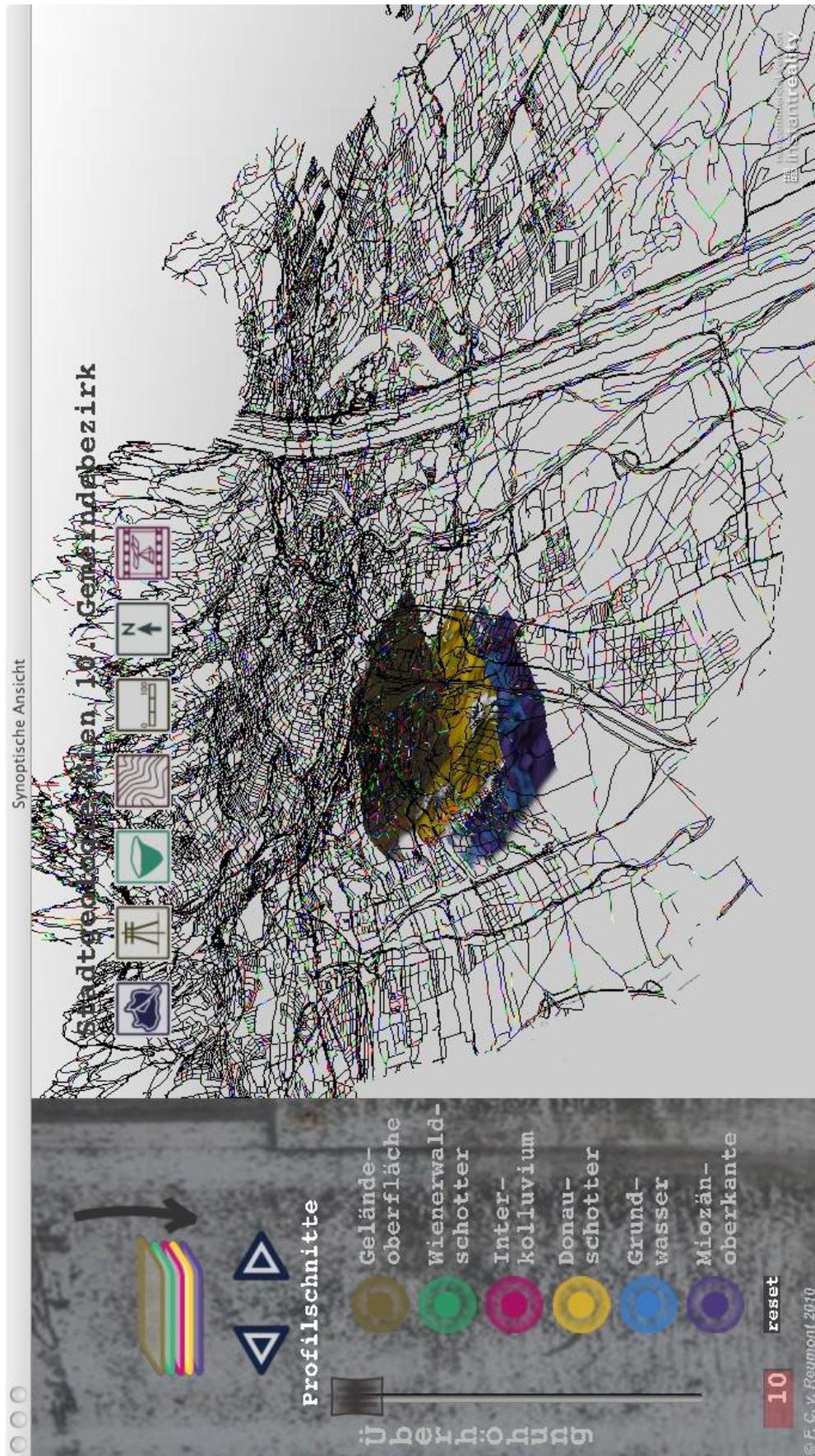
**Stadtgeologie Wien
10. Gemeindebezirk**



Screenshot III: Einzelansicht der Schicht *Miozänoberkannte*



Screenshot IV: Ansicht mit Orthofoto der Stadt Wien – MA 41 mit Überhöhung



Screenshot V: Ansicht der Schichten mit Profilschnitt und Stadtplan

II X3D-Code

Auszüge:

- Funktion *Explodieren*
- Funktion *Profilschnitte*
- Funktion *Überhöhung*

Vollständige Hauptdatei

Funktion *Explodieren*

```

<!-- Implode _ Explode Buttons -->
<Transform DEF="explode_implode" translation="0.009 0.012 0" center=".03 .03 0"
rotation="0 0 0 0">
  <Transform>
    ...
    <TouchSensor DEF="explode" description="Click to explode!" />
  </Transform>
    ...
    <TouchSensor DEF="implode" description="Click to implode!" />
  </Transform>
</Transform>

<!-- explodieren -->
<ROUTE
  fromNode="Zeit_ex" fromField="fraction_changed"
  toNode="explode_int" toField="set_fraction" />
<ROUTE
  fromNode="explode_int" fromField="value_changed"
  toNode="explode_implode" toField="rotation" />

<!-- implodieren -->
<ROUTE
  fromNode="Zeit_im" fromField="fraction_changed"
  toNode="implode_int" toField="set_fraction" />
<ROUTE
  fromNode="implode_int" fromField="value_changed"
  toNode="explode_implode" toField="rotation" />

<!-- PROTOTYPE Extern Deklaration -->
...
<!-- PROTOTYPE Instanziierung Schichten externalLib_strata.x3d -->
<Transform DEF="weltall">
<Group>

<!-- PROTOTYPE dgm_orig -->
  <Transform DEF="dgm" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_dgm" name="dgm_coarse" />
    <PositionInterpolator DEF="up_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
    <PositionInterpolator DEF="down_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
  </Transform>

<!-- PROTOTYPE wws_orig -->
  <Transform DEF="wws" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_wws" name="wws_orig" />
    <PositionInterpolator DEF="up_wws" key="0, 0.4" keyValue="0 0 0, 0 -2.5 0" />
    <PositionInterpolator DEF="down_wws" key="0, 0.4" keyValue="0 -2.5 0, 0 0 0" />
  </Transform>

<!-- PROTOTYPE ik_orig -->
...
<!-- explode -->
<ROUTE
  fromNode="explode" fromField="touchTime"
  toNode="Zeit_ex" toField="startTime" />

<ROUTE
  fromNode="Zeit_ex" fromField="fraction_changed"

```

```
        toNode="up_dgm" toField="set_fraction" />
<ROUTE
  fromNode="up_dgm" fromField="value_changed"
  toNode="dgm" toField="translation" />
...
<!-- implode -->
<ROUTE
  fromNode="implode" fromField="touchTime"
  toNode="Zeit_im" toField="startTime" />

<ROUTE
  fromNode="Zeit_im" fromField="fraction_changed"
  toNode="down_dgm" toField="set_fraction" />
<ROUTE
  fromNode="down_dgm" fromField="value_changed"
  toNode="dgm" toField="translation" />
...
</Scene>
</X3D>
```

