

Anwendung immersiver Visualisierungssysteme zur  
Exploration geophysikalischer und geologischer  
Daten

**Dissertation**

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Björn Zehner

aus

Bremen

Bonn 2002



Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät  
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Referent: Prof. Dr. Hans-Joachim Kümpel
2. Referent: PD Dr. Andreas Hördt

Tag der Promotion: 25.04.2002



## Kurzfassung

Bei der dreidimensionalen Erkundung des geologischen Untergrundes mit Hilfe von geophysikalischen Methoden stellt sich die Frage, wie möglichst schnell ein Überblick über die Daten gewonnen werden kann. Wenn 3D-Modelle am Bildschirm dargestellt werden, geht in der Regel Information durch die Projektion auf die zweidimensionale Bildebene verloren und die Darstellung kann schnell unübersichtlich werden. Bei der Verwendung von immersiven Visualisierungssystemen wird, mit Hilfe von Stereo-Darstellung und Shutter-Brillen, ein echter dreidimensionaler Eindruck von den Datensätzen vermittelt. Die Abbildung erfolgt hierbei auf großen Projektionsflächen, so daß mehrere Benutzer das Bild gleichzeitig betrachten können.

Für die Produktentwicklung und für Designstudien, z.B. in der Autoindustrie, wird diese Technologie bereits seit längerem eingesetzt. Seit kurzem beginnt auch die Erdölindustrie sie zur Darstellung der komplexen und umfangreichen Reservoir-Datensätze zu verwenden, in der Regel um die Kommunikation innerhalb ihrer multidisziplinären Teams zu fördern. In dieser Arbeit wurde der Schwerpunkt auf weitere Anwendungsbereiche der Geologie und Geophysik gelegt, um zu klären, welche weiteren Möglichkeiten sich, z.B. in der Forschung und Lehre oder in der angewandten Geologie, ergeben können. Da für diese Anwendungsbereiche keine kommerzielle Software zur Verfügung stand, welche in der Lage war, immersive Visualisierungssysteme zu nutzen, wurden verschiedene Software-Prototypen für die Visualisierung von Bohrlochdaten, invertierten geoelektrischen Meßwerten, seismologischen Daten und zur Darstellung von Radargrammen und Reflektionsseismik entwickelt. Zusätzlich bestand die Möglichkeit, geometrische Modelle in die virtuelle Umgebung zu übernehmen.

Mit Hilfe der erstellten Software wurden eine 3D-Georadar-Vermessung, die geophysikalische Vermessung eines hydrogeologischen Versuchsfeldes, seismologische Daten eines hydraulischen Experimentes und das geometrische Modell eines Bergwerkes in virtuellen Umgebungen dargestellt. Der Benutzer kann sich durch das Modell navigieren und unmittelbar im Raum mit den Daten interagieren. Als Visualisierungssysteme kamen die Workbench und die CyberStage, eine CAVE-ähnliche Umgebung, zum Einsatz. Aufbauend auf den hierbei gewonnenen Erfahrungen werden die Vor- und Nachteile der Verwendung von Virtueller Realität und verschiedene mögliche Anwendungsgebiete bei der Interpretation geologischer und geophysikalischer Daten aufgezeigt.



# Abstract

When applying geophysical methods to investigate the geological subsurface, the question arises of how to obtain a good general view of the data. If 3-D data sets are displayed on a monitor, information is often lost because of the projection onto the 2-D plane, and the image quickly becomes confusing. By using immersive visualization systems and virtual reality, it is possible to get a real three-dimensional impression with help of stereo visualization and shutter glasses. When images are shown on large projection planes, different users can see it simultaneously.

For product development and design studies (e.g. in the automobile industry) such techniques are being applied for a couple of years. More recently, the oil industry uses this technology for the visualization of large and complex reservoir data; most often to enhance the communication within their asset teams. In this work the emphasis is on other branches within geology and geophysics to look for possible further fields of applications, such as research and training or issues in applied geology. For these fields of applications, no commercial software was available which could make use of immersive visualization systems. For this reason, software prototypes were implemented for the visualization of seismological data, well-logging data, inverted geoelectric data and surveys with ground penetrating radar (GPR) as well as 2-D seismics. In addition the possibility to load geometric models into a virtual environment is provided.

With the help of these software prototypes, a 3-D GPR survey, the geophysical investigation of a hydrogeological test site, seismological data from a hydraulic stimulation experiment and the geometric model of a silver mine were visualized within virtual environments. The Workbench and the CyberStage (a CAVE like system) were used as visualization systems. Based on the experiences gained during the implementation and the use of the software-prototypes, the advantages and disadvantages of the new technology and possible fields of applications are outlined.





# Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Medienkommunikation in Sankt Augustin (früher Teil der GMD-Forschungszentrum für Informationstechnik GmbH) und als Doktorand am Geologischen Institut der Friedrich-Wilhelms-Universität Bonn. Mein herzlicher Dank gilt Herrn Prof. Dr. Hans-Joachim Kümpel für die Betreuung der Arbeit und Herrn PD Dr. Andreas Hördt für die Übernahme des Korreferats. Dr. Christoph Ramshorn von der Firma Schlumberger machte mich während eines Praktikums in Austin/USA mit der Methode der Virtuellen Realität vertraut und stellte auch den Kontakt zum Institut für Medienkommunikation her. Ohne ihn hätte es diese Arbeit also wohl nicht gegeben.

Die Arbeit wurde im Rahmen des Doktorandenprogramms der GMD-Forschungszentrum für Informationstechnik GmbH finanziert. Hierfür sei meinen Vorgesetzten Dr. Martin Reiser und Dr. Martin Göbel gedankt. Am Institut für Medienkommunikation gilt mein Dank für die gute Unterstützung meinen Kollegen Dr. Florian Dombois, Dr. Bernd Fröhlich, Jürgen Wind, Jan Springer, Andreas Simon, Johannes Strassner und den weiteren Mitgliedern der Arbeitsgruppe "Virtual Environments" sowie dem wieder nach "Down Under" zurückgekehrten Dr. Stephen Barrass, der mir gerade am Anfang meiner Arbeit besonders hilfreich war.

Die Durchführung dieser Dissertation war nur mit geeignetem Datenmaterial möglich. In diesem Sinne danke ich Herrn Dr. Vereecken und Herrn Hashagen vom Forschungszentrum Jülich sowie Katja C. Schulze von der Universität Bonn und Matthias Seuffert, daß sie Daten des hydrogeologischen Versuchsfeldes Krauthausen zur Verfügung gestellt haben. Herr Dr. Jung und Herr Dr. Weidler von der Bundesanstalt für Geowissenschaften und Rohstoffe (BGR) in Hannover sowie Herr Dr. Hinzen und Frau Dr. Reamer von der Universität Köln stellten die seismologischen Daten eines hydraulischen Stimulationstests zur Verfügung. Carsten Alteköster von der Universität Bonn danke ich für die Bereitstellung von Bodenradar-Daten. Stefan Ganter aus Freiburg sendete mir freundlicherweise seine Diplomarbeit über das digitale Modell des Schauinsland-Bergwerkes zu. Ferner ist Andreas Thomsen vom Geologischen Institut der Universität Bonn zu erwähnen, mit dem ich zahlreiche interessante Diskussionen führen konnte. Meiner Mutter, Alexa Zehner, danke ich für das Korrekturlesen des Manuskripts.

Für die Durchführung dieser Arbeit war es sehr nützlich, wenn nicht gar unerlässlich, daß ich bereits während des Geologiestudiums in Freiburg die Möglichkeit hatte Erfahrung im Bereich Programmierung und Visualisierung zu sammeln. Daher möchte ich diese Gelegenheit nutzen, um auch einer Reihe von "Freiburgern" meinen Dank auszusprechen. Dies sind zum einen die "Computergeologen", Prof. Dr. Reinhard Pflug und die Mitglieder seiner Arbeitsgruppe, zum anderen die "Strukturgeologen" Prof. Dr. Jan Behrmann und Dr. David Tanner und natürlich meine damaligen Kommilitonen. Ich habe die offene Atmosphäre am Geologischen Institut in Freiburg immer sehr zu schätzen gewußt.

Die vorliegende Arbeit ist in digitaler Form über den Hochschulschriftenserver der Universitäts- und Landesbibliothek Bonn erhältlich ("[http://hss.ulb.uni-bonn.de:90/ulb\\_bonn/diss\\_online](http://hss.ulb.uni-bonn.de:90/ulb_bonn/diss_online)").

Bonn, im Februar 2002

Björn Zehner

Teile aus dieser Arbeit wurden bereits in folgenden Beiträgen vorab veröffentlicht:

#### Publikationen

Barrass, S., Zehner, B. (2000): Responsive Sonification of Well-Logs, Proceedings of the Sixth International Conference on Auditory Display, Atlanta, Georgia

Fröhlich, B., Barrass, S., Zehner, B., Plate, J., Göbel, M. (1999): Exploring GeoScience Data in Virtual Environments, Proceedings of IEEE Conference on Visualization, San Francisco, 1999

Zehner, B., Barrass, S., Plate, J., Göbel, M., Fröhlich, B. (1999): Interaktive Visualisierung und Validierung von Geophysikalischen Daten in Virtuellen Umgebungen, In: Schmidt, B., Uhlenkücken, C. (Hrsg.), Visualisierung raumbezogener Daten: Methoden und Anwendungen, IFGI-prints 6, Verlag Natur & Wissenschaft, Solingen, S. 59-72

Zehner, B., Kümpel, H.-J. (2001): Immersive Visualisierungstechniken bei der Geophysikalischen Erkundung des Untergrundes, Kartographische Nachrichten, 51/4 (2001), Kirschbaum Verlag, Bonn, S. 196-200

Zehner, B., Alteköster, C., Kümpel, H.-J. (2001): Visualization of 3-D GPR Surveys: Application of Virtual Reality in Geosciences, European Journal of Environmental and Engineering Geophysics, Vol. 6, Geophysical Press, Montferrier-sur-Lez, France, S. 141-152

#### Tagungsbeiträge

Vortrag: Zehner, B. (1999): "Anwendung virtueller Umgebungen bei der Exploration geologischer und geophysikalischer Daten", 32. Herbsttagung des Arbeitskreises Geodäsie/Geophysik, Bucha-Hohenwarte bei Saalfeld, Nov. 1999

Poster: Zehner, B., Kümpel, H.-J. (2000): "Geophysik im Cyberspace ?", Tagung der Deutschen Geophysikalischen Gesellschaft in München, Februar 2000

Poster: Zehner, B., Alteköster, C., Kümpel, H.-J. (2001): "Visualization of 3D GPR Surveys: Application of Virtual Reality in Geosciences", Tagung der European Geophysical Society in Nizza, April 2000



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Hardwarebetrachtungen</b>	<b>5</b>
2.1	Einführung . . . . .	5
2.2	Die CyberStage . . . . .	10
2.3	Die Workbench . . . . .	12
2.4	Verwendete Rechnersysteme . . . . .	13
2.5	Arbeiten im Bereich Low-Cost VR . . . . .	15
<b>3</b>	<b>Softwarebetrachtungen</b>	<b>19</b>
3.1	Anforderungen an die verwendete Software . . . . .	19
3.2	Die Rahmenapplikation AVANGO . . . . .	23
<b>4</b>	<b>Allgemeine Anwendungsbetrachtungen</b>	<b>29</b>
4.1	Vor- und Nachteile der Anwendung . . . . .	29
4.2	Anwendungsszenarios . . . . .	36
<b>5</b>	<b>Stand der Forschung</b>	<b>39</b>
<b>6</b>	<b>Beispielanwendungen</b>	<b>45</b>
6.1	Das VRGeo-Projekt . . . . .	47
6.1.1	Übersicht und verwendete Daten . . . . .	47
6.1.2	Visualisierung der verschiedenen Datentypen . . . . .	48
6.1.3	Erfahrungen mit dem Software-Prototypen . . . . .	52
6.2	Visualisierung von Georadardaten . . . . .	55
6.2.1	3D-Erkundung des Untergrundes mit Bodenradar . . . . .	55
6.2.2	Eine Bodenradar-Vermessung als interaktives virtuelles Modell	57
6.2.3	Erfahrung mit der Verwendung des Systems . . . . .	62
6.3	Hydrogeologisches Testfeld Krauthausen . . . . .	67
6.3.1	Das Testfeld und die verwendeten Daten . . . . .	67

6.3.2	Visualisierung der verschiedenen Datentypen . . . . .	72
6.3.3	Gesamtmodell und Interaktion . . . . .	76
6.3.4	Erfahrung beim Arbeiten mit dem System . . . . .	78
6.4	4D-Darstellung seismologischer Daten . . . . .	81
6.4.1	Geothermie in Soultz-sous-Forêts . . . . .	81
6.4.2	Seismologie zur Beobachtung hydraulischer Tests . . . . .	82
6.4.3	Verwendung von VR zur Betrachtung der Messungen . . . . .	83
6.4.4	Erfahrung mit der Verwendung des Modells . . . . .	86
6.5	Visualisierung eines Silberbergwerkes . . . . .	87
6.5.1	Allgemeines . . . . .	87
6.5.2	Der Beispiel-Datensatz Schauinsland-Bergwerk . . . . .	87
6.5.3	Zum Thema verteiltes Arbeiten . . . . .	88
6.5.4	Erfahrung mit dem Modell . . . . .	91
<b>7</b>	<b>Diskussion</b>	<b>93</b>
7.1	Möglichkeiten der Anwendung . . . . .	93
7.1.1	Visuelle Datenanalyse . . . . .	96
7.1.2	Geometrische Modellierung . . . . .	97
7.1.3	Computational Steering . . . . .	101
7.1.4	3D-Informationssysteme . . . . .	102
7.1.5	Einsatz für Ausbildung und Training . . . . .	102
7.2	Resumee . . . . .	104
	<b>Abbildungsverzeichnis</b>	<b>107</b>
	<b>Literatur</b>	<b>111</b>
	<b>Liste der Abkürzungen</b>	<b>121</b>
	<b>Anhang Implementierung</b>	<b>123</b>

# Kapitel 1

## Einleitung

Geowissenschaftliche Fragestellungen sind von Natur aus oft dreidimensional. Da zu ihrer bildlichen Darstellung lange Zeit keine geeigneten Methoden existierten, mußte die Visualisierung in 2D erfolgen. Beispiele hierfür sind seismische Schnitte oder die Darstellung von strukturgeologischen Zeichen auf topographischen Karten. Die Erstellung des 3D-Modells vom Untergrund erfolgte rein mental. In jüngerer Zeit wird die Modellbildung unterstützt durch Methoden der geometrischen Modellierung und der 3D-Computergraphik. Die Projektion der berechneten Bilder auf die 2D-Ebene des Bildschirms setzt hier jedoch Grenzen bei der Anwendung, was die geometrische Komplexität und die Menge an Daten betrifft. Auch die Interaktion ist häufig zu kompliziert oder zu langsam. Die in jüngerer Zeit entwickelten immersiven Visualisierungssysteme stellen mit der Methode der Virtuellen Realität (VR) die Technik zur Verfügung, um die Darstellung der Daten und die Diskussion über mögliche Untergrundstrukturen im dreidimensionalen Raum durchführen zu können.