Funktion *Profilschnitte*

```

...
  <!-- Profilschnitte -->
<Transform DEF="vorzurueck" translation=".03 -.004 0">
  <Transform DEF="zurueck" translation="-.02 0 0">
...
    <TouchSensor DEF="zurueckSens" />
  </Transform>
  <Transform DEF="vor" translation=".005 0 0">
...
    <TouchSensor DEF="vorSens" />
  </Transform>
...
</Transform> <!-- vorzurueck -->
<!-- PROTOTYPE Extern Deklaration -->
...
<!-- PROTOTYPE Instanziierung Schichten externalLib_strata.x3d -->

<Transform DEF="weltall">
<Group>

<!-- PROTOTYPE dgm_orig -->

  <Transform DEF="dgm" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_dgm" name="dgm_coarse" />
    <PositionInterpolator DEF="up_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
    <PositionInterpolator DEF="down_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
  </Transform>

<!-- PROTOTYPE wws_orig -->

  <Transform DEF="wws" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_wws" name="wws_orig" />
    <PositionInterpolator DEF="up_wws" key="0, 0.4" keyValue="0 0 0, 0 -2.5 0" />
    <PositionInterpolator DEF="down_wws" key="0, 0.4" keyValue="0 -2.5 0, 0 0 0" />
  </Transform>

<!-- PROTOTYPE ik_orig -->
...
<!-- Profilschnitte -->

<ROUTE
  fromNode="vorSens" fromField="isActive"
  toNode="vorFilter" toField="set_boolean" />
<ROUTE
  fromNode="vorFilter" fromField="inputTrue"
  toNode="auswahlSeq" toField="next" />

<ROUTE
  fromNode="zurueckSens" fromField="isActive"
  toNode="zurueckFilter" toField="set_boolean" />
<ROUTE
  fromNode="zurueckFilter" fromField="inputTrue"
  toNode="auswahlSeq" toField="previous" />

<Script DEF="int_to_ps">
  <field accessType="inputOnly" name="schnitt" type="SFInt32" />
  <field accessType="outputOnly" name="dgm_texture" type="MFString" />
  <field accessType="outputOnly" name="wws_texture" type="MFString" />
  <field accessType="outputOnly" name="ik_texture" type="MFString" />
  <field accessType="outputOnly" name="jds_texture" type="MFString" />
  <field accessType="outputOnly" name="gw_texture" type="MFString" />
  <field accessType="outputOnly" name="mio_texture" type="MFString" />

```

```
<![CDATA[ecmascript:
function schnitt(wert) {
    dgm_texture = new MFString('dgm_ps' + wert + '.png');
    wws_texture = new MFString('wws_ps' + wert + '.png');
    ik_texture = new MFString('ik_ps' + wert + '.png');
    jds_texture = new MFString('jds_ps' + wert + '.png');
    gw_texture = new MFString('gw_ps' + wert + '.png');
    mio_texture = new MFString('mio_ps' + wert + '.png');
}
]]>
</Script>

<ROUTE
fromNode="auswahlSeq" fromField="value_changed"
toNode="int_to_ps" toField="schnitt" />
<ROUTE
fromNode="int_to_ps" fromField="dgm_texture"
toNode="proto_dgm" toField="texture_dgm" />
...

```

Funktion *Überhöhung*

```

...
<!-- Schieberegler mit Dragsensor und Funktion -->
<Group>
  <Transform DEF="trans_regler" translation=".005 -.1115 0">
    <PlaneSensor DEF="sens_regler" minPosition="0 0" maxPosition="0 .09" />
    <Transform DEF="regler">
      <Shape>
        ...
      </Shape>
    </Transform><!-- regler -->
  <Transform DEF="scalebar" translation="0 0.045 0">
    <Shape>
      <Appearance>
        <Material emissiveColor="0 0 0" shininess="1" />
      </Appearance>
      <Cylinder radius=".001" height=".09" />
    </Shape>
  </Transform> <!-- scalebar -->

<Transform DEF="value_display" translation="0 -.02 0">
  <Shape>
    <Appearance>
      <Material diffuseColor="1 0 0" transparency="0.7" shininess="1" />
    </Appearance>
    <Box size=".015 .0075 .001" />
  </Shape>
  <Transform DEF="text_disp" translation="-0.002 -.002 0">
    <Shape>
      <Appearance>
        <Material emissiveColor="1 1 1" />
      </Appearance>
      <Text DEF="value_exagg" string="1">
        <FontStyle size=".0075" family="TYPEWRITER" />
      </Text>
    </Shape>
  </Transform> <!-- text_disp -->
  <Transform DEF="scalebar_title" translation="-0.01 .11 0">
    <Shape>
      ...
    </Shape>
  </Transform> <!-- scalebar_title -->
</Transform> <!-- value display -->
</Transform> <!-- transregler -->
</Group>
...

<!-- PROTOTYPE Extern Deklaration -->
...

<!-- PROTOTYPE Instanziierung Schichten externalLib_strata.x3d -->

<Transform DEF="weltall">
<Group>

<!-- PROTOTYPE dgm_orig -->

  <Transform DEF="dgm" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_dgm" name="dgm_coarse" />
    <PositionInterpolator DEF="up_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
    <PositionInterpolator DEF="down_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
  </Transform>

<!-- PROTOTYPE wws_orig -->

  <Transform DEF="wws" translation="0 0 0" scale='.01 .01 .01'>
    <ProtoInstance DEF="proto_wws" name="wws_orig" />
    <PositionInterpolator DEF="up_wws" key="0, 0.4" keyValue="0 0 0, 0 -2.5 0" />
    <PositionInterpolator DEF="down_wws" key="0, 0.4" keyValue="0 -2.5 0, 0 0 0" />
  </Transform>

```

```
<!-- PROTOTYPE ik_orig -->
...

<!-- Überhöhung -->
<Script DEF="translation_to_scale">
  <field accessType="inputOnly" name="from_translation" type="SFVec3f" />
  <field accessType="outputOnly" name="to_scale" type="SFVec3f" />
  <field accessType="outputOnly" name="new_transl_strata" type="SFVec3f" />
  <field accessType="outputOnly" name="new_value" type="MFString" />

  <![CDATA[ecmascript:
    function from_translation(wert) {
      to_scale = new SFVec3f(.01, wert.y+.01, .01);
      new_transl_strata = new SFVec3f(0, 0-200*wert.y, 0);
      ex_value = (wert.y*100+1);
      round_value = Math.round(ex_value, 4);
      new_value = new MFString(round_value);
      if (new_value == '1') {
        new_value = new MFString('keine');
      } else {
        new_value = new MFString(round_value);
      }
    }
  ]]>
  print(to_scale);
}
]]>
</Script>

<ROUTE
  fromNode="sens_regler" fromField="translation_changed"
  toNode="translation_to_scale" toField="from_translation" />

<ROUTE
  fromNode="translation_to_scale" fromField="new_value"
  toNode="value_exagg" toField="string" />

<ROUTE
  fromNode="translation_to_scale" fromField="new_transl_strata"
  toNode="weltall" toField="translation" />

<ROUTE
  fromNode="translation_to_scale" fromField="to_scale"
  toNode="dgm" toField="scale" />

<ROUTE
  fromNode="translation_to_scale" fromField="to_scale"
  toNode="jds" toField="scale" />
...