In der Industrie wird VR seit einigen Jahren zunehmend angewendet, um Designfragen zu klären. In Deutschland ist die Autoindustrie auf diesem Gebiet führend. Die möglichst realistische Darstellung der CAD-Modelle in Lebensgröße soll die Erstellung von "echten" Prototypen weitgehend ersetzen und so die Produktzyklen verkürzen. Innerhalb des geowissenschaftlichen Bereichs übernehmen eindeutig die Unternehmen der Erdölindustrie die Rolle der Vorreiter, während in Forschung, Lehre und in Bereichen der angewandten Geowissenschaften wie der Ingenieurgeologie und Hydrogeologie bisher nur geringe Aktivitäten zu verzeichnen sind. Dies dürfte zum einen an den enormen Kosten liegen, die zur Zeit noch mit VR verbunden sind und die nur von solchen großen Firmen getragen werden können. Zum ande-

ren haben Erdölfirmer sehr viel größere Datensätze zu verarbeiten als sie in der akademischen Forschung oder in geologischen Ingenieurbüros üblich sind. Dennoch ist anzunehmen, daß sich auch in anderen Bereichen der Geologie und Geophysik sinnvolle Anwendungen für diese Technologie finden lassen.

Parallel zu dieser Arbeit fand bezüglich der Verwendung von VR bei geophysikalischen Untersuchungen eine sehr schnelle Entwicklung auf dem Erdölsektor statt. Waren vor 1998 mit Hilfe der GeoRef-Datenbank kaum Publikationen zum Thema VR in der Erdölindustrie oder den Geowissenschaften zu finden, kann man ab ca. 1998/1999 einen regelrechten Boom aus dieser Richtung verzeichnen. Ältere Veröffentlichungen aus dem geologischen Bereich, die mit Hilfe der Datenbank gefunden wurden, verwendeten den Begriff bereits bei der Benutzung der Virtual Reality Modelling Language (VRML). Hierbei handelt es sich um 3D-Visualisierung im klassischen Sinne und nicht um die Arbeit mit immersiven Visualisierungstechniken, wie sie Gegenstand dieser Arbeit sind.

Die häufigste Anwendung in der Erdölindustrie ist die Darstellung von 3D-Computermodellen des Untergrundes auf großflächigen zylindrischen Projektionsdisplays (sogenannten Visionarien), welche es ihren Asset-Teams ermöglichen sollen, über die Daten zu diskutieren. Die Darstellung ist allerdings häufig nicht in Stereo, und die Steuerung erfolgt extern von einer Konsole aus. Es handelt sich also nur bedingt um VR im eigentlichen Sinne. Sehr selten werden solche virtuellen Umgebungen auch unmittelbar für die Interpretation und Modellierung des Untergrundes eingesetzt. Für diesen Fall werden erhebliche Effizienzsteigerungen angegeben. Meist wird hierbei allerdings die Interpretation auf Standard-Workstations verglichen mit der "neuen" Methode, ausgeführt auf High-End Graphikrechnern mit Multiprocessing. Um den Nutzen dieser Methode wirklich beurteilen zu können, wäre hier eine weitergehende Differenzierung notwendig: Welche Vorteile bringt die Verwendung schneller paralleler Algorithmen? Welche Vorteile lassen sich durch die Kopplung dieser mit High-End 3D-Computergraphik erreichen? Und welche Vorteile fügt die Verwendung von VR hinzu? Diese Differenzierung ist insofern wichtig, da sie parallel läuft mit einer Abstufung der notwendigen Kosten und damit der Verfügbarkeit für einen möglichst breiten Teil der Geowissenschaften. Die Verwendung von Parallel-Processing ist bereits kostengünstig auf Linux-Clustern mit Hilfe des Message-Passing-Interfaces zu haben. Die Kosten steigen sehr viel schneller, wenn man besonders schnelle Graphik und Stereo-Darstellung benötigt. Die Verwendung von immersiven Visualisierungsumgebungen erfordert in der Regel die Kopplung und Synchronisation mehrerer Graphik-Systeme, was die Verwendung von



speziellen Graphik-Workstations notwendig macht und die Kosten auf ein Vielfaches steigert.

Die vorliegende Arbeit beschäftigt sich hauptsächlich mit der Frage, welche Vorteile die Verwendung Virtueller Realität für die Geowissenschaften bietet und welche potentiellen Anwendungen, insbesondere außerhalb der Erdölindustrie, sich eventuell ergeben. Um die hierfür notwendigen Kenntnisse und Erfahrungen zu sammeln, wurden verschiedene Anwendungen entwickelt und Beispiel-Datensätze aus dem Bereich Geologie/Geophysik visualisiert. Diese Beispiele können in Form von simplen Anwendungen zur Demonstration verwendet werden und dienen gleichzeitig dazu, die Grenzen der Methode anhand von Beispielen aufzuzeigen. Ein weiteres Anliegen dieser Arbeit ist der Wissenschaftstransfer aus der Informatik in die Geowissenschaften. Der vorliegende Text soll eine Übersicht über die Verwendung von immersiven Visualisierungstechniken in den Geowissenschaften bereitstellen sowie die notwendigen Informationen, welche benötigt werden, um für neue Probleme in der Geologie und Geophysik zu beurteilen, ob eine Anwendung dieser Techniken jeweils sinnvoll ist oder nicht. Am Fraunhofer Institut für Medienkommunikation konnte für diese Arbeit auf eine gut ausgebaute Infrastruktur in Verbindung mit umfangreicher Erfahrung auf dem Gebiet der VR zugegriffen werden. Hinzu kam, daß kurz vor Beginn dieser Arbeit ein Konsortium von Erdölfirmen, der damaligen GMD (jetzt Teil der Fraunhofer Gesellschaft) und des Virtual Environment Technology Laboratories der University of Houston gegründet worden ist, das zum Ziel hatte, eine Demonstrationssoftware für eine mögliche VR-Anwendung in der Erdölindustrie, insbesondere mit Hinblick auf Bohrlochplanung, zu entwickeln. Die Möglichkeit, als Geowissenschaftler an der Forschung auf dem Gebiet VR teilzunehmen, ergab sich erst hierdurch.

Während der Durchführung der Arbeit ergaben sich einige Randbedingungen, mit denen im voraus nicht gerechnet werden konnte. So ist die Implementierung von interaktiven Anwendungen aufwendiger als erwartet und daher in der Regel nur in größeren Teams zu bewältigen. Aus diesem Grunde haben die erstellten Beispiele immer nur exemplarischen Charakter und stellen bei weitem keine vollwertigen Anwendungen dar. Ferner zeigte sich, daß die Entwicklung einer auf einem realen Datensatz beruhenden Beispielanwendung, die das Potential von VR für eine geowissenschaftliche Fragestellung optimal aufzeigen und nutzen kann, Ähnlichkeit zum klassischen "Ei-Henne Problem" aufweist. Da den Geologen und Geophysikern außerhalb von Erdölfirmen bisher meist kein Verfahren zur Verfügung steht, Daten mit echt dreidimensionalem Eindruck darzustellen und in Echtzeit mit ihnen zu in-

teragieren, wurden Probleme meist so angegangen, daß dies auch nicht notwendig war, und dabei oft vereinfacht. Dementsprechend waren auch keine Datensätze zu bekommen, mit deren Hilfe man das Potential von VR hätte optimal ausschöpfen und aufzeigen können. Dieser Umstand zeigt aber nicht, daß die Anwendung von VR in dieser Anwendungsdomäne keinen Sinn macht und keine Vorteile bringt. Unter Umständen würde die Verwendung von VR überhaupt erst die Voraussetzung schaffen, die eine oder andere Untersuchung durchzuführen oder an sie mit neuen Verfahrensweisen heranzugehen. Die visuelle Analyse hat auf jeden Fall den Vorteil, daß sie weitgehend unabhängig von der Heterogenität der Daten ist und sich für neue Datenformate meist schneller realisieren läßt als zum Beispiel die Erstellung neuer Programme für das Prozessieren der Daten.

Diese Arbeit gliedert sich in weitere sechs Kapitel. In Kapitel 2, "Hardwarebetrachtungen", soll zunächst eine allgemeine Einführung in die für die Anwendung von VR notwendige Technik gegeben und die für diese Arbeit verwendeten Visualisierungsumgebungen und die Computertechnik vorgestellt werden. Da am Fraunhofer Institut für Medienkommunikation nur sehr teure, hochqualitative Hardware zur Verfügung stand, sollen hier auch noch einige Arbeiten vorgestellt werden, die sich mit der Entwicklung von wesentlich günstigeren Systemen beschäftigen. Die Verwendung von VR stellt besondere Anforderungen an die Software, denen kommerzielle Programme in der Regel nicht genügen. Daher mußte diese selbst entwickelt werden. Das Kapitel "Software-Betrachtungen" beschäftigt sich mit diesen Anforderungen und stellt die Umgebung vor, in welcher die in dieser Arbeit verwendete Software implementiert worden ist. Das darauf folgende Kapitel "Allgemeine Anwendungsbetrachtungen" soll aufzeigen, weshalb die Verwendung von VR für die geologische Interpretation geophysikalischer Daten eventuell sinnvoll sein kann, und erläutert, was die Vor- und Nachteile dieser Technologie sind. Ferner werden im Abschnitt 4.2 die verschiedenen Anwendungsszenarien erläutert. Kapitel 5 beschäftigt sich mit dem momentanen Stand der Entwicklung auf dem Gebiet der Geowissenschaften. Nach dem Kapitel 6, welches die verschiedenen Beispielanwendungen beschreibt, werden im Kapitel 7 "Diskussion" einige Arbeitsbereiche innerhalb der Geologie und Geophysik aufgezeigt, für welche die Anwendung von VR sinnvoll erscheint. Ferner werden für andere Bereiche die Voraussetzungen erläutert, die geschaffen werden müssen, um auch dort diese Methode anzuwenden.

# Kapitel 2

## Hardwarebetrachtungen

### 2.1 Einführung

Die Verwendung von Virtual Reality (VR) soll es dem Benutzer ermöglichen, in eine computerberechnete Umgebung einzutauchen und direkt mit den Objekten im Raum zu interagieren. Das Gefühl, sich in einer anderen (digitalen) Welt zu befinden, wird auch als Immersion bezeichnet. Ein System ist um so immersiver, je weniger der Benutzer die reale Umwelt bei der Betrachtung der virtuellen Welt wahrnimmt und von dieser abgelenkt wird. Sehr wichtig ist hierbei der Blickwinkel des Benutzers, der vom computergenerierten Bild eingenommen und als Field Of View (FOV) bezeichnet wird. Diese Arbeit beschränkt sich hauptsächlich auf den visuellen Teil, welcher zusätzlich um einen akustischen und einen haptischen (kraftrückgekoppelten) Teil erweitert werden kann. Im Sinne der Visualisierung stellt VR eine Erweiterung der 3D-Computergraphik um mindestens zwei Eigenschaften dar, die Stereo-Visualisierung und das Head-Tracking. Durch stereoskopische Darstellung (getrennte Darstellung für rechtes und linkes Auge) wird über die Konvergenz der Augen ein räumlicher Eindruck erzeugt. Der Computer verfolgt die Position und die Blickrichtung des Benutzers (Head-Tracking) und berechnet jeweils die Perspektive korrekt für den Benutzer. Auf diese Weise kann der Benutzer sich im Idealfall frei in der virtuellen Welt bewegen.

Um eine getrennte Darstellung für beide Augen zu erreichen, gibt es mehrere Möglichkeiten. Die wohl bekannteste ist die Verwendung von Head-Mounted-Displays (HMD's), bei denen in einem Helm zwei kleine LCD-Bildschirme unter-

gebracht sind, welche die getrennte Darstellung für die beiden Augen übernehmen. Diese Systeme haben jedoch den Nachteil, daß sie erstens den Benutzer visuell von seiner Umwelt isolieren (und so kooperatives Arbeiten erschweren), zweitens meistens nur eine relativ geringe Bildauflösung haben und drittens in der Regel nur einen relativ beschränkten Blickwinkel zulassen. Ist der Blickwinkel zu gering, muß der Benutzer, statt mit den Augen zur Seite zu sehen, den ganzen Kopf drehen. Ein Überblick über die verschiedenen am Markt erhältlichen HMD's findet sich in Real Time Graphics (2000).