```

Vollständige Hauptdatei

Hauptdatei

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
  http://www.web3d.org/specifications/x3d-3.0.dtd">
3 <X3D xmlns:xsd='http://www.w3.org/2001/XMLSchema-instance' profile='Immersive'
  version='3.0'
  xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.0.xsd'>
4   <Scene DEF='scene'>
5     <WorldInfo info="Masterarbeit Version 9" title='Synoptische Ansicht'/>
6     <Background skyAngle="1.67 1.97" skyColor="1 1 1, 1 1 1, 0.8 0.8 0.8"/>
7     <!-- <DirectionalLight ambientIntensity='.02' direction='0.6 -0.5 0.6' global="true"
  /> -->
8     <!-- <SpotLight location="33.6941 16.1361 -33.3977" direction="-0.166068 0.98518
  0.0428961 2.86137" ambientIntensity=".4" beamWidth=".2" cutOffAngle=".5" radius="100" /> -->
9     <NavigationInfo speed='50' type="ANY" />
10
11    <Viewpoint description='default' position='41.113 64.1828 128.482'
  orientation="-0.821727 0.549170 0.152465 0.651704" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
12    <Viewpoint description='Profilschnitte' position='-58.2761 26.3194 91.5617'
  orientation="-0.405167 -0.911761 0.067314 0.905728" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
13    <Viewpoint description='Plan 1' position='-20.5802 309.486 69.1752'
  orientation="-0.988988 -0.068542 -0.131179 1.2331" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
14    <Viewpoint description='Plan 2 (Zentrum)' position='8.52787 119.267 51.773'
  orientation="-0.92861 0.357768 -0.0984112 1.30222" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
15    <Viewpoint description='Plan 3' position='48.5502 73.4096 157.376'
  orientation="-0.722935 0.674118 0.151426 0.771158" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
16    <Viewpoint description='Plan nah' position='8.54694 20.3831 75.8895'
  orientation="-0.986391 0.16021 -0.03695 0.594464" centerOfRotation='11 2.00 36.00'
  zNear=".01" zFar="9000" />
17    <Viewpoint description='Plan mittel' position='102.342 86.8429 116.25'
  orientation="-0.532087 0.775013 0.340937 1.0025" centerOfRotation='11 2.00 36.00' zNear=".01"
  zFar="9000" />
18
19    <NavigationInfo headlight="false" speed='50' type="ANY" />
20
21
22
23 <!-- Head Up Display -->
24 <!-- ProximitySensor mit Switch: ex-/implode -->
25
26 <ProximitySensor DEF="All" center="0 0 0" size="1000 1000 1000" />
27
28 <Transform DEF="HUD" translation="0 0 0">
29   <Transform DEF="HUDpos" translation="-.18 .04 -.25">
30
31     <!-- Titel -->
32     <Transform DEF="title" translation=".1 .052 0">
33       <Shape>
34         <Appearance>
35           <Material diffuseColor=".3 .3 .3" />
36         </Appearance>
37         <Text string="Hydrogeologie Wien 10. Gemeindebezirk">
38           <FontStyle family="TYPEWRITER" justify="FIRST" size=".008" style="BOLD" />
39         </Text>
40       </Shape>
41     </Transform>
42   <Transform DEF="schrift_hintergrund">
43     <Shape>
44       <Appearance>
45         <Material diffuseColor="1 1 1" transparency=".2" />
46       </Appearance>
47       <IndexedFaceSet DEF="schrift_extent" coordIndex="0 1 2 3 -1">
48         <Coordinate point="
49           .285 0.065 -.01,
50           .095 0.065 -.01,

```

Hauptdatei

```

51         .095 .047 -.01,
52         .285 .047 -.01" />
53     </IndexedFaceSet>
54 </Shape>
55 </Transform>
56
57 <!-- HUD quer -->
58 <Transform DEF="bcgp_button" translation="0.1 .027 0">
59 <Transform DEF="plan_button">
60 <Shape>
61 <Appearance>
62 <Material diffuseColor="0.91 0.745 1" />
63 <ImageTexture url="plan_button.png" />
64 </Appearance>
65 <IndexedFaceSet DEF="button" coordIndex="0 1 2 3 -1">
66 <Coordinate point="
67     0 0 0,
68     .015 0 0,
69     .015 .015 0,
70     0 .015 0" />
71 </IndexedFaceSet>
72 </Shape>
73 <TouchSensor DEF="planSens" description="Click to add/remove" />
74 </Transform>
75
76 <Transform DEF="bohrungen_button" translation=".02 0 0">
77 <Shape>
78 <Appearance>
79 <Material diffuseColor="0.91 0.745 1" />
80 <ImageTexture url="bohrungen_button.png" />
81 </Appearance>
82 <IndexedFaceSet USE="button" />
83 </Shape>
84 <TouchSensor DEF="bohrungenSens" description="Click to add/remove!" />
85 </Transform>
86
87 <Transform DEF="gruben_button" translation=".04 0 0">
88 <Shape>
89 <Appearance>
90 <Material diffuseColor="0.91 0.745 1" />
91 <ImageTexture url="gruben_button.png" />
92 </Appearance>
93 <IndexedFaceSet USE="button" />
94 </Shape>
95 <TouchSensor DEF="grubenSens" description="Click to add/remove!" />
96 </Transform>
97
98 <Transform DEF="contour_button" translation=".06 0 0">
99 <Shape>
100 <Appearance>
101 <Material diffuseColor="0.91 0.745 1" />
102 <ImageTexture url="contour_button.png" />
103 </Appearance>
104 <IndexedFaceSet USE="button" />
105 </Shape>
106 <TouchSensor DEF="contourSens" description="Click to add/remove!" />
107 </Transform>
108
109 <Transform DEF="massstab_button" translation=".08 0 0">
110 <Shape>
111 <Appearance>
112 <Material diffuseColor="0.91 0.745 1" />
113 <ImageTexture url="massstab_button.png" />
114 </Appearance>
115 <IndexedFaceSet USE="button" />
116 </Shape>
117 <TouchSensor DEF="massstabSens" description="Click to add/remove!" />
118 </Transform>
119

```

2

```
Hauptdatei
120 <Transform DEF="nord_button" translation=".1 0 0">
121   <Shape>
122     <Appearance>
123       <Material diffuseColor="0.91 0.745 1" />
124       <ImageTexture url="nord_button.png" />
125     </Appearance>
126     <IndexedFaceSet USE="button" />
127   </Shape>
128   <TouchSensor DEF="nordSens" description="Click to add/remove!" />
129 </Transform>
130
131 <!-- o-foto-button -->
132 <Anchor description="Click to add/remove!" url="child_ofoto.x3d">
133 <Transform DEF="ofoto_button" translation=".12 0 0">
134   <Shape>
135     <Appearance>
136       <Material diffuseColor="0.91 0.745 1" />
137       <ImageTexture url="ofoto_button.png" />
138     </Appearance>
139     <IndexedFaceSet USE="button" />
140   </Shape>
141 </Transform>
142 </Anchor>
143
144 </Transform> <!-- bcgp_button -->
145
146
147 <!-- Dashboard (Hintergrund) -->
148
149 <Transform DEF="dashboard">
150   <Shape>
151     <Appearance>
152       <Material diffuseColor="1 1 0" />
153       <ImageTexture url="dashboard.jpg" />
154     </Appearance>
155     <IndexedFaceSet DEF="dash_field" coordIndex="0 1 2 3 -1">
156       <Coordinate point="
157         -.02 .1 -.01,
158         -.02 -.15 -.01,
159         0.08 -.15 -.01,
160         .08 .1 -.01
161       " />
162     </IndexedFaceSet>
163   </Shape>
164 </Transform>
165
166
167 <!-- Implode _ Explode Buttons -->
168
169 <Transform DEF="explode_implode" translation="0.009 0.012 0" center=".03 .03 0"
rotation="0 0 0">
170 <Transform>
171   <Shape>
172     <Appearance>
173       <Material emissiveColor="1 1 1" />
174       <ImageTexture url="explode.png" />
175     </Appearance>
176     <IndexedFaceSet DEF="expl_field" coordIndex="0 1 2 3 -1">
177       <Coordinate point="
178         0 0 0,
179         .05 0 0,
180         .05 .05 0,
181         0 .05 0" />
182     </IndexedFaceSet>
183   </Shape>
184   <TouchSensor DEF="explode" description="Click to explode!" />
185 </Transform>
186 <Transform translation=".01 0 -.001">
187   <Shape>
```

Hauptdatei