Bei den in dieser Arbeit verwendeten immersiven Visualisierungssystemen wird das Bild mit CRT-Projektoren für beide Augen auf großflächige Projektionswände geworfen und mit Hilfe von speziellen Brillen getrennt. Bei der als "Passives Stereo" bezeichneten Variante wird das Licht der Projektoren orthogonal zueinander polarisiert und der Betrachter trägt eine Brille mit zwei orthogonal zueinander polarisierten Gläsern, so daß nur die dem jeweiligen Auge entsprechenden Lichtanteile durchgelassen werden. Diese Technologie hat allerdings den Nachteil, daß die Bilder oft, z.B. beim Schrägstellen des Kopfes, nicht sauber getrennt werden und jedes Auge daher auch zu einem geringen Teil das für das andere Auge bestimmte Bild sieht. Dieser Fehler wird als "ghosting" bezeichnet. Aus diesem Grunde werden Passiv-Stereo-Systeme meist nur mit einfachen einseitigen Projektionswänden verwendet. Die am Institut für Medienkommunikation vorhandenen Systeme verwenden alle "Aktives Stereo". Die Bilder für die beiden Augen werden immer abwechselnd projiziert und der Benutzer trägt eine Shutterbrille, welche mit dem Computer/Projektor synchronisiert ist und jeweils entsprechend die Sicht für die Augen öffnet und schließt. Diese Methode liefert die qualitativ hochwertigsten Bilder, erfordert aber spezielle und daher teure CRT-Projektoren, welche in der Lage sind, eine Bildwiederholrate von 120 Hz zu liefern ohne nachzuleuchten, und bei denen sich das Videosignal synchronisieren läßt. Von der Firma TAN Projektionstechnologie GmbH & Co. KG wurde in Zusammenarbeit mit DaimlerChrysler eine neue Filtertechnologie entwickelt, und zwar das Infitec-Stereo-Viewing. Diese Technik ist günstiger als die Shutter-Systeme, da die Auftrennung der beiden Bilder für die Augen von der Blickrichtung laut Herstellerangaben nicht beeinflußt wird und die Technik auch in Verbindung mit helleren DLP-Projektoren (Digital) verwendet werden kann. Bei der Demonstration eines Prototyps auf dem Benutzertreffen der Firma Advanced Visual Systems im Februar 2001 in Düsseldorf zeigte sich allerdings schwaches "ghosting" in Abhängigkeit von der Blickrichtung. Außerdem wiesen die Gläser an der Rückseite sehr starke Spiegelungen auf, was als störend empfunden wurde.

Die Projektion der Bilder erfolgt nach Möglichkeit von der Rückseite aus, so daß die Betrachter keine störenden Schatten werfen. Dies ist jedoch nicht immer möglich. Die Klassifizierung der verschiedenen virtuellen Umgebungen erfolgt maßgeblich anhand der Anzahl und der Anordnung der Projektionsflächen. Einen Überblick über die verschiedenen Projektionssysteme, ihre Vor und Nachteile und die ungefähren Kosten, die mit ihrer Verwendung verbunden sind, wird in Wright (2000) gegeben. Bei den Kosten ist zu beachten, daß die Kosten für die Projektionssysteme im Vergleich zu den Kosten der für VR benötigten Rechner nur langsam fallen. Wright (2000) unterteilt die Projektionssysteme in Workbench-Systeme, Walls mit planarer Projektionsfläche, Walls mit zylindrischer Projektionsfläche, Domes mit konkaver Projektionsfläche und "Spatially Immersive Rooms". Zusätzlich sollte man allerdings bei den Workbenches noch unterscheiden zwischen denen mit einer und denen mit zwei Projektionsflächen, da sich bei letzteren wesentliche bessere Interaktionsmöglichkeiten ergeben. Da bei der Entwicklung meist der Markenname geschützt wird, gibt es eine Fülle von Systemen, deren Aufbau sehr ähnlich ist, welche aber unter unterschiedlichen Namen geführt werden. Abbildung 2.1 soll hier einen kleinen, mit Sicherheit nicht vollständigen Überblick für die Visualisierungssysteme verschaffen, die den in dieser Untersuchung verwendeten ähnlich sind.

Wichtig in dieser Arbeit ist der Begriff des Interaktionsraumes. Je nach Konfiguration der Displays hat man vor den Projektionsflächen ein mehr oder weniger großes Volumen in dem sich ein virtuelles Modell frei schwebend im Raum darstellen läßt, so daß man direkt mit der Hand in das Modell hereinreichen kann. Dieses Volumen wird als Interaktionsraum bezeichnet.

Als Trackingsystem kommt am Institut für Medienkommunikation ein Polhemus 3Space-Fastrak-System zum Einsatz. Dieses liefert 6 Freiheitsgrade (6 Degrees Of Freedom = 6DOF), das heißt sowohl die Position des Kopfes als auch seine Orientierung. Da dieses System elektromagnetische Wellen benutzt, wird es durch Metall gestört, so daß die virtuellen Umgebungen aus unmagnetischem Material konstruiert werden müssen.

Für die Interaktion in einer virtuellen Umgebung benötigt man spezielle Eingabegeräte (siehe Abbildung 2.2), da weder eine Tastatur noch eine Maus zur Verfügung stehen. Am Institut für Medienkommunikation am häufigsten eingesetzt wird der Polhemus Stylus, ein Zeigestift mit einem Knopf, der mit einem Polhemus Fasttrack-6DOF-Tracker ausgestattet ist, so daß Position und Orientierung im Raum bekannt sind. Dieser wird in der Regel mit einem Zeigestrahl ausgestattet und kann dann wie

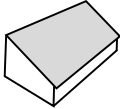
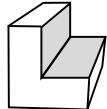
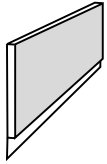

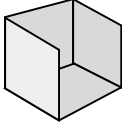
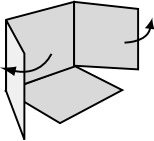
	Eine Projektionsfläche mit unterschiedlichem Anstellwinkel	Responsive Workbench (Trademark GMD, Vertrieb TAN) BARON (BARCO) MD Table (MechDyne) ImmersaDesk (Trademark University of Illinois, Vertrieb Fakespace) V-Desk 5 (Trimension)
	Zwei Projektionsflächen mit dazwischen liegendem Interaktionsraum	Holobench (TAN) CONSUL (BARCO) V-Desk 6 (Trimension)
	Eine oder mehrere Projektionsflächen nebeneinander. Meist Rückprojektion	TANORAMA Powerwall (TAN) CAD-Wall (BARCO) MD-Wall (MechDyne) Immersive Workwall (Fakespace) ReaCTor Wall (Trimension) V-Desk 8 (Trimension)
	Mehrere Abbildungen auf zylindrische Projektionsfläche. Meist Front-Projektion. Allgemeine Bezeichnung: Visionarium.	TANORAMA Zylinder (TAN) BR CENTER (BARCO) MD CSVR (MechDyne) R2 - Family (Trimension Systems) GVR-120 VisCenter (Panoram Technologies) MD PLEX (MechDyne), drei planare Projektionen
	4-6 Projektionen auf die Außenseiten des Raumes. Allgemeine Bezeichnung: CAVE	VR-CUBE (TAN) Cyberstage (GMD) SSVR (MechDyne) CAVE (University of Illinois, Vertrieb durch Fakespace) ReaCTor (Trimension Systems)
	Durch Ausklappen der beiden Seitenwände kann der CAVE-artige Raum in ein Visionarium umgewandelt werden.	MD-FLEX (MechDyne) Rave (Fakespace)
Quellen:	<a href="http://www.tan.de">http://www.tan.de</a> <a href="http://www.barco.com/projection_systems/index.asp?topic=product">http://www.barco.com/projection_systems/index.asp?topic=product</a> <a href="http://www.mechdyne.com">http://www.mechdyne.com</a> <a href="http://www.trimension-inc.com">http://www.trimension-inc.com</a> <a href="http://www.panoramtech.com">http://www.panoramtech.com</a>	

Abbildung 2.1: Übersicht über die verschiedenen auf dem Markt erhältlichen Projektionssysteme mit ihren Trademarks und den jeweiligen Herstellern.

ein Laserpointer eingesetzt werden, um z.B. auf Objekte im Raum zu zeigen oder diese zu transferieren und zu drehen. Das “3-Button-Tool“ wird wie der Stylus verwendet, hat aber zusätzlich drei Knöpfe, die mit Funktionen belegt werden können. Ein anderes Gerät ist die speziell für die Interaktion mit Datenvolumen entwickelte Cubic Mouse (Fröhlich & Plate, 2000). Wie in der Abbildung 2.2 zu sehen ist, besteht diese aus einer würfelförmigen Box mit drei beweglichen Stäben durch das Zentrum, welche mit einem 6DOF-Tracker und mehreren Knöpfen ausgestattet ist. Die Orientierung der Cubic-Mouse ist hierbei an die Orientierung der meist quaderförmigen Datenvolumen der 3D-Seismik und der Computertomographie gekoppelt. Mit den drei Stäben läßt sich die jeweils entsprechende Abbildungsebene durch das Datenvolumen bewegen. Da der Benutzer sozusagen die volumetrischen Daten in seiner Hand hat, ist die Interaktion auch für Computernovizen sehr intuitiv.

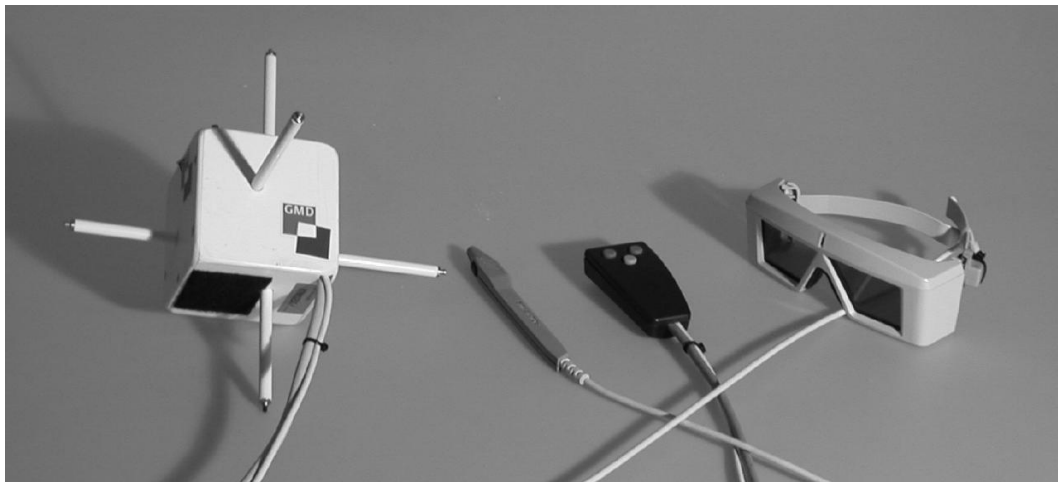


Abbildung 2.2: In dieser Arbeit verwendete Interaktionsgeräte und die notwendige Spezial-Brille zum Trennen der Bilder für rechtes und linkes Auge (von links nach rechts: Cubic-Mouse, Polhemus-Stylus, 3-Button-Tool, Shutterbrille).

Viele Interaktionen lassen sich mit einer klassischen 2D-Benutzerschnittstelle besser vornehmen als im 3D-Raum. Als Beispiel sei hier die Eingabe von Dateinamen oder das Editieren einer Farbtabelle genannt. Das Nachbilden von 2D-Interaktionstechniken im Raum stellt hierfür keine befriedigende Lösung dar. Umfangreiche Menüs sind kompliziert zu bedienen und schränken den Blick des Benutzers auf den eigentlichen Untersuchungsgegenstand ein. Am Fraunhofer Institut für Medienkommunikation wurde aus diesem Grunde ein Laptop mit Touchscreen in die virtuellen Umgebung integriert. Eingaben und Ereignisse der hierauf laufenden Benutzerschnittstelle werden über CORBA (Common Object Request Broker Ar-

chitecture, Schnittstelle zur Entwicklung verteilter Applikationen) an die eigentliche Anwendung kommuniziert. In Watsen et al. (1999) wird die Verwendung eines 3Com PalmPilot-Handheld-Computers für denselben Zweck beschrieben.

Bei der Interaktion wurde es oft als störend empfunden, daß die Eingabegeräte mit Hilfe von Kabeln mit der auswertenden Hardware verbunden sind. Auch der Sensor für das Head-Tracking, welcher sich an der Brille befindet, ist über Kabel mit der entsprechenden Auswertebox verbunden. Die Kabel werden als um so störender empfunden, je mehr Bewegungsfreiheit der Benutzer innerhalb der virtuellen Umgebung hat, da hierdurch die Kabel immer leichter durcheinander geraten. In Häfner et al. (2000a) wird die Verwendung eines kabellosen Interaktionssystems beschrieben. Das Tracking, das hier nur die Position erfaßt (3 Freiheitsgrade = 3DOF), erfolgt optisch mit Hilfe mehrerer Trackingsysteme, deren Meßräume sich einander überlagern. Auf diese Weise konnte ein relativ großes Arbeitsvolumen abgedeckt werden. Als Projektionssystem kam eine 6,4 m x 1,96 m große Wall zum Einsatz. Als Beispielanwendung wurde die Evaluierung des Designs eines Autos gewählt. Die Erfahrung mit diesem kostengünstigen System war sehr gut. Laut Häfner et al. (2000a) scheint die Kabelfreiheit der Interaktionsgeräte wichtiger zu sein als das 6DOF-Tracking, welches zum Beispiel am Institut für Medienkommunikation für diese Arbeit verwendet wurde.

## 2.2 Die CyberStage

Im Jahre 1992 wurde auf der SIGGRAPH Konferenz die CAVE, ein ca. 3 m x 3 m x 3 m großer Raum mit Stereo-Projektion auf drei Wände und den Boden vorgestellt (Cruz-Neira et al., 1993). Diese wurde für die interaktive wissenschaftliche Visualisierung und Präsentation auf Konferenzen konzipiert. Bereits bei Cruz-Neira et al. (1993) wurde angemerkt, daß die zusätzliche Verwendung von räumlichem Sound über ein Lautsprechersystem mit acht Boxen wünschenswert wäre.

Am Institut für Medienkommunikation wurde 1996 zusammen mit der Firma TAN Projektionstechnologie GmbH & Co. KG die CyberStage als erste CAVE-ähnliche VR-Umgebung in Deutschland errichtet (Abbildung 2.3). Zusätzlich zur Visualisierung wurden hier ein Vibrationssystem für den Boden (Soundfloor) und ein akustisches System installiert. Der Soundfloor kann über Vibrationen tiefe Frequenzen bis ca. 100 Hz wiedergeben und so die virtuelle Umgebung mit Wahrnehmungen, wie z.B. das Fahren auf einer Rolltreppe, unterstützen. Das erste, aus vier Lautspre-



chern bestehende akustische System wurde nach einem Jahr auf acht Lautsprecher erweitert und ist in der Lage, mit Hilfe von speziell entwickelter Software virtuelle Soundquellen im Raum zu plazieren und akustisch darzustellen (Eckel, 1999). Der akustische Eindruck wird allerdings beeinträchtigt durch Widerhall von den Spiegeln und durch die Tatsache, daß ein Teil der Boxen hinter den Projektionsflächen plaziert werden muß und dadurch der Schall gedämpft wird.

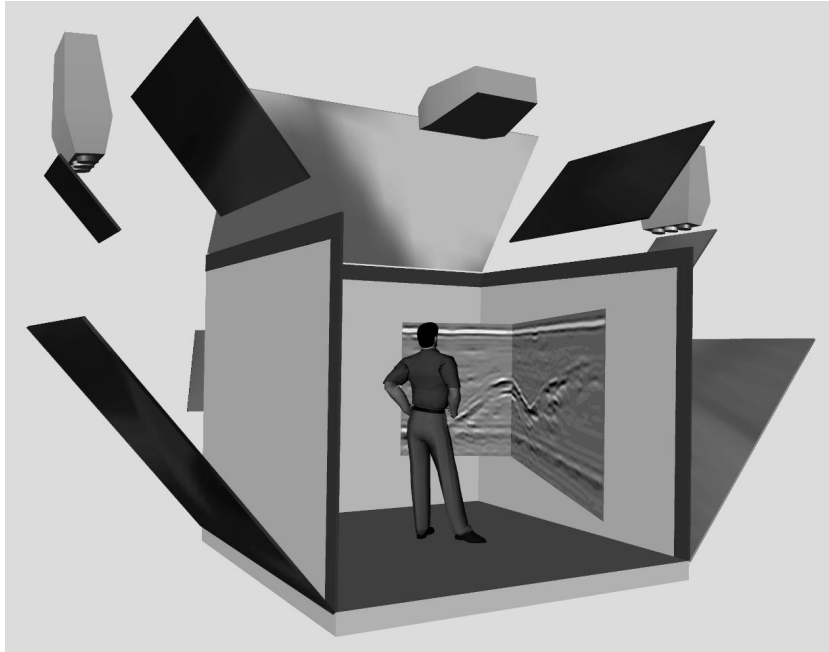


Abbildung 2.3: Modell der CyberStage

Die CyberStage hat eine Grundfläche von 3 m x 3 m, ist aber aus baulichen Gründen nur 2,25 m hoch. Dies wird bei großräumigen Modellen schnell als störend empfunden, da die Szene am oberen Ende abgeschnitten wirkt. Ebenfalls aus baulichen Gründen mußten, wie in Abbildung 2.3 dargestellt, die Projektionen auf die Seitenwände über mehrere Spiegel geleitet werden, was die Abbildungsqualität empfindlich beeinträchtigt. Die beste Abbildung bekommt man daher auf der Rückwand, wo die Projektion über nur einen Spiegel geleitet wird. Die Bildauflösung beträgt 1024 x 768 Pixel bei einer Bildwiederholrate von 120 Hz. Als Tracking-System kommt ein Polhemus Fasttrack zum Einsatz. Dieses weist am Rand des Raumes große Abweichungen auf, so daß der Interaktionsraum etwas eingeschränkt wird.

Die CyberStage läßt sich zum einen gut zum interaktiven Arbeiten mit sehr großem Interaktionsraum und zum anderen zur Präsentation von wissenschaftlichen Daten oder für Entertainment einsetzen. Hierbei ist jedoch zu beachten, daß sich

mehrere Personen schnell gegenseitig stören, wenn sie sich innerhalb der Szene bewegen, da sie den Blick auf die Projektionswände verdecken. Durch die Abbildung auf drei Wände können die Benutzer sehr unterschiedliche Blickrichtungen einnehmen, die Perspektive kann aber nur für eine Person korrekt dargestellt werden. Aus diesem Grunde sind Präsentationen auf “walk-through“ Anwendungen beschränkt, bei denen die Betrachter eng beieinander stehen und in dieselbe Richtung blicken, während sich die Szene bewegt.

Mittlerweile wurden die ersten Rundumprojektionen (sechseckige CAVES) errichtet, so z.B. an der Iowa-State-University und am Fraunhofer Institut für Arbeit und Organisation in Stuttgart.

## 2.3 Die Workbench

Bei der ersten, am Institut für Medienkommunikation entwickelten Workbench handelte es sich um ein Displaysystem mit einer Projektionsfläche in der Tischplatte (Krüger et al., 1995). Ziel war es, ein VR-System zu entwickeln, welches es ermöglicht, Aufgaben und Szenarien visuell zu simulieren, welche gewöhnlich in einer vergleichbaren Umgebung, also zum Beispiel an einer Werkbank oder an einem Zeichentisch, stattfinden. Beispiele waren medizinische Anwendungen (Lehre, Planung chirurgischer Eingriffe), Architektur (Vorstellung und Besprechung von Hausmodellen und Landschaftsplanungen) und das Design von Kraftfahrzeugen. All diesen Anwendungen war gemein, daß sie kooperativ sind und sich daher für den Einsatz von Head-Mounted-Displays nicht besonders gut eignen. Bei einer einseitigen Workbench tritt wie bei der stereoskopischen Darstellung am Bildschirm das Problem auf, daß der räumliche Eindruck bei Objekten, die sich vor dem Bildschirm befinden, sehr schnell gestört wird, wenn diese sich zu sehr dem Rand des Displays nähern. Da die Betrachtung in der Regel schräg auf die Projektionsfläche erfolgt, verstärkt sich dieser Effekt je weiter sich das Objekt virtuell vor der Projektionsfläche befindet. Hierdurch wird der Interaktionsraum eingeschränkt.

Die für diese Arbeit verwendete zweiseitige Workbench wurde 1997 in Zusammenarbeit mit der Firma TAN Projektionstechnologie GmbH & Co. KG entwickelt und wird von dieser unter dem Namen Holobench kommerziell vertrieben. Wie in Abbildung 2.4 zu sehen, erfolgt die Projektion von der Rückseite auf die Tischfläche und die rückseitige Wand. Die Projektionsflächen sind jeweils 110 cm x 180 cm

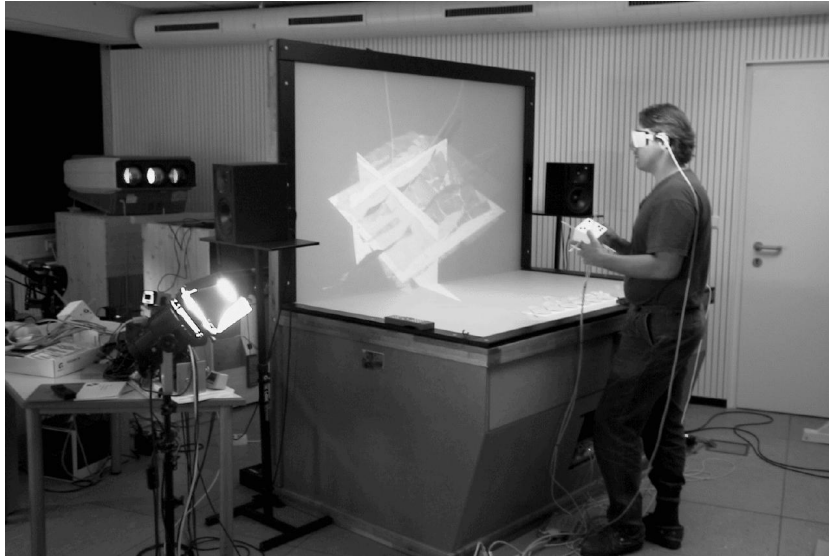


Abbildung 2.4: Die zweiseitige Workbench (Holobench der Firma TAN Projektionstechnologie GmbH & Co. KG in Düsseldorf)

groß, und die Bildauflösung beträgt 1024 x 768 Pixel bei einer Bildwiederholrate von 120 Hz. Zwischen diesen beiden Projektionsflächen spannt sich der Interaktionsraum auf. Im Vergleich zu Workbench-Systemen mit nur einer Abbildungsfläche erfordert die Verwendung einer zweiseitigen Workbench sehr viel mehr Rechenleistung und zur Zeit noch spezielle Graphikworkstations, welche in der Lage sind, beide Projektoren miteinander und mit den Shutterbrillen zu synchronisieren. Im Low-End-Bereich haben sich daher eher einseitige Projektionssysteme wie der Barco BARON-Projectiontable durchgesetzt, eine einseitige Workbench, bei der sich die Projektionsfläche kippen läßt. Bei der zweiseitigen Workbench bzw. bei der einseitigen im gekippten Zustand ergibt sich der Vorteil, daß alle betrachtenden Personen auf derselben Seite stehen und in dieselbe Richtung blicken. Daher ist die Perspektive für die Betrachter ähnlich, was gemeinsames Arbeiten ermöglicht.

## 2.4 Verwendete Rechnersysteme

Interaktive Stereo-Visualisierung stellt hohe Anforderungen an die verwendeten Computer. Die Rechner müssen nach Möglichkeit eine Bildwiederholrate von 120 Hz liefern können, um flimmerfreies Shuttern zu gewährleisten. Die Neuberechnung des Bildes in Abhängigkeit von der Benutzerperspektive und vom Tracking der

Interaktionsgeräte muß mindestens 10 - 15 mal pro Sekunde erfolgen, damit interaktives Arbeiten möglich ist und dem Benutzer nicht schwindlig wird (Framerate von 10-15 Hz notwendig). Um die erforderlichen Berechnungen ausreichend schnell durchzuführen, müssen meistens Multiprozessing-Rechner eingesetzt werden. Um die Daten von bildgebenden geophysikalischen Verfahren darzustellen (3D-Seismik, Tomographie), muß der Rechner zudem nach Möglichkeit über einen speziellen Bildspeicher verfügen (Texturspeicher).

Um virtuelle Umgebungen wie die CyberStage oder die Workbench zu betreiben, ist es außerdem erforderlich, daß der Rechner in der Lage ist, gleichzeitig mehrere Displaysysteme im Stereo-Modus anzusteuern. Um hierbei ein einwandfreies Bild zu gewährleisten, muß zum einen der Wechsel von einem Frame zum darauffolgenden bei den verschiedenen Graphikeinheiten (z.B. zwei bei der Workbench, vier bei der CyberStage) synchronisiert sein, zum anderen muß bei ihnen das Ausgangssignal für die verschiedenen Projektoren miteinander und mit den Shutterbrillen synchron verlaufen.

Die vorgenannten Anforderungen wurden während des Zeitraumes dieser Arbeit im wesentlichen nur von Silicon Graphics Onyx2 InfiniteReality2 Workstations erfüllt, welche den Standard in der VR-Forschungsgemeinde darstellen. Die Rackmount-Systeme sind unterteilt in Prozessoreinheiten, welche aus mehreren Nodeboards mit jeweils zwei Prozessoren und dazugehörigem Speicher bestehen (cc-Numa Architektur) und Graphikeinheiten. Der verteilte Speicher wird vom Betriebssystem so verwaltet, daß der Benutzer ihn wie einen gemeinsamen Speicher (Shared Memory) benutzen kann. Jedes Rack verfügt über eine Graphikeinheit mit jeweils zwei Graphikpipelines. Jede Graphikpipeline kann bis zu zwei Projektoren oder Bildschirme ansteuern, wobei dann allerdings Einbußen in der Performance hingenommen werden müssen. Mehrere dieser Racks lassen sich miteinander koppeln, so daß das System in einem weiten Bereich je nach Anforderungen skaliert werden kann. Die am Institut für Medienkommunikation für den Demonstrationsbetrieb eingesetzten Konfigurationen reichten von 4 Mips R10000 Prozessoren und zwei Graphikpipelines bis zum Einsatz von 12 Mips R10000 Prozessoren und 4 Graphikpipelines. Die Entwicklungsarbeit konnte auf Silicon Graphics O2-Desktop-Workstations erfolgen, wobei dann allerdings keine Stereo-Visualisierung eingesetzt wurde.

Im Verlauf der Arbeit erreichten auch die für PCs verfügbaren Graphikkarten eine ausreichende Performance, um VR zu betreiben, und einige unterstützten auch die Synchronisation mit Shutterbrillen. Allerdings ist die Bildqualität oft schlech-

ter, da die PC-Graphikkarten zum Beispiel über schlechteres Antialiasing verfügen als die Workstations der Firma Silicon Graphics (SGI). Der Fortschritt auf dem PC-Graphik-Markt wird hauptsächlich von der Spiele-Industrie angetrieben, die auf Desktopbetrieb fixiert ist. Daher gibt es bei den Graphikkarten für PCs keine Möglichkeit zur Synchronisation des Videosignals und es können nur Displays mit einer Projektion angetrieben werden. Eine Ausnahme stellt die Verwendung von passivem Stereo dar, da hier mehrere PCs eingesetzt werden können, welche über das Netzwerk synchronisiert werden.

## 2.5 Arbeiten im Bereich Low-Cost VR

Die bisher vorgestellten und in dieser Arbeit verwendeten Displays kamen alle aus dem sehr kostenintensiven Bereich der High-End-Visualisierung (Verwendung mehrerer Projektionsflächen, aktives Stereo, Verwendung von sehr schnellen Rechnern). Die hiermit verbundenen Kosten sind für einen alltäglichen Einsatz im gewissenschaftlichen Umfeld in der Regel zu hoch. Wenn man bereit ist, Abstriche bezüglich der Immersivität, der Abbildungsqualität oder der Flexibilität beim Einsatz zu machen, lassen sich jedoch auch wesentlich günstigere Visualisierungsumgebungen entwickeln, die für VR angewendet werden können. Nachfolgend sollen hier Arbeiten vorgestellt werden, welche sich mit der Entwicklung von Displaysystemen für die immersive Visualisierung beschäftigen, die erschwinglich sind und so VR für ein breiteres Publikum verfügbar machen.

In Gardner & Boswell (1998) bzw. in Whitehouse (1999) wird die Wedge vorgestellt, ein relativ preisgünstiges Displaysystem, welches an der Australian National University in Canberra entwickelt wurde, um die Möglichkeit zu schaffen, VR für die Ausbildung von Mathematik- und Physikstudenten zu verwenden. Es handelt sich um zwei über Eck aufgestellte Projektionswände mit Rückprojektion (Abbildung 2.5). Auf diese Weise ergibt sich wie bei der zweiseitigen Workbench ein großer Interaktionsraum, in welchem man jedoch bei der Wedge herumlaufen kann, so daß die Objekte besser erreichbar sind. Die Berechnungen wurden mit einer Intergraph TDZ2000-Windows-NT-Workstation ausgeführt, welche mit zwei Graphikkarten ausgestattet war. Als Software wurde pSpace implementiert, eine Rahmenapplikation, welche den Szenengraphen der Cosmo3D-API nachbildet. Die Projektion erfolgte mit aktivem Stereo bei 120 Hz und einer Auflösung von 1280 x 1024 Pixel. Laut Gardner & Boswell (1998) war Interaktivität und Head-Tracking

mit mehr als 10000 - 15000 Polygonen nicht möglich. Diese Zahlen dürften allerdings bei der aktuell verfügbaren Hardware bereits wesentlich höher liegen. Die erste Ausführung hatte zwei 1,5 m x 1,5 m große Wände und die Gesamtkosten lagen unter 100 TDM. Ein zweites System wurde für Präsentationszwecke im Powerhouse Museum Sydney, einem Museum für Technologie, aufgestellt und verfügte über 2,9 m x 2,3 m große Projektionsflächen. Das neueste an der Australian National University aufgestellte System verfügt über 4,2 m x 2,2 m große Displays (Whitehouse, 1999).