```

188     <Appearance>
189       <Material emissiveColor="1 1 1" />
190       <ImageTexture url="implode.png" />
191     </Appearance>
192     <IndexedFaceSet DEF="impl_field" coordIndex="2 3 0 1 -1" ccw="false">
193       <Coordinate point="
194         0 0 0,
195         .05 0 0,
196         .05 .05 0,
197         0 .05 0" />
198     </IndexedFaceSet>
199   </Shape>
200   <TouchSensor DEF="implode" description="Click to implode!" />
201 </Transform>
202 </Transform>
203
204   <!-- Profilschnitte -->
205 <Transform DEF="vorzurueck" translation=".03 -.004 0">
206   <Transform DEF="zurueck" translation="-.02 0 0">
207     <Shape>
208       <Appearance>
209         <Material diffuseColor="0.91 0.745 1" />
210         <ImageTexture url="zurueck.png" />
211       </Appearance>
212       <IndexedFaceSet DEF="zurueck_field" coordIndex="0 1 2 3 -1">
213         <Coordinate point="
214           0 0 0,
215           .015 0 0,
216           .015 .015 0,
217           0 .015 0" />
218       </IndexedFaceSet>
219     </Shape>
220     <TouchSensor DEF="zurueckSens" />
221   </Transform>
222   <Transform DEF="vor" translation=".005 0 0">
223     <Shape>
224       <Appearance>
225         <Material diffuseColor="0.91 0.745 1" />
226         <ImageTexture url="vor.png" />
227       </Appearance>
228       <IndexedFaceSet USE="zurueck_field" />
229     </Shape>
230     <TouchSensor DEF="vorSens" />
231   </Transform>
232   <Transform translation="0 -.007 0">
233     <Shape>
234       <Appearance>
235         <Material emissiveColor="1 1 1" />
236       </Appearance>
237       <Text string="Profilschnitte">
238         <FontStyle size=".0065" family="TYPEWRITER" justify="MIDDLE" style="BOLD" />
239       </Text>
240     </Shape>
241   </Transform>
242 </Transform> <!-- vorzurueck -->
243
244   <!-- Schieberegler mit Dragsensor und Funktion -->
245 <Group>
246   <Transform DEF="trans_regler" translation=".005 -.1115 0">
247     <PlaneSensor DEF="sens_regler" minPosition="0 0" maxPosition="0 .09" />
248     <Transform DEF="regler">
249       <Shape>
250         <Appearance>
251           <Material diffuseColor="0.91 0.745 1" />
252           <ImageTexture url="schieberegler.png" />
253         </Appearance>
254         <IndexedFaceSet DEF="regler_field" coordIndex="2 3 0 1 -1">
255           <Coordinate point="
256             -.009 -.01 0.001,

```

4

```
Hauptdatei
257         .011 -.01 0.001,
258         .011 .01 0.001,
259         -.009 .01 0.001" />
260     </IndexedFaceSet>
261 </Shape>
262 </Transform><!-- regler -->
263 <Transform DEF="scalebar" translation="0 0.045 0">
264 <Shape>
265     <Appearance>
266     <Material emissiveColor="0 0 0" shininess="1" />
267     </Appearance>
268     <Cylinder radius=".001" height=".09" />
269 </Shape>
270 </Transform> <!-- scalebar -->
271
272 <Transform DEF="value_display" translation="0 -.02 0">
273 <Shape>
274     <Appearance>
275     <Material diffuseColor="1 0 0" transparency="0.7" shininess="1" />
276     </Appearance>
277     <Box size=".015 .0075 .001" />
278 </Shape>
279 <Transform DEF="text_disp" translation="-0.002 -.002 0">
280 <Shape>
281     <Appearance>
282     <Material emissiveColor="1 1 1" />
283     </Appearance>
284     <Text DEF="value_exagg" string="1">
285     <FontStyle size=".0075" family="TYPEWRITER" />
286     </Text>
287 </Shape>
288 </Transform> <!-- text_disp -->
289 <Transform DEF="scalebar_title" translation="-0.01 .11 0">
290 <Shape>
291     <Appearance>
292     <Material emissiveColor="1 1 1" />
293     </Appearance>
294     <Text DEF="sb_title" string="Überhöhung">
295     <FontStyle size=".009" family="TYPEWRITER" horizontal="false" topToBottom
296     ="true" style="BOLD" />
297     </Text>
298 </Shape>
299 </Transform> <!-- scalebar_title -->
300 </Transform> <!-- value display -->
301 </Transform> <!-- transregler -->
302 </Group>
303
304 <!-- Legende fuer den Iso-Mode -->
305 <Group DEF="iso">
306 <Transform DEF="legend" translation=".022 -.05 0">
307 <Transform DEF="allfields">
308 <Anchor url="child_dgm.x3d" description="Click to isolate layer!">
309 <Transform DEF="dgm_legend" translation="0 .015 0">
310 <Shape>
311     <Appearance>
312     <Material diffuseColor="0.91 0.745 1" />
313     <ImageTexture url="dgm_button.png" />
314     </Appearance>
315     <IndexedFaceSet DEF="field" coordIndex="0 1 2 3 -1">
316     <Coordinate point="
317     0 0 0,
318     .015 0 0,
319     .015 .015 0,
320     0 .015 0" />
321 </IndexedFaceSet>
322 </Shape>
323 </Transform>
324 </Anchor> <!-- dgm_legend -->
```

Hauptdatei