Abbildung 2.5: Die Wedge, ein Low-Cost Displaysystem

In Häfner et al. (2000b) wird ein weiteres Low-Cost-System vorgestellt, welches am Fraunhofer Institut für Arbeit und Organisation in Stuttgart entwickelt wurde. Es besteht aus einer 2,0 m x 1,5 m großen Projektionswand, auf welche von der Rückseite mit Hilfe von zwei Liesegang ddv-1500 Projektoren eine passive Stereoprojektion mit einer Auflösung von 1024 x 768 Pixeln erfolgt. Für das Head-Tracking kommt ein optisches Trackingsystem (Origin Dynasight) zum Einsatz, welches die Position, aber nicht die Orientierung liefert (3DOF-Tracking). Daher wird immer die Annahme gemacht, daß der Betrachter auf ein Objekt zwei Meter hinter der Projektionswand blickt. Das Berechnen der Bilder erfolgt mit zwei handelsüblichen PCs, bei welchen das Umschalten der beiden Bildzwischenspeicher miteinander synchronisiert ist.

Ein sich von den bisher vorgestellten Visualisierungssystemen unterscheidendes Konzept zeigt das Reachin-Display der schwedischen Firma Reachin Technologies AB (Reachin, 2001), welches zu einem wesentlich günstigeren Preis zu erwerben ist als die anderen vorgestellten Systeme (ca. 20.000 \$). Das System ist in der Abbildung 2.6 dargestellt.



Abbildung 2.6: Die "Haptic Workbench" der Firma Reachin Technologies AB

Die Darstellung auf dem Monitor wird für den Benutzer in der teildurchlässigen Glasplatte gespiegelt, welche der Benutzer von oben betrachtet. Die virtuellen Objekte befinden sich dann scheinbar unterhalb der Glasplatte, wo der Benutzer mit ihnen interagieren kann. Für die Interaktion steht ein kraftrückgekoppeltes Eingabegerät (Phantom) zur Verfügung. Bei dem Reachin-Display befindet sich die Hand vom Benutzer aus hinter dem Spiegel. Mit Hilfe des Tracking-Systems wird sie aber in der virtuellen Welt dort abgebildet, wo der Benutzer sie seinem Gefühl nach wahrnimmt. Auf diese Weise werden die Stereo-Visualisierung und die kraftrückgekoppelte Eingabe optimal miteinander integriert. Die Abbildungsfläche ist aber nur ca. 26 cm x 20 cm groß, so daß nur eine Person zur Zeit mit dem Gerät arbeiten kann und es sich daher nicht für kooperatives Arbeiten eignet. Zusätzlich zu der Hardware liefert die Firma auch eine speziell für diese Visualisierungsumgebung entwickelte Programmbibliothek, die die Softwareentwicklung vereinfacht.





# Kapitel 3

## Softwarebetrachtungen

### 3.1 Anforderungen an die verwendete Software

Bevor die virtuelle Szene für den Benutzer perspektivisch korrekt sichtbar wird, muß der Computer eine hohe Anzahl von Berechnungen ausführen. So muß er zuallererst das Trackingsystem evaluieren, das gesamte Model entsprechend dem Standpunkt und der Blickrichtung des Benutzers transformieren und danach entscheiden, welche der Objekte für den Benutzer sichtbar sind. Darauf aufbauend erfolgt dann die Abbildung auf Projektionsflächen, deren Konfiguration vom verwendeten Display-system abhängt. Um wie in dieser Arbeit geologische Modelle und geophysikalische Daten interaktiv in verschiedenen Umgebungen zu visualisieren, ergeben sich daher hohe Anforderungen an die Hard- und Software.

- Da die virtuellen Szenen oft bis zu 100.000 Polygone und 200.000 Punkte enthalten, muß die Rechengeschwindigkeit sehr hoch sein. Daher ist nach Möglichkeit die Nutzung von Multiprocessing, die zusätzliche Verwendung von Optimierungsstrategien wie Level of Detail Methoden und eine effiziente Verwaltung des Szenengraphen notwendig.
- Die Abbildung auf mehrere Projektionsflächen mit unterschiedlicher Konfiguration und Orientierung muß möglich sein (Fähigkeit für Multipipe-Darstellung).
- Die Entwicklung von Applikationen sollte unabhängig vom Displaysystem sein,

so daß eine einmal entwickelte Applikation zum Beispiel sowohl in der Cyber-Stage als auch auf der Workbench und am Monitor läuft. Dies läßt sich aber aufgrund der unterschiedlichen Benutzerschnittstellen nur im begrenzten Maße erreichen.

- Es muß die Möglichkeit bestehen, verschiedene Eingabegeräte wie zum Beispiel die Cubic-Mouse und das Trackingsystem in die Applikation einzubinden.
- Um die Entwicklungszeiten zu verkürzen, ist es wünschenswert, die Anwendungen interaktiv entwickeln zu können, zum Beispiel mit Hilfe einer Interpretersprache (Rapid-Prototyping).

Um den hohen Anforderungen an die Darstellungsgeschwindigkeit gerecht zu werden, bietet sich bei der Implementierung die Verwendung der IRIS Performer Graphikbibliothek an (Rohlf & Helmann, 1994). Hierbei handelt es sich um eine Szenengraphen-basierte Bibliothek, welche auf OpenGL aufsetzt und mit Hinblick auf die speziellen Anforderungen bei der Echtzeitvisualisierung entwickelt wurde. Daher werden verschiedene Optimierungsstrategien angewendet und die Verwendung von Shared Memory und Multiprocessing für eine kleine Anzahl an Prozessoren unterstützt (10 er Bereich). Die Implementierung von Multiprocessing erfolgt durch die Verwendung von Processing-Pipelines. Diese teilen die verschiedenen Aufgaben, die der Rechner zur Darstellung der Szene abarbeiten muß, auf. Zuerst werden die Berechnungen der eigentlichen Anwendung wie z.B. Rotation, Translation und das Erstellen von Objekten ausgeführt (Application). Danach wird getestet, welche der Objekte sich im sichtbaren Bereich des Benutzer befinden und daher dargestellt werden müssen (Culling). Zuletzt erfolgt die Darstellung der sichtbaren Objekte über die Graphikeinheit in den sogenannten Framebuffer (Drawing), in dem dann sozusagen das dem momentanen Zustand entsprechende perspektivisch korrekte Bild vorliegt (bei Stereo-Visualisierung jeweils für rechtes und linkes Auge). Um interaktives Arbeiten zu ermöglichen, muß dieser Vorgang mindestens 10 - 15 mal pro Sekunde wiederholt werden. Aus dem Framebuffer werden die Bilder ausgelesen und projiziert. Dies geschieht in der Regel mit mindestens 60 Hz.

Die vorgenannten Schritte *Application*  $\rightarrow$  *Cull*  $\rightarrow$  *Draw* lassen sich auf verschiedene Prozesse aufteilen und werden dann framesequentiell bearbeitet. Beim Einsatz von mehreren Projektionen können für jede Graphikeinheit ein CULL- und ein DRAW-Prozeß verwendet werden. Die Verwaltung des Szenengraphen in einer Art Datenbank kann in einem weiteren Prozeß erfolgen. Abbildung 3.1 zeigt als Beispiel eine mögliche Prozeßkonfiguration für die Visualisierung eines digitalen Modells

auf einer zweiseitigen Workbench mit Hilfe von zwei Graphikpipelines. Jeder rote Rahmen stellt hierbei einen Prozeß dar, insgesamt kann die Rechenarbeit für dieses Display also auf bis zu sechs Prozesse aufgeteilt werden. Bei der Verwendung von zwei Graphikeinheiten ist es wichtig, daß beim Wechsel der Perspektive oder des Objektzustandes das Bild für beide Projektionen gleichzeitig geändert wird. Daher müssen, wie in Abbildung 3.1 zu sehen, die Wechsel der Framebuffer miteinander synchronisiert werden (Swaplock). Bei der Verwendung von aktivem Stereo mit Shutterbrillen ist es außerdem wichtig, daß das Videosignal für die beiden Projektionen synchronisiert ist (Genlock). IRIS Performer wurde von Mitarbeitern der Firma Silicon Graphics entwickelt und zeigt seine Stärken besonders auf deren Onyx Multiprocessing Visualisierungsrechnern. Daher war diese Graphikbibliothek lange Zeit auch nur für deren Betriebssystem IRIX verfügbar. Neuerdings wird Performer jedoch auch für Linux Rechner herausgegeben.

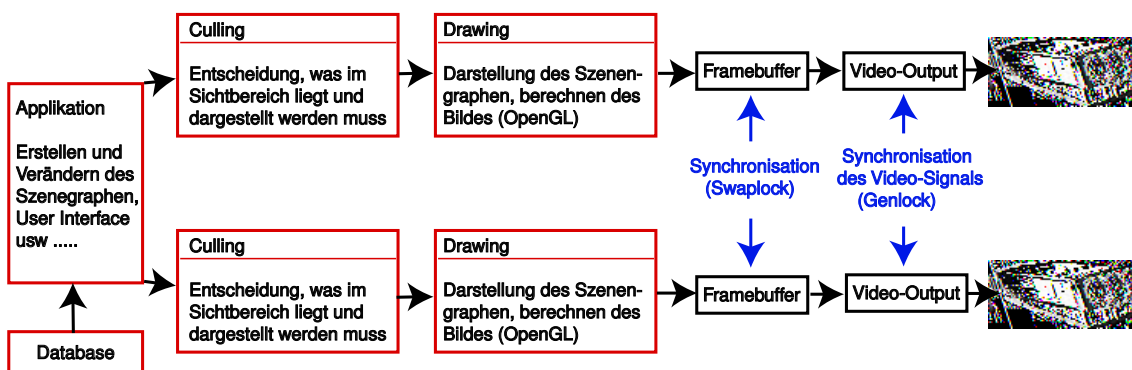


Abbildung 3.1: Mögliche Prozeßkonfiguration für die Visualisierung über zwei Graphikausgänge. Jedes rote Kästchen stellt einen Prozeß dar. Will man aktives Stereo mit Shutterbrillen verwenden, muß jeweils zwischen den beiden Graphikeinheiten das “Swappen“ der Framebuffer und das Videosignal synchronisiert werden.

Während die für VR notwendige Hardware und entsprechende Graphikbibliotheken auf dem Markt erworben werden können, gab es insbesondere am Anfang und während des Hauptteiles des Bearbeitungszeitraumes keine kommerziellen Visualisierungsprogramme, welche den hier spezifizierten Anforderungen gerecht werden. Um eine interaktive Applikation bereitzustellen, welche verschiedenartige geowissenschaftliche Datensätze visualisiert und dabei ausreichend flexibel ist, verschiedene Interaktionsgeräte einzubinden und verschiedene Displaysysteme anzusteuern, muß daher die Anwendung aufbauend auf VR-Entwicklungsbibliotheken selbst erstellt werden. Bei diesen Entwicklungsumgebungen handelt es sich meist um Rahmenapplikationen, die das Ansteuern der verschiedenen Displaysysteme und das Ein-

binden der Eingabegeräte übernehmen und verschiedene Interaktionsmöglichkeiten bereitstellen. Mit Hilfe von Interpretersprachen wie Scheme oder TCL lassen sich die Anwendungen auch noch zur Laufzeit bearbeiten. Beispiele für solche Rahmenapplikationen sind das am Fraunhofer Institut für Medienkommunikation entwickelte und in dieser Arbeit verwendete AVANGO (Tramberend, 1999) und das am Fraunhofer Institut für Arbeitswirtschaft und Organisation entwickelte Lightning (Blach et al., 1998). Bei Verwendung solcher Bibliotheken ist es möglich, die Darstellung mit einem vertretbaren Aufwand unter Zuhilfenahme von Standard Visualisierungsbibliotheken zu implementieren.

Bei beiden vorgenannten Rahmenapplikationen war man aufgrund der zugrundeliegenden Graphikbibliothek auf Rechner der Firma Silicon Graphics (SGI) festgelegt. Bei AVANGO wird es angestrebt, die Rahmenapplikation bis Ende des Jahres 2001 auch auf Linux-Rechnern lauffähig zu machen. Eine VR-Rahmenapplikation die nicht auf Rechner der Firma Silicon Graphics angewiesen ist, ist die Applikation ViSTA, welche an der RWTH Aachen entwickelt wurde. ViSTA basiert auf der kommerziellen C-Bibliothek WorldToolKit (WTK, Engineering Animation Inc.) und ist daher unabhängiger von der Hardware-Plattform (van Reimersdahl et al., 2000). Die eigentliche Implementierung für die Visualisierung kann dann mit Hilfe des Visualisierungs-Toolkits (VTK) (Schroeder et al., 1996) erfolgen.

Die in dieser Arbeit gewählten Darstellungsformen der Daten sind oft ähnlich zu ihrer möglichen Darstellung mit Standardpaketen, so daß sich die später vorgestellten Daten und Modelle zum Beispiel auch mit Hilfe des in der wissenschaftlichen Visualisierung weit verbreiteten Programmpaketes AVS der Firma Advanced Visual Systems darstellen ließen (Advanced Visual Systems, 2001). Dieses verfügt über ein leistungsfähiges grafisches Benutzerinterface, welches es erlaubt, auch ohne Programmierkenntnisse komplizierte Visualisierungsanwendungen zu entwickeln. AVS war allerdings erst zum Ende dieser Arbeit in der Lage, auch mehrere Projektionsflächen anzusteuern (AVS-Multipipe Edition) und hat zudem einen recht hohen Preis. Zum Zeitpunkt des AVS-Benutzertreffens im Februar 2001 konnten außerdem noch keine speziellen Eingabegeräte eingebunden werden, so daß die Entwicklung von Applikationen, die eine direkte Interaktion im Raum erlauben, noch nicht möglich war. Der Benutzer würde in diesem Fall immer auf einen externen Operator angewiesen sein, der am Rechner sitzt und für ihn die Anwendung steuert.

Da es in dieser Arbeit das Ziel war, interaktive Beispielanwendungen zu entwickeln und mit diesen geophysikalische Daten auf verschiedenen Displaysystemen

darzustellen, mußte also die dafür notwendige Software weitgehend selber entwickelt werden. Die Implementierung baut auf der Rahmenapplikation AVANGO auf, welche im weiteren vorgestellt werden soll.

## 3.2 Die Rahmenapplikation AVANGO

Um die Entwicklung von interaktiven und verteilten VR-Anwendungen mit relativ geringem Aufwand zu ermöglichen, wurde am Institut für Medienkommunikation die objektorientierte Rahmenapplikation AVANGO entwickelt (Tramberend, 1999). Diese liest beim Starten Konfigurationsfiles für die verschiedenen Displaysysteme ein, so daß man dieselbe Applikation auf sehr unterschiedlichen virtuellen Umgebungen laufen lassen kann. Ein Scripting-Interface, das auf der Sprache Scheme beruht (zu Scheme siehe z.B. Harvey & Wright, 1994), bietet die Möglichkeit, die Anwendung interaktiv zu verändern und weiterzuentwickeln, ohne jedesmal den Code neu kompilieren zu müssen. Die Instantiierung von Objekten (das Erzeugen von Objekten) und ihr Zusammenbau zu einer Szene erfolgt wahlweise interaktiv von der Shell oder in Skripten, so daß ein schneller Entwicklungsprozeß (rapid prototyping) möglich ist. Für komplexe performance-kritische Aufgaben kann die Rahmenapplikation um neue Objekte, welche in C++ implementiert sind, erweitert werden.

AVANGO setzt auf IRIS Performer auf und bildet dessen Szenengraphen nach, so daß vom Scripting-Interface aus die Klassen der libpf (z.B. pfDCS, pfNode, pfLOD usw.) und einige Klassen der libpr (pfMaterial, pfGeoState usw.) zur Verfügung stehen. Diese haben dann in AVANGO den gleichen Namen, wobei das Präfix "pf" jeweils durch "fp" ersetzt wird (also fpDCS, fpNode, fpLOD). Von Open Inventor (Wernecke, 1994) wurde der Mechanismus übernommen, die Objekte als Container für Felder zu implementieren, wobei die Felder den Zustand des Objektes definieren. Wird ein Feld geändert, so werden die notwendigen Aktionen ausgeführt, um das Objekt dem hierdurch neu definierten Zustand anzupassen. Wird bei einem Objekt z.B. das Feld Matrix mit einer Rotationsmatrix belegt, so wird das Objekt entsprechend gedreht. Ferner lassen sich Felder miteinander verbinden, um Abhängigkeiten zwischen den Objekten abzubilden. Eingabegeräte werden als "Sensoren" in die Applikation eingebunden, so daß auf ihre Orientierungskordinaten und z.B. auf den Zustand von Knöpfen zugegriffen werden kann.

Das nachfolgende Stück Scheme-Code soll als simples Beispiel dafür dienen, wie

eine sehr einfache Applikation aussieht. Wird dieser Code im Interpreter ausgeführt, so werden zuerst zwei Objekte der Klasse `fpFile` erzeugt und danach jeweils beim Setzen des Dateinamens die Geometrie, die in den entsprechenden Dateien definiert ist, eingelesen. Diese beiden Objekte werden dann unter den gemeinsamen Gruppenknoten gesetzt und als Gruppe 45 Grad um die Z-Achse rotiert.

```
(define file-1 (make-instance-by-name "fpFile"))
(fp-set-value file-1 'Filename "geometry-file-1.iv")
(define file-2 (make-instance-by-name "fpFile"))
(fp-set-value file-2 'Filename "geometry-file-2.iv")
(define parent (make-instance-by-name "fpDCS"))
(fp-set-value parent 'Children (list file-1 file-2))
(fp-set-value parent 'Matrix (make-rot-mat 45.0 0.0 0.0 1.0))
(av-add parent)
```

Um mit den einzelnen Objekten zu interagieren, werden virtuelle Werkzeuge verwendet. Diese werden an den Stylus angehängt und verfügen über einen Strahl, mit dem man auf die Objekte zeigen kann. Die Implementierung der verschiedenen Interaktionen folgt dem Konzept von Tramberend et al. (1999). Hier werden Mediatoren definiert, welche in das "Dragger"-Feld der Objekte eingetragen werden. Für die verschiedenen möglichen Paare von Mediatoren und Werkzeugen werden jeweils spezifische Interaktionsoperatoren definiert. Wird mit einem virtuellen Werkzeug ein Objekt im Szenengraphen ausgewählt, so wird im Graphen aufwärts nach einem Mediator gesucht, für den der Interaktionsoperator für das Paar Werkzeug/Mediator existiert, und dieser dann aufgerufen. Die eigentliche Interaktion ist in den Interaktionsoperatoren definiert. Der folgende vereinfachte Code baut auf dem vorherigen auf und soll die Verwendung demonstrieren:

```
(define matrix-mediator (make-instance-by-name "fpMatrixDragger"))
(fp-add-1value file-2 'Dragger matrix-mediator)
(fp-connect-from file-2 'Matrix matrix-mediator 'Matrix)
(define drag-tool (make-instance-by-name "fpDragTool"))
(fp-set-value av-stylus 'Tool drag-tool)
```

Wird mit dem am Stylus befestigten `drag-tool` das Objekt `file-2` ausgewählt, so wird der entsprechende Interaktionsoperator zwischen dem `MatrixDragger` und dem `DragTool` aufgerufen. Dieser definiert, daß die Transformationsänderung des Stylus

mit der vorherigen Transformation des Objektes verknüpft wird und die Gesamttransformation in das Matrix-Feld des Mediators und damit auch in das Matrix-Feld des Objekts file-2 eingesetzt wird. Auf diese Weise läßt sich das Objekt file-2 bewegen und rotieren. Wählt man hingegen das Objekt file-1 aus, so passiert gar nichts, da die Suche im Szenengraphen nach einem passenden Mediator erfolglos verläuft. Um den Überblick über den Aufbau des Szenegraphen zu behalten, läßt sich dieser, wie in Abbildung 3.2 zu sehen, grafisch darstellen.

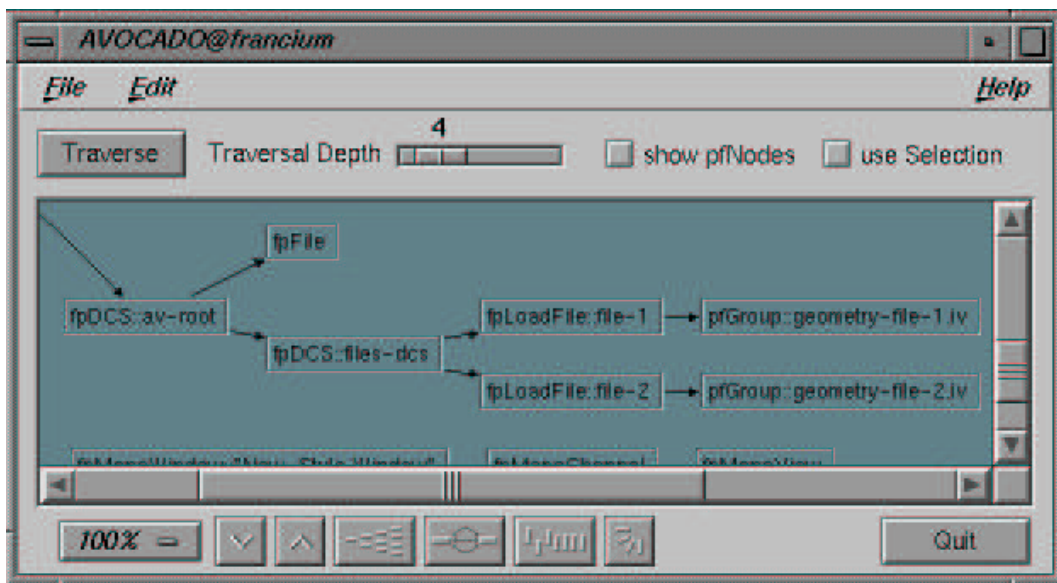


Abbildung 3.2: Grafische Darstellung des Szenegraphen bei AVANGO. Hier anhand des Beispielcodes wie vorstehend.

Durch Ableitung von AVANGO-Klassen besteht die Möglichkeit, in C++ ganz neue Klassen mit eigener Funktionalität zu implementieren. Dies ist zum Beispiel notwendig, wenn die Möglichkeit bestehen soll für ein Problem spezifische Funktionen auf die Daten anzuwenden. So wurden in dieser Arbeit zum Beispiel neue Klassen und für diese spezifische Mediatoren und Interaktionsoperatoren entwickelt, um mit Georadar-Daten, Bohrlochdaten und seismologischen Daten zu arbeiten. Dies ermöglicht es, spezielle Interaktionen zu definieren, wie zum Beispiel das Auswählen und Autotracken von Reflektoren bei den Georadar-Daten. Statische Modelle, bei denen keine spezifischen Interaktionen benötigt werden, lassen sich mit weit geringerem Aufwand darstellen, indem man sie mit einem Standardprogramm zum Modellieren wie 3D-Studio-Max der Firma Autodesk oder Creator der Firma Multigen-Paradigm erstellt und dann die gesamte Geometrie einfach aus der Datei einliest.

In der Abbildung 3.3 wird graphisch dargestellt, in welchem Umfeld die Entwick-

lung der Software-Prototypen stattfand. Wie bereits erwähnt, baut die Rahmenapplikation AVANGO auf die Graphikbibliothek IRIS Performer auf. Die Klassen, die speziell für die Visualisierung der geowissenschaftlichen Daten entwickelt wurden, bauen wiederum auf Basisklassen von AVANGO auf (werden von diesen abgeleitet), verwenden für die Implementierung der Visualisierung aber hauptsächlich IRIS Performer. Selten greifen diese Klassen auch direkt auf die OpenGL zu. Die Applikationen werden dann, basierend auf den AVANGO Basisklassen und den speziell für die geowissenschaftlichen Daten entwickelten Klassen, mit Hilfe der Interpretersprache Scheme entwickelt.

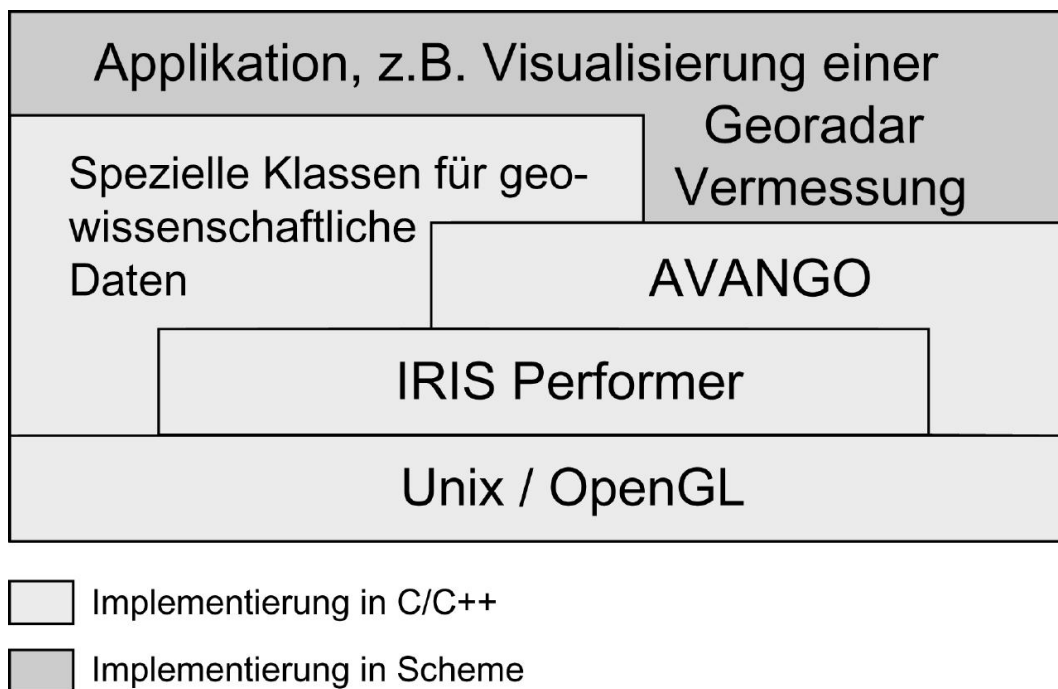


Abbildung 3.3: Graphische Darstellung des Programmierumfeldes bezüglich der Entwicklung der Software-Prototypen.

Die geographisch verteilte Visualisierung von AVANGO-Objekten beruht auf der Verteilung ihrer Felder und damit auch der Belegung der Felder. Von jedem Objekt werden so an allen Orten Kopien angelegt und die Werte für die Felder eingetragen und synchron gehalten, so daß der Zustand der Objekte an allen Orten gleich ist. Somit ist durch die Darstellung der Abhängigkeiten an allen Orten eine Kopie des Szenengraphen vorhanden. Betrachtet man noch einmal den oben beschriebenen Beispiel-Code im Fall der verteilten Darstellung, so ergibt sich folgendes: Wird bei der Instanz file-1 das Feld "Filename" gesetzt, so wird die Geometrie aus dem entsprechenden File eingelesen. Im verteilten Fall wird außerdem das Feld "Filename"



auch bei der Instanz an der Remote-Lokation gesetzt, was bewirkt, daß auch die entsprechende Datei eingelesen wird. Wird eines der Objekte rotiert, so wird das zu ihm gehörende Feld "Matrix" neu gesetzt. Da dies auch an der Remote-Lokation geschieht, wird auch dort das Feld "Matrix" neu gesetzt und damit das Objekt entsprechend gedreht. Auf diese Weise läßt sich relativ einfach eine verteilte Anwendung entwickeln, wobei jedoch nur der Zustand, welcher anhand der Felder definiert ist, übertragen werden kann. Funktionalität, die für einzelne Klassen in C++ implementiert wird, ist nicht verteilbar, weswegen sich nur relativ simple Anwendungen mit geringer Funktionalität entwickeln lassen. Außer an dem Ort, an dem der Szenengraph zusammengebaut wurde, ist dieser anonym. Das heißt, daß zu den einzelnen Objekten keine "Pointer" existieren und diese auch keine Namen haben. Dies bringt Schwierigkeiten mit sich, auf die von Springer et al. (2000) genauer eingegangen wird. In der vorliegenden Arbeit wird das Problem umgangen, indem die wichtigsten Objekte mit Hilfe des Namenfeldes gekennzeichnet werden. Beim Starten der Anwendung an der Remote-Lokation können diese Objekte dann im Szenengraphen gesucht und ihnen ein Symbol zugewiesen werden, mit dem sie angesprochen werden können. Dies funktioniert allerdings nur mit Objekten, deren Existenz von Anfang an feststeht.