```

325 <Anchor url="child_wws.x3d" description="Click to isolate layer!">
326   <Transform DEF="wws_legend" translation="0 -.0025 0">
327     <Shape>
328       <Appearance>
329         <Material diffuseColor="0.941 0.949 0.192" />
330         <ImageTexture url="wws_button.png" />
331       </Appearance>
332       <IndexedFaceSet USE="field" />
333     </Shape>
334   </Transform>
335 </Anchor> <!-- wws_legend -->
336
337 <Anchor url="child_ik.x3d" description="Click to isolate layer!">
338   <Transform DEF="ik_legend" translation="0 -.02 0">
339     <Shape>
340       <Appearance>
341         <Material diffuseColor="0.509, 0.172, 0.847" />
342         <ImageTexture url="ik_button.png" />
343       </Appearance>
344       <IndexedFaceSet USE="field" />
345     </Shape>
346   </Transform>
347 </Anchor> <!-- ik_legend -->
348
349 <Anchor url="child_jds.x3d" description="Click to isolate layer!">
350   <Transform DEF="jds_legend" translation="0 -.0375 0">
351     <Shape>
352       <Appearance>
353         <Material diffuseColor="0.870, 0.090, 0.258" />
354         <ImageTexture url="jds_button.png" />
355       </Appearance>
356       <IndexedFaceSet USE="field" />
357     </Shape>
358   </Transform>
359 </Anchor> <!-- jds_legend -->
360
361 <Anchor url="child_gw.x3d" description="Click to isolate layer!">
362   <Transform DEF="gw_legend" translation="0 -.055 0">
363     <Shape>
364       <Appearance>
365         <Material diffuseColor="0.360, 0.901, 0.396" />
366         <ImageTexture url="gw_button.png" />
367       </Appearance>
368       <IndexedFaceSet USE="field" />
369     </Shape>
370   </Transform>
371 </Anchor> <!-- gw_legend -->
372
373 <Anchor url="child_mio.x3d" description="Click to isolate layer!">
374   <Transform DEF="mio_legend" translation="0 -.0725 0">
375     <Shape>
376       <Appearance>
377         <Material diffuseColor="0.360, 0.901, 0.396" />
378         <ImageTexture url="mio_button.png" />
379       </Appearance>
380       <IndexedFaceSet USE="field" />
381     </Shape>
382   </Transform>
383 </Anchor> <!-- mio_legend -->
384
385   <Transform DEF="reset_legend" translation="0 -.082 0">
386     <Transform DEF="schalter" scale="1 0.3 1">
387       <Shape>
388         <Appearance>
389           <Material diffuseColor="0.3, 0.3, 0.3" />
390         </Appearance>
391         <IndexedFaceSet USE="field" />
392       </Shape>

```

6

Hauptdatei

```
394     </Transform> <!-- schalter -->
395     <Transform DEF="text_reset" translation="0.001 0.001 0.001">
396     <Shape>
397     <Appearance>
398     <Material emissiveColor="1 1 1" />
399     </Appearance>
400     <Text DEF="reset_button" string="reset">
401     <FontStyle size=".005" family="TYPEWRITER" />
402     </Text>
403     </Shape>
404     </Transform> <!-- text_reset -->
405     <TouchSensor DEF="reset" description="Click to isolate layer!" />
406     </Transform> <!-- reset_legend -->
407
408 </Transform> <!-- allfields -->
409
410 <!-- Beschriftung -->
411 <Transform DEF="dgm_text" translation="0.018 .024 0">
412 <Shape>
413 <Appearance>
414 <Material emissiveColor="1 1 1" />
415 </Appearance>
416 <Text DEF="text_dgm" string="'Gelände-', "oberfläche'">
417 <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
418 </Text>
419 </Shape>
420 </Transform> <!-- dgm_text -->
421
422 <Transform DEF="wvs_text" translation="0.018 .006 0">
423 <Shape>
424 <Appearance>
425 <Material emissiveColor="1 1 1" />
426 </Appearance>
427 <Text DEF="text_wvs" string="'Wienerwald-', "schotter'">
428 <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
429 </Text>
430 </Shape>
431 </Transform> <!-- wvs_text -->
432
433 <Transform DEF="ik_text" translation="0.018 -.011 0">
434 <Shape>
435 <Appearance>
436 <Material emissiveColor="1 1 1" />
437 </Appearance>
438 <Text DEF="text_ik" string="'Inter-', "kolluvium'">
439 <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
440 </Text>
441 </Shape>
442 </Transform> <!-- ik_text -->
443
444 <Transform DEF="jds_text" translation="0.018 -.028 0">
445 <Shape>
446 <Appearance>
447 <Material emissiveColor="1 1 1" />
448 </Appearance>
449 <Text DEF="text_jds" string="'Donau-', "schotter'">
450 <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
451 </Text>
452 </Shape>
453 </Transform> <!-- jds_text -->
454
455 <Transform DEF="gw_text" translation="0.018 -.046 0">
456 <Shape>
457 <Appearance>
458 <Material emissiveColor="1 1 1" />
459 </Appearance>
460 <Text DEF="text_gw" string="'Grund-', "wasser'">
461 <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
462 </Text>
```

7

Hauptdatei

```

463     </Shape>
464 </Transform> <!-- mio_text -->
465
466     <Transform DEF="mio_text" translation="0.018 -.064 0">
467     <Shape>
468     <Appearance>
469     <Material emissiveColor="1 1 1" />
470     </Appearance>
471     <Text DEF="text_mio" string="'Miozän-", "oberkante"'>
472     <FontStyle size=".0065" family="TYPEWRITER" style="BOLD" />
473     </Text>
474     </Shape>
475 </Transform> <!-- mio_text -->
476
477 </Transform> <!-- legend -->
478 </Group>
479
480 <!-- Copyright -->
481
482 <Transform DEF="copyright_text" translation="-.01 -.141 0">
483 <Shape>
484 <Appearance>
485 <Material diffuseColor=".9 .9 .9" />
486 </Appearance>
487 <Text DEF="text_copyright" string="'© F. C. v. Reumont 2010"'>
488 <FontStyle size=".0045" family="SANS" style="BOLD" />
489 </Text>
490 </Shape>
491 </Transform> <!-- Copyright -->
492
493 </Transform>
494 </Transform> <!-- End HUD -->
495
496 <ROUTE
497 fromNode="All" fromField="position_changed"
498 toNode="HUD" toField="translation" />
499
500 <ROUTE
501 fromNode="All" fromField="orientation_changed"
502 toNode="HUD" toField="rotation" />
503
504
505 <!-- explode/implode -->
506 <!-- Sensoren -->
507 <TimeSensor DEF="Zeit_ex" cycleInterval=".75" loop="false"/>
508 <TimeSensor DEF="Zeit_im" cycleInterval=".75" loop="false"/>
509
510 <OrientationInterpolator DEF="explode_int" key="0 0.2" keyValue="0 0 0, 0 1 0 3.142"
511 />
512 <OrientationInterpolator DEF="implode_int" key="0 0.2" keyValue="0 1 0 3.142, 0 0 0 0"
513 />
514
515 <!-- explodieren -->
516 <ROUTE
517 fromNode="Zeit_ex" fromField="fraction_changed"
518 toNode="explode_int" toField="set_fraction" />
519 <ROUTE
520 fromNode="explode_int" fromField="value_changed"
521 toNode="explode_implode" toField="rotation" />
522
523 <!-- implodieren -->
524 <ROUTE
525 fromNode="Zeit_im" fromField="fraction_changed"
526 toNode="implode_int" toField="set_fraction" />
527 <ROUTE
528 fromNode="implode_int" fromField="value_changed"
529 toNode="explode_implode" toField="rotation" />

```

8

Hauptdatei