# Kapitel 4

## Allgemeine Anwendungsbetrachtungen

Um die Möglichkeiten besser diskutieren zu können, welche sich durch die Verwendung von VR in der in den vorigen Kapiteln dargestellten Form für die Geowissenschaften ergeben, sollen in diesem Kapitel zuerst einmal die verschiedenen Vor- und Nachteile dieser Technologie im Vergleich zur 3D-Visualisierung auf Standard-Desktop-Workstations erläutert werden. Als Vorteile werden zum Beispiel die Darstellung in Stereo, die Verwendung von großen Projektionsdisplays, die direkte Interaktion im dreidimensionalen Raum und die in der Regel mit der Verwendung von VR verbundene High-Performance-Visualisierung gesehen. Insbesondere bei der Entwicklung der nachstehend beschriebenen Beispielmodelle zeigten sich jedoch allgemeine Probleme, die bereits hier erläutert werden sollen. Danach werden in diesem Kapitel verschiedene verallgemeinerte Szenarien aufgezeigt, wie und wofür sich virtuelle Umgebungen einsetzen lassen. In dieser Arbeit werden hierbei die Datenanalyse bzw. Interpretation, kooperatives Arbeiten und Remote-Collaboration betrachtet. Diese machen in unterschiedlicher Intensität von den verschiedenen Vorteilen, welche die Verwendung von VR bringt, Gebrauch.

### 4.1 Vor- und Nachteile der Anwendung

Die Stereo-Darstellung vereinfacht die Erkennung von räumlichen Zusammenhängen auch bei komplizierten geometrischen Verhältnissen oder bei einer extrem hohen

Anzahl von Daten. Dies liegt daran, daß man im Gegensatz zur Darstellung von 3D-Modellen am Bildschirm auch die räumliche Tiefe visuell unter Kontrolle hat. Jeder, der schon einmal mit Software zur 3D-Modellierung am Bildschirm gearbeitet hat, kennt den Umstand, daß man das Modell permanent in verschiedene Richtungen rotieren muß, um sich über den räumlichen Aufbau klar zu werden. Oft ist der dreidimensionale Zusammenhang nur aus bestimmten Perspektiven deutlich zu erkennen, wie es als Beispiel in der Abbildung 4.1 verdeutlicht wird. Dargestellt sind sechs Bohrlöcher, welche entsprechend der Stratigrafie eingefärbt sind. Aus der links in Abbildung 4.1 gewählten Perspektive ist die Mächtigkeit und der räumliche Verlauf der mit gelb kodierten Schicht nicht zu ersehen. Auf der rechten Seite wird eine günstigere Blickrichtung verwendet und der räumliche Eindruck durch Hilfslinien unterstützt, so daß der Verlauf der gelb gefärbten Schicht im Raum wesentlich besser zu sehen ist. Da bei der Stereo-Darstellung die Raumwinkel zwischen einzelnen Objekten (Orientierung) und deren relative Verschiebungen zueinander sofort sichtbar sind, entfällt diese Abhängigkeit von der Blickrichtung. Die räumliche Darstellung unterstützt somit die mentale Modellbildung, und man kann daher annehmen, daß sie dem Betrachter bei der iterativen Erstellung konsistenter 3D-Modelle aus lückenhaften oder heterogenen Daten hilft. Dies ist besonders deswegen interessant, weil sich solche Datensätze meist einer automatischen Datenverarbeitung entziehen.

Die Verwendung von großflächigen Projektionsdisplays unterstützt die Immersion und fördert so die mentale Modellbildung und die Konzentration auf den Datensatz. Kleine Details lassen sich besser wahrnehmen als am Bildschirm. Ein weiterer Vorteil ist die Möglichkeit, mit mehreren Leuten denselben Datensatz zu betrachten und über die Interpretation zu diskutieren. Für das Arbeiten mit mehreren Personen ist es in der Praxis allerdings notwendig, daß die Betrachter möglichst nahe beieinander stehen und daß das virtuelle Modell sich möglichst weit von den Personen entfernt befindet. So werden allzu starke perspektivische Verzerrungen vermieden.

Die direkte Interaktion ermöglicht ein sehr rasches und flüssiges Arbeiten und stellt ein wichtiges Argument für die Verwendung von virtuellen Umgebungen (Virtual Environments, VE's) dar. Sie ist besonders dann von Vorteil, wenn 4D-Modelle (räumlich-zeitlich) analysiert werden sollen, da es hier darauf ankommt, auf die Abläufe schnell reagieren zu können. Als ein einfaches Beispiel für die Interaktion im Raum kann die Orientierung eines Modells dienen. Am Bildschirm kann dies z.B. erfolgen, indem die Mausbewegung in eine Rotation der Szene umgesetzt wird, wobei die Achse als senkrecht zur Mausbewegung angenommen wird und meist durch den Bildschirmmittelpunkt verläuft (siehe zum Beispiel beim Programm SceneView-

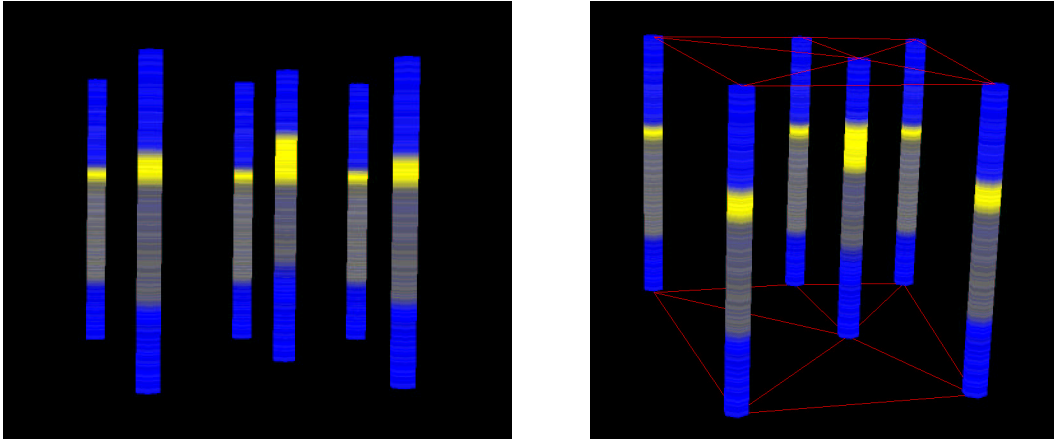


Abbildung 4.1: Zwei Darstellungen derselben sechs Bohrlöcher aus unterschiedlichen Perspektiven zur Verdeutlichung, welche Vorteile eine wirklich dreidimensionale Darstellung haben kann. Unterschiedliche stratigrafische Schichten werden über Farbkodierung angezeigt. Auf dem linken Bild ist der Verlauf der gelben Schicht im Raum nicht zu sehen. Auf der rechten Seite wird dieser durch die Wahl einer günstigen Perspektive und Hinzufügen von Hilfslinien sichtbar gemacht. Bei realen geologischen Fragestellungen ist die Abhängigkeit von der Perspektive und das Hinzufügen weiterer Elemente zur Unterstützung der Orientierung oft störend.

er, welches auf den meisten Silicon Graphics Rechnern mit IRIX Betriebssystem verfügbar ist und dem in Wernecke (1994) beschriebenen ExaminerViewer verwendet). Bei dieser Vorgehensweise ist das Resultat aber abhängig von der Ausgangslage des Modells und erfordert immer, daß man sich experimentell an die gewünschte Orientierung annähert. Eine andere Möglichkeit wäre die Spezifizierung der Rotationswinkel (z. B. Streichen und Fallen) und des Rotationszentrums mit Hilfe der Tastatur, was aber zum einen weniger intuitiv und zum anderen zeitaufwendiger ist. Bei einer virtuellen Umgebung hat man die Möglichkeit, die Orientierung des Modells an ein Eingabegerät zu koppeln, wie dies z.B. von Fröhlich & Plate (2000) für die Cubic-Mouse beschrieben wird. Will der Benutzer das Modell von unten sehen, dreht er einfach die Cubic-Mouse auf den Kopf. Ein anderes Beispiel, das in den im nachfolgenden Kapitel beschriebenen Modellen oft verwendet wird, ist die Festlegung eines entlang der Axen ausgerichteten Subvolumens. Dieses kann z.B. Schnittebenen steuern oder Abfragen an eine Datenbank generieren. Das Volumen wird anhand seiner acht Eckpunkte und der dazugehörigen Seiten definiert. Wird einer der acht Ecksteine verschoben, werden die dazugehörigen Seiten und ihre weiteren Ecksteine modifiziert, wie dies in der Abbildung 4.2 dargestellt ist. In einer

virtuellen Umgebung lassen sich diese Handlungen sehr einfach ausführen, da man bei der Verschiebung alle drei Raumrichtungen unter Kontrolle hat. Am Bildschirm müßte man erst die Ausdehnung einer Fläche spezifizieren und dann das Modell um 90 Grad drehen, um die Ausdehnung in der verbleibenden Raumrichtung anzugeben.

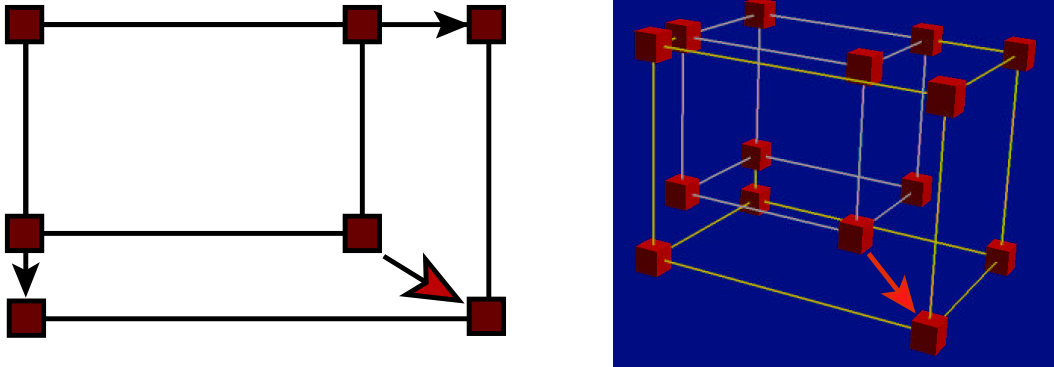


Abbildung 4.2: Definition eines achsenparallelen Unterbereiches in 2D und in 3D. Um den Unterbereich im 3D-Raum mit Hilfe einer 2D-Benutzerschnittstelle festzulegen, wären wesentlich mehr Interaktionen notwendig als mit einer speziell auf 3D-Interaktionen abgestimmte Schnittstelle, wie sie in virtuellen Umgebungen verwendet wird.

Bei der Verwendung von VR kommen fast immer spezielle High-End Graphik-Workstations zum Einsatz. Dies ermöglicht interaktive Visualisierung und Modellierung, auch von großen Datensätzen. Viele Prozesse bei der Interpretation von geowissenschaftlichen Daten verlaufen iterativ, da die Befunde häufig nicht eindeutig sind und sich einer automatischen Inversion entziehen. Zum Beispiel wird eine Grenzfläche anhand von seismischen Daten konstruiert, dann wird überprüft, wie gut die konstruierte Fläche mit den Bohrlochdaten übereinstimmt. Danach wird sie anhand dieser Daten korrigiert, um jetzt wieder die Übereinstimmung mit dem seismischen Befund zu klären. Die Möglichkeit, diesen Ablauf interaktiv zu gestalten, beschleunigt nicht nur die Interpretation, sondern ermöglicht einen spielerischen und experimentellen Umgang mit den Daten. Ein für die Erdölfirmen wichtiges Beispiel ist die Untersuchung von Datensätzen der 3D-Seismik.

Während dieser Arbeit zeigte sich, daß Nachteile für die angestrebten geowissenschaftlichen Anwendungen hauptsächlich in bezug auf die Präzision der Interaktion auftreten. Dies liegt daran, daß die Führung der Eingabegeräte frei im Raum erfolgt und der Benutzer sie darum nie völlig ruhig an einer Stelle halten kann. Zusätzlich wird die direkte Interaktion dadurch behindert, daß bei immersiven Projektionsdis-

plays keine Schärfentiefe existiert. Hat man ein Eingabegerät sehr nahe an einem virtuellen Objekt, so stellen die Augen die Schärfe entweder auf das Objekt und damit auf die Projektionsfläche ein, auf die das Objekt abgebildet wird, oder sie fokussieren auf das Eingabegerät. Daher hat man nie beides gleichzeitig scharf. Für exaktes Arbeiten wäre daher eine indirekte Interaktion notwendig, wie sie sich z.B. mit einer Space-Mouse (3DConnexion, 2001) realisieren läßt. Der Einsatz eines solchen Gerätes hat sich bei der Verwendung von CAD-Programmen bereits vielfach bewährt und könnte zusätzlich neben der direkten Interaktion eingesetzt werden, so daß sich beide Schnittstellen gegenseitig ergänzen. Auf diese Weise könnte der Benutzer einen virtuellen Zeiger im 3D-Raum je nach Bedarf sehr genau oder sehr schnell steuern.