```
530
531
532 <!-- event utilities bcgp -->
533 <BooleanToggle DEF="jaNein_bohrungen" toggle="false" />
534 <BooleanToggle DEF="jaNein_contour" toggle="false" />
535 <BooleanToggle DEF="jaNein_gruben" toggle="false" />
536 <BooleanToggle DEF="jaNein_plan" toggle="false" />
537 <BooleanToggle DEF="jaNein_nord" toggle="true" />
538 <BooleanToggle DEF="jaNein_masstab" toggle="true" />
539
540 <BooleanFilter DEF="vorFilter" />
541 <BooleanFilter DEF="zurueckFilter" />
542 <IntegerSequencer DEF="auswahlSeq" key="0, .33, .66, 1" keyValue="0 1 2 3" />
543
544
545 <!-- Schieberegler für Überhöhung -->
546 <ROUTE
547     fromNode="sens_regler" fromField="translation_changed"
548     toNode="regler" toField="set_translation" />
549
550
551
552
553
554 <!-- PROTOTYPE Extern Deklaration -->
555 <!-- PROTOTYPE Extern Deklaration -->
556
557 <ExternProtoDeclare name="dgm_LOD" url="dgm_proto_LOD.x3d">
558     <field name="texture_dgm" type="MFString" accessType="inputOutput" />
559     <field name="farbe_dgm" type="SFColor" accessType="inputOutput" />
560 </ExternProtoDeclare>
561
562 <ExternProtoDeclare name="dgm_coarse" url="dgm_proto_coarse.x3d">
563     <field name="texture_dgm" type="MFString" accessType="inputOutput" />
564     <field name="farbe_dgm" type="SFColor" accessType="inputOutput" />
565 </ExternProtoDeclare>
566
567
568 <ExternProtoDeclare name="wvs_orig" url="wvs_proto.x3d">
569     <field name="texture_wvs" type="MFString" accessType="inputOutput" />
570 </ExternProtoDeclare>
571
572
573 <ExternProtoDeclare name="ik_orig" url="ik_proto.x3d">
574     <field name="texture_ik" type="MFString" accessType="inputOutput" />
575 </ExternProtoDeclare>
576
577
578 <ExternProtoDeclare name="jds_LOD" url="jds_proto_LOD.x3d">
579     <field name="texture_jds" type="MFString" accessType="inputOutput" />
580     <field name="farbe_dgm" type="SFColor" accessType="inputOutput" />
581 </ExternProtoDeclare>
582
583 <ExternProtoDeclare name="jds_coarse" url="jds_proto_coarse.x3d">
584     <field name="texture_jds" type="MFString" accessType="inputOutput" />
585 </ExternProtoDeclare>
586
587
588 <ExternProtoDeclare name="gw_LOD" url="gw_proto_LOD.x3d">
589     <field name="texture_gw" type="MFString" accessType="inputOutput" />
590 </ExternProtoDeclare>
591
592 <ExternProtoDeclare name="gw_coarse" url="gw_proto_coarse.x3d">
593     <field name="texture_gw" type="MFString" accessType="inputOutput" />
594 </ExternProtoDeclare>
595
596
597 <ExternProtoDeclare name="mio_LOD" url="mio_proto_LOD.x3d">
598     <field name="texture_mio" type="MFString" accessType="inputOutput" />
```

9

Hauptdatei

```

599 </ExternProtoDeclare>
600
601 <ExternProtoDeclare name="mio_coarse" url="mio_proto_coarse.x3d">
602   <field name="texture_mio" type="MFString" accessType="inputOutput" />
603 </ExternProtoDeclare>
604
605
606
607 <!-- PROTOTYPE Instanziierung Schichten externalLib_strata.x3d -->
608
609 <Transform DEF="weltall">
610 <Group>
611
612
613 <!-- PROTOTYPE dgm_orig -->
614
615   <Transform DEF="dgm" translation="0 0 0" scale='.01 .01 .01'>
616     <ProtoInstance DEF="proto_dgm" name="dgm_coarse" />
617     <PositionInterpolator DEF="up_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
618     <PositionInterpolator DEF="down_dgm" key="0, 0.4" keyValue="0 0 0, 0 0 0" />
619   </Transform>
620
621
622
623 <!-- PROTOTYPE wws_orig -->
624
625   <Transform DEF="wws" translation="0 0 0" scale='.01 .01 .01'>
626     <ProtoInstance DEF="proto_wws" name="wws_orig" />
627     <PositionInterpolator DEF="up_wws" key="0, 0.4" keyValue="0 0 0, 0 -2.5 0" />
628     <PositionInterpolator DEF="down_wws" key="0, 0.4" keyValue="0 -2.5 0, 0 0 0" />
629   </Transform>
630
631
632
633 <!-- PROTOTYPE ik_orig -->
634
635   <Transform DEF="ik" translation="0 0 0" scale='.01 .01 .01'>
636     <ProtoInstance DEF="proto_ik" name="ik_orig" />
637     <PositionInterpolator DEF="up_ik" key="0, 0.4" keyValue="0 0 0, 0 -4.5 0" />
638     <PositionInterpolator DEF="down_ik" key="0, 0.4" keyValue="0 -4.5 0, 0 0 0" />
639   </Transform>
640
641
642
643 <!-- PROTOTYPE jds_orig -->
644
645   <Transform DEF="jds" translation="0 0 0" scale='.01 .01 .01'>
646     <ProtoInstance DEF="proto_jds" name="jds_coarse" />
647     <PositionInterpolator DEF="up_jds" key="0, 0.4" keyValue="0 0 0, 0 -10 0" />
648     <PositionInterpolator DEF="down_jds" key="0, 0.4" keyValue="0 -10 0, 0 0 0" />
649   </Transform>
650
651
652
653 <!-- PROTOTYPE gw_orig -->
654
655   <Transform DEF="gw" scale='.01 .01 .01'>
656     <ProtoInstance DEF="proto_gw" name="gw_coarse" />
657     <PositionInterpolator DEF="up_gw" key="0, 0.4" keyValue="0 0 0, 0 -15 0" />
658     <PositionInterpolator DEF="down_gw" key="0, 0.4" keyValue="0 -15 0, 0 0 0" />
659   </Transform>
660
661
662
663
664 <!-- PROTOTYPE mio_orig -->

```

10

Hauptdatei

```

665
666     <Transform DEF="mio" scale='.01 .01 .01'>
667     <ProtoInstance DEF="proto_mio" name="mio_coarse" />
668     <PositionInterpolator DEF="up_mio" key="0, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
1.0" keyValue="0 0 0, 0 -20 0, 0 -19.8 0, 0 -20.2 0, 0 -19.85 0, 0 -20.1 0, 0 -19.95 0, 0
-20.03 0, 0 -20 0" />
669     <PositionInterpolator DEF="down_mio" key="0, 0.4" keyValue="0 -20 0, 0 0 0" />
670     </Transform>
671 </Group> <!-- strata -->
672
673 <!-- Inline nodes bcgp -->
674     <Transform DEF="bcgp" scale='.01 .01 .01'>
675     <Inline DEF="bohrungen" url="bohrungen.x3d" load="false" />
676     <Inline DEF="gruben" url="gruben.x3d" load="false" />
677     <Inline DEF="contour" url="contour.x3d" load="false" />
678     <Inline DEF="plan" url="vienna_osm.wrl" load="false" />
679     </Transform> <!-- bcgp -->
680
681 </Transform> <!-- Weltall -->
682
683
684     <Transform DEF="nm" scale='.01 .01 .01'>
685     <Inline DEF="nord" url="nord.x3d" load="true" />
686     <Inline DEF="massstab" url="massstab.x3d" load="true" />
687     </Transform> <!-- nm -->
688
689
690 <!-- bcgp -->
691 <ROUTE
692     fromNode="bohrungenSens" fromField="isActive"
693     toNode="jaNein_bohrungen" toField="set_boolean" />
694 <ROUTE
695     fromNode="jaNein_bohrungen" fromField="toggle"
696     toNode="bohrungen" toField="load" />
697
698 <ROUTE
699     fromNode="contourSens" fromField="isActive"
700     toNode="jaNein_contour" toField="set_boolean" />
701 <ROUTE
702     fromNode="jaNein_contour" fromField="toggle"
703     toNode="contour" toField="load" />
704
705 <ROUTE
706     fromNode="grubenSens" fromField="isActive"
707     toNode="jaNein_gruben" toField="set_boolean" />
708 <ROUTE
709     fromNode="jaNein_gruben" fromField="toggle"
710     toNode="gruben" toField="load" />
711
712 <ROUTE
713     fromNode="planSens" fromField="isActive"
714     toNode="jaNein_plan" toField="set_boolean" />
715 <ROUTE
716     fromNode="jaNein_plan" fromField="toggle"
717     toNode="plan" toField="load" />
718
719 <ROUTE
720     fromNode="nordSens" fromField="isActive"
721     toNode="jaNein_nord" toField="set_boolean" />
722 <ROUTE
723     fromNode="jaNein_nord" fromField="toggle"
724     toNode="nord" toField="load" />
725
726 <ROUTE
727     fromNode="massstabSens" fromField="isActive"
728     toNode="jaNein_massstab" toField="set_boolean" />
729 <ROUTE
730     fromNode="jaNein_massstab" fromField="toggle"
731     toNode="massstab" toField="load" />

```

11

Hauptdatei

```

732
733
734 <!-- explode -->
735 <ROUTE
736   fromNode="explode" fromField="touchTime"
737   toNode="Zeit_ex" toField="startTime" />
738
739 <ROUTE
740   fromNode="Zeit_ex" fromField="fraction_changed"
741   toNode="up_dgm" toField="set_fraction" />
742 <ROUTE
743   fromNode="up_dgm" fromField="value_changed"
744   toNode="dgm" toField="translation" />
745
746 <ROUTE
747   fromNode="Zeit_ex" fromField="fraction_changed"
748   toNode="up_jds" toField="set_fraction" />
749 <ROUTE
750   fromNode="up_jds" fromField="value_changed"
751   toNode="jds" toField="translation" />
752
753 <ROUTE
754   fromNode="Zeit_ex" fromField="fraction_changed"
755   toNode="up_gw" toField="set_fraction" />
756 <ROUTE
757   fromNode="up_gw" fromField="value_changed"
758   toNode="gw" toField="translation" />
759
760 <ROUTE
761   fromNode="Zeit_ex" fromField="fraction_changed"
762   toNode="up_mio" toField="set_fraction" />
763 <ROUTE
764   fromNode="up_mio" fromField="value_changed"
765   toNode="mio" toField="translation" />
766
767 <ROUTE
768   fromNode="Zeit_ex" fromField="fraction_changed"
769   toNode="up_ik" toField="set_fraction" />
770 <ROUTE
771   fromNode="up_ik" fromField="value_changed"
772   toNode="ik" toField="translation" />
773
774 <ROUTE
775   fromNode="Zeit_ex" fromField="fraction_changed"
776   toNode="up_wws" toField="set_fraction" />
777 <ROUTE
778   fromNode="up_wws" fromField="value_changed"
779   toNode="wws" toField="translation" />
780
781 <!-- implode -->
782 <ROUTE
783   fromNode="implode" fromField="touchTime"
784   toNode="Zeit_im" toField="startTime" />
785
786 <ROUTE
787   fromNode="Zeit_im" fromField="fraction_changed"
788   toNode="down_dgm" toField="set_fraction" />
789 <ROUTE
790   fromNode="down_dgm" fromField="value_changed"
791   toNode="dgm" toField="translation" />
792
793 <ROUTE
794   fromNode="Zeit_im" fromField="fraction_changed"
795   toNode="down_jds" toField="set_fraction" />
796 <ROUTE
797   fromNode="down_jds" fromField="value_changed"
798   toNode="jds" toField="translation" />
799
800 <ROUTE

```

12

```
Hauptdatei
801     fromNode="Zeit_im" fromField="fraction_changed"
802     toNode="down_gw" toField="set_fraction" />
803 <ROUTE
804     fromNode="down_gw" fromField="value_changed"
805     toNode="gw" toField="translation" />
806
807 <ROUTE
808     fromNode="Zeit_im" fromField="fraction_changed"
809     toNode="down_mio" toField="set_fraction" />
810 <ROUTE
811     fromNode="down_mio" fromField="value_changed"
812     toNode="mio" toField="translation" />
813
814 <ROUTE
815     fromNode="Zeit_im" fromField="fraction_changed"
816     toNode="down_ik" toField="set_fraction" />
817 <ROUTE
818     fromNode="down_ik" fromField="value_changed"
819     toNode="ik" toField="translation" />
820
821 <ROUTE
822     fromNode="Zeit_im" fromField="fraction_changed"
823     toNode="down_wws" toField="set_fraction" />
824 <ROUTE
825     fromNode="down_wws" fromField="value_changed"
826     toNode="wws" toField="translation" />
827
828
829
830 <!-- Überhöhung -->
831
832
833 <Script DEF="translation_to_scale">
834     <field accessType="inputOnly" name="from_translation" type="SFVec3f" />
835     <field accessType="outputOnly" name="to_scale" type="SFVec3f" />
836     <field accessType="outputOnly" name="new_transl_strata" type="SFVec3f" />
837     <field accessType="outputOnly" name="new_value" type="MFString" />
838
839     <![CDATA[ecmascript:
840         function from_translation(wert) {
841             to_scale = new SFVec3f(.01, wert.y+.01, .01);
842             new_transl_strata = new SFVec3f(0, 0-200*wert.y, 0);
843             ex_value = (wert.y*100+1);
844             round_value = Math.round(ex_value, 4);
845             new_value = new MFString(round_value);
846             if (new_value == '1') {
847                 new_value = new MFString('keine');
848             } else {
849                 new_value = new MFString(round_value);
850             }
851             print(to_scale);
852         }
853     ]]>
854 </Script>
855
856 <ROUTE
857     fromNode="sens_regler" fromField="translation_changed"
858     toNode="translation_to_scale" toField="from_translation" />
859
860 <ROUTE
861     fromNode="translation_to_scale" fromField="new_value"
862     toNode="value_exagg" toField="string" />
863
864 <ROUTE
865     fromNode="translation_to_scale" fromField="new_transl_strata"
866     toNode="weltall" toField="translation" />
867
868
869 <ROUTE
```

13

Hauptdatei

```

870     fromNode="translation_to_scale" fromField="to_scale"
871     toNode="dgm" toField="scale" />
872
873 <ROUTE
874     fromNode="translation_to_scale" fromField="to_scale"
875     toNode="jds" toField="scale" />
876
877 <ROUTE
878     fromNode="translation_to_scale" fromField="to_scale"
879     toNode="gw" toField="scale" />
880
881 <ROUTE
882     fromNode="translation_to_scale" fromField="to_scale"
883     toNode="mio" toField="scale" />
884
885 <ROUTE
886     fromNode="translation_to_scale" fromField="to_scale"
887     toNode="ik" toField="scale" />
888
889 <ROUTE
890     fromNode="translation_to_scale" fromField="to_scale"
891     toNode="wvs" toField="scale" />
892
893 <ROUTE
894     fromNode="translation_to_scale" fromField="to_scale"
895     toNode="bcgp" toField="scale" />
896
897
898 <!-- Profilschnitte -->
899
900 <ROUTE
901     fromNode="vorSens" fromField="isActive"
902     toNode="vorFilter" toField="set_boolean" />
903 <ROUTE
904     fromNode="vorFilter" fromField="inputTrue"
905     toNode="auswahlSeq" toField="next" />
906
907 <ROUTE
908     fromNode="zurueckSens" fromField="isActive"
909     toNode="zurueckFilter" toField="set_boolean" />
910 <ROUTE
911     fromNode="zurueckFilter" fromField="inputTrue"
912     toNode="auswahlSeq" toField="previous" />
913
914
915
916 <Script DEF="int_to_ps">
917     <field accessType="inputOnly" name="schnitt" type="SFInt32" />
918     <field accessType="outputOnly" name="dgm_texture" type="MFString" />
919     <field accessType="outputOnly" name="wvs_texture" type="MFString" />
920     <field accessType="outputOnly" name="ik_texture" type="MFString" />
921     <field accessType="outputOnly" name="jds_texture" type="MFString" />
922     <field accessType="outputOnly" name="gw_texture" type="MFString" />
923     <field accessType="outputOnly" name="mio_texture" type="MFString" />
924
925     <![CDATA[ecmascript:
926         function schnitt(wert) {
927             dgm_texture = new MFString('dgm_ps' + wert + '.png');
928             wvs_texture = new MFString('wvs_ps' + wert + '.png');
929             ik_texture = new MFString('ik_ps' + wert + '.png');
930             jds_texture = new MFString('jds_ps' + wert + '.png');
931             gw_texture = new MFString('gw_ps' + wert + '.png');
932             mio_texture = new MFString('mio_ps' + wert + '.png');
933         }
934     ]]>
935 </Script>
936
937
938 <ROUTE

```

14

Hauptdatei

```
939     fromNode="auswahlSeq" fromField="value_changed"  
940     toNode="int_to_ps" toField="schnitt" />  
941 <ROUTE  
942     fromNode="int_to_ps" fromField="dgm_texture"  
943     toNode="proto_dgm" toField="texture_dgm" />  
944 <ROUTE  
945     fromNode="int_to_ps" fromField="wvs_texture"  
946     toNode="proto_wvs" toField="texture_wvs" />  
947 <ROUTE  
948     fromNode="int_to_ps" fromField="ik_texture"  
949     toNode="proto_ik" toField="texture_ik" />  
950 <ROUTE  
951     fromNode="int_to_ps" fromField="jds_texture"  
952     toNode="proto_jds" toField="texture_jds" />  
953 <ROUTE  
954     fromNode="int_to_ps" fromField="gw_texture"  
955     toNode="proto_gw" toField="texture_gw" />  
956 <ROUTE  
957     fromNode="int_to_ps" fromField="mio_texture"  
958     toNode="proto_mio" toField="texture_mio" />  
959  
960 </Scene>  
961 </X3D>
```

15

Ich versichere:

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

Datum 24.11.2010

A handwritten signature in blue ink, appearing to read 'F. v. Quimont'.

Unterschrift

PERSÖNLICHE DATEN

*18.02.1978

Geboren in Wuppertal, Deutschland

HOCHSCHULSTUDIUM

Okt 2008-vorauss. Jan 2011

Master Kartographie und Geokommunikation

Frühwarnung & Risikomanagement, Multimedia, Programmieren, Fuzzy-Logic für GIS, Fernerkundung

Mai 2004

Magister Artium der RWTH Aachen

1998-2004

**Geographie Fernerkundung & GIS, Klimaentw.,
Naturraum Alpen**

Internationale Zusammenarbeit

Technische Zusammenarbeit, Klimawandel

Philosophie

Interkulturelle Philosophie, formale Logik

ARBEITSERFAHRUNG UND

PRAKTIKA

Feb-März 2010

United Nations (SPIDER)

Visual Globe, Space Application Matrix, Web-Knowledge Portal: Content Management u. CSS

Feb-März 2009

GeoExpert GmbH

Erstellung von Gefahrenzonenplänen, Erkundung der gefährdeten Gebiete, Auswertung von Luftbildern

seit Jan 2009

United Nations Information Service Vienna

Öffentlichkeitsarbeit

Apr 2008-Okt 2008

Greenpeace, Wien

Pflege, Erweiterung und Umbau der Internetpräsenz, Übersetzungen Englisch-Deutsch, Texte für Website

März 2007-Mai 2008

Comic/Graphic Novel

Zeichnungen und Texte, inspiriert von meinem Leben in New York, (asiatischer) Philosophie und Kampfkunst

Feb 2006-März 2007

**Department for Public Information, United Nations
New York, USA**

Präsentation internationaler Themen und Veranschaulichung der UN-Strukturen (auf engl.,

Dez 2005-Feb 2006	franz., deutsch), Fortbildungen in Präsentationstechniken und Kundenorientierung atmosfair gGmbH
Mai 2004	Organisation des Klima-Workshops, Erstellen der Adressdatenbank, Pflege der Kundenkontakte Vertretungsstelle bei Greenpeace, Hamburg
Sep 2003-Feb 2004	Recherche zum Energiemarkt, Aufbereitung von Informationen für die Öffentlichkeit Praktikum bei Greenpeace, Hamburg
Aug 2001	Assistent des Campaigners Klima/Öl, Internet Recherche Praktikum/Workcamp in Dakar, Senegal
Dez 2000-Feb 2001	Entwicklungsprojekt für Menschen mit Behinderungen GTZ-Praktikum in Saratov, Russland
	Artikel für interkulturelle Zeitung, Datenbankpflege

SCHULLAUFBAHN

1997	Abitur (Matura)
1994-1995	Toll Gate High School, RI, USA
1988-1997	Carl-Fuhlrott-Gymnasium, Wuppertal, Deutschland
	Schüleraustausch nach Frankreich
1984-1988	Grundschule Küllenhahn, Wuppertal, Deutschland

EDV-KENNTNISSE

Kartographie Esri ArcGIS, Mapserver
GIS/Fernerkundung Erdas Imagine
Graphik X3D, Adobe Illustrator, Photoshop, Director, Cinema4D
CMS Typo3, Drupal
Office MS Access, MS Office (alle), Lotus Notes, Mac OS X, Open Office

SPRACHEN

Englisch	fließend IELTS 8
Französisch	sehr gut Arbeitserfahrung
Japanisch	Fortgeschritten JLPT 3 (alt)
Russisch	Grundkenntnisse
Spanisch	Grundkenntnisse