Die Qualität des visuellen Eindruckes wird durch mehrere Faktoren negativ beeinflusst, die sich (auch aufgrund der ohnehin meist schon relativ schlechten Bildqualität) zwar als einzelne Effekte nur sehr schwer beobachten und quantifizieren lassen, in ihrer Summe aber die Qualität der Darstellung negativ beeinflussen und die Anwendung von VR, zum Beispiel bei der Datenanalyse, erschweren. Negative Auswirkungen ergeben sich zum Beispiel durch die Tatsache, daß die Repräsentation der räumlichen Tiefe in immersiven Visualisierungssystemen nur auf der Parallaxe beruht und keine Schärfentiefe existiert. So tritt zum Beispiel ein Konflikt zwischen der Einstellung der Augen (Schärfe) und ihrer Konvergenz bezüglich des Objektes auf (Southard, 1992). Dies soll in der nach Southard (1992) erstellten Abbildung 4.3 verdeutlicht werden. Die Fokussierung auf die Bildebene in Entfernung  $D_{real}$  geht normalerweise einher mit der Konvergenz der Augen auf den Punkt A mit dem Winkel  $\alpha$ . Wird ein virtueller Punkt B in der Entfernung  $D_{virt}$  betrachtet, so konvergieren die Augen mit dem Winkel  $\beta$ , bleiben aber scharfgestellt auf die Entfernung  $D_{real}$ , welche mit dem Winkel  $\alpha$  korrespondiert. Nach Southard (1992) gibt es Untersuchungen, daß von den meisten Menschen eine Winkeldifferenz von 1,6 Grad toleriert wird ( $\gamma = \alpha - \beta$ ). Führt man das virtuelle Objekt näher an sich heran, so wird die Parallaxe immer größer und macht sich im zunehmenden Maße als störend bemerkbar. Ferner unterstützt in der realen Welt die Schärfentiefe auch den Betrachter, sich bei einer hohen Informationsdichte auf bestimmte Bereiche innerhalb der Daten zu konzentrieren. Kleine Objekte, die sich zwischen dem Benutzer und dem interessanten Bereich befinden, werden in der realen Welt nur unscharf wahrgenommen und im Bewußtsein ausgeblendet. Ein Beispiel hierfür ist Schmutz auf der Windschutzscheibe eines Autos. In virtuellen Umgebungen ist der Raum in seiner gesamten Tiefe scharf, weshalb naheliegende Objekte oft viel stärker als störend wahrgenommen werden als in der Realität.

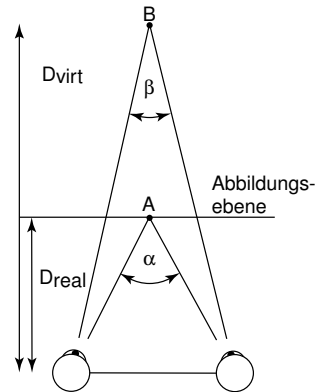


Abbildung 4.3: Konflikt zwischen der Scharfstellung der Augen und ihrer Konvergenz. Für weitere Erklärungen siehe Text.

Von den darzustellenden Daten und Modellen müssen verschiedene Randbedingungen eingehalten werden, damit die Verwendung von virtuellen Umgebungen sinnvoll ist. Wie in Bolas et al. (2000) verdeutlicht, führt die Diskretisierung der Parallaxe durch die Pixel des Projektionssystems zu einer Anzahl von Tiefenebenen. Daher kann man sich den Raum als ein (virtuelles) 3D-Gitter stereoskopischer Voxel vorstellen, welches sich mit dem Interaktionsraum überschneidet. Bei nicht voll immersiven Systemen (z.B. bei der Workbench) wird zusätzlich zum Rand der Projektionsflächen hin der räumliche Eindruck gestört. Dies gilt insbesondere, wenn sich die Objekte vor der Projektionsfläche befinden, da dann die als natürlich empfundene Ordnung für die Sichtbarkeit der Objekte verletzt wird und der Bildschirmrand als weiter hinten liegend die davor liegenden Objekte verdeckt. Bei der Workbench läßt sich daher nicht der gesamte Raum vor der Projektionsfläche als Modellraum nutzen, sondern nur ein prismenförmiger Unterbereich, der aber von der Person durch Hin- und Herbewegen ausgeweitet werden kann. Der Interaktionsraum definiert sich durch das Volumen, welches der Benutzer mit seinen Händen erreichen kann. Dies hängt hauptsächlich von der Armlänge ab. Von Hofmann (2001) wird zum Beispiel für den Greifraum vor der Brust eine maximale Entfernung von bis zu 65 cm, komfortabel nur bis zu 40 cm, angegeben. Zur besseren Verdeutlichung, welche Bereiche sich bei einer zweiseitigen Workbench für Visualisierung und Interaktion nutzen lassen, sind in Abbildung 4.4 die verschiedenen Raumbegrenzungen für eine Holobench als Projektion auf den Tisch skizziert. Will man direkt im Raum mit dem digitalen Modell arbeiten, so muß das Modell sich in dem Bereich befinden, wo sich der Interaktionsraum mit dem Modellraum überschneidet. Dieser Bereich ist relativ klein, weshalb es notwendig ist, sich zum Beispiel über den Tisch zu beugen



(um den Greifraum zu vergrößern) oder mit den einzelnen Objekten mit Hilfe eines Zeigestrahls zu interagieren (was weniger exakt ist).

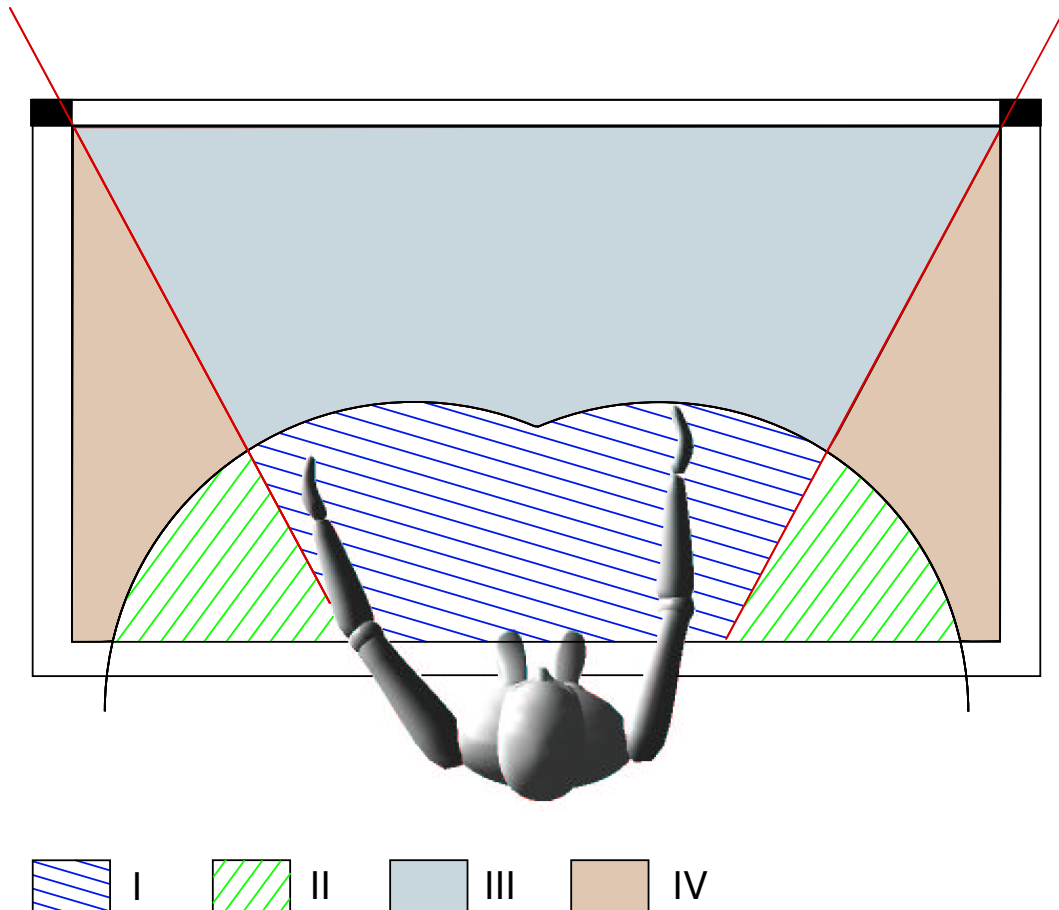


Abbildung 4.4: Skizze für die verschiedenen Zonen auf der Holobench bezüglich der Interaktion und der Visualisierung. (I): Die Objekte können in voller Höhe zwischen den beiden Projektionsflächen dargestellt werden und der Benutzer kann direkt mit ihnen interagieren. (II): Interaktion und Visualisierung von Objekten über der Projektionsfläche ist nur in geringer Höhe möglich. (III): Visualisierung von Objekten vor der Projektionsfläche ist auf voller Höhe möglich, der Benutzer kann aber nur unter erschwerten Bedingungen mit ihnen interagieren. Dieser Bereich streckt sich von der senkrechten Projektionsfläche nach hinten bis ins Unendliche aus. (IV): Objekte können über der Projektionsfläche nur in geringer Höhe dargestellt werden und sind nur schwer zu erreichen. Hinter der Projektionsfläche sind virtuelle Objekte auch nach unten in beliebiger Entfernung darstellbar.

Ein virtuelles Modell oder ein Datensatz, der visualisiert werden soll, muß entweder vollständig in den Modellraum und möglichst auch gleichzeitig in den Interakti-

onsraum abbildbar oder in einzelne Teilmodelle zerlegbar sein. Auf der anderen Seite sollte das Modell aber die zur Verfügung stehenden stereoskopischen Pixel so weit wie möglich nutzen. Hieraus ergibt sich, daß z.B. sehr längliche Datensätze wie die eines einzelnen Bohrloches über eine längere Strecke oder Datensätze ausgedehnter flachgründiger Untersuchungen für diese Darstellungsform ungeeignet sind, da sie nur einen sehr kleinen Bereich des stereoskopischen Voxelgitters nutzen.

## 4.2 Anwendungsszenarios

Die verschiedenen Vorteile, die sich durch die Verwendung von immersiven Displaysystemen ergeben, sind für die verschiedenen allgemeinen Anwendungsszenarios unterschiedlich wichtig. Hierbei soll unterschieden werden zwischen der Verwendung von virtuellen Umgebungen zur Visualisierung von Daten, ihrer Verwendung bei der geometrischen Modellierung und der Verwendung zur Verbesserung der Kommunikation zwischen verschiedenen Anwendern.

Beim klassischen Gebiet der “Scientific Visualisation“ bzw. der Datenanalyse geht es darum, Daten verständlich und übersichtlich darzustellen, um sie visuell zu analysieren. Hierfür ist die Stereo-Visualisierung und die Verwendung von High-End Workstations mit guter Abbildungsleistung sehr hilfreich. Je nach Fragestellung kann sich auch die direkte Interaktion im Raum als Vorteil erweisen. Die Verwendung von großen und daher kostenintensiven Displays erübrigt sich hier jedoch. Die Verwendung von VR zur geometrischen Modellierung stellt sehr hohe Anforderungen an die Genauigkeit der Interaktion. Wie bereits erwähnt, ist diese bei den momentan auf dem Markt verfügbaren Systemen noch nicht in ausreichendem Maße gegeben. Die direkte Editierung von Objekten im Raum bei gleichzeitigem Sehen von 3D-Zusammenhängen in den zugrunde liegenden Daten hätte sonst mit Sicherheit Potential.

Weitere Szenarios sind kooperatives Arbeiten und Remote-Collaboration. Im ersteren Falle betrachten mehrere Personen gleichzeitig das virtuelle Modell und können Informationen und ihre Ansichten über die verschiedenen Interpretationsmöglichkeiten austauschen. Damit flüssig diskutiert werden kann, ist hier schnelle Interaktion notwendig. Außerdem helfen die großen Projektionsdisplays bei der Betrachtung. Generell muß man in diesem Falle unterscheiden zwischen einem Szenario, bei dem alle Benutzer gleichberechtigt sind (z.B. Asset-Teams), und einem, bei dem

einzelne Personen bei der Interaktion bevorrechtigt sind (z.B. Professor-Studenten-Verhältnis in Vorlesungen). Generell müssen sich die Benutzer aber erst einmal an die Diskussion in den virtuellen Umgebungen mit ihren Eigenheiten gewöhnen und ihr Verhalten an diese anpassen. Zum Beispiel verleitet das räumlich wahrgenommene Modell die Benutzer oft dazu, mit dem Finger auf Objekte zu zeigen oder die Hand zu benutzen, um den Verlauf von Flächen im Raum anzudeuten. Da die anderen Betrachter die Perspektive jedoch verzerrt sehen, nehmen sie die Objekte oder die Fläche an der falschen Stelle wahr. In Agrawala et al. (1997) wird ein Szenario beschrieben, in dem die Perspektive für zwei Benutzer getrennt berechnet und mit aktivem Stereo dargestellt wird. Insgesamt wurden vier Bilder framesequentiell abgebildet (für zwei Benutzer jeweils rechtes und linkes Auge). Dies hat allerdings den Nachteil, daß das Bild für die Benutzer dunkler erscheint. Dafür sehen beide ein unverzerrtes Bild und es ist möglich, Gestik in der Umgebung zu benutzen. Interessant ist auch der Ansatz, daß beide Personen jeweils unterschiedliche Ergänzungen zu der allgemeinen Szene sehen können.

Bei der Remote-Collaboration diskutieren mehrere Personen von geographisch verteilten Visualisierungssystemen aus. Die Anforderungen an Interaktion und Visualisierung sind hier ähnlich, aber höher als beim kooperativen Arbeiten, da keine Gestik für die zwischenmenschlichen Kommunikation zur Verfügung steht. Wichtig ist, daß die Personen sich über die Daten und deren Informationen austauschen können (datenzentrierter Ansatz). Dies steht im Gegensatz zum personenzentrierten Ansatz, wie er z.B. bei Videokonferenzen auftritt. Am besten wäre es, beide Ansätze miteinander zu koppeln, wie es z.B. von Goebbels (2001) für den Bereich der medizinischen Visualisierung realisiert wurde. Dies würde jedoch den Umfang der vorliegenden Arbeit sprengen.



# Kapitel 5

## Stand der Forschung

Eine Übersicht über die geschichtliche Entwicklung und die frühen Anwendungsgebiete der Virtuellen Realität (VR) ist in Bormann (1994) gegeben. Hiernach kamen die ersten Anstöße für die Entwicklung dieser Technologie aus dem Bereich Militär und Raumfahrt, wo man sie zum Beispiel für Flugsimulatoren verwendete oder Versuche durchgeführt wurden, Piloten beim Fliegen mit Head-Mounted-Displays dabei zu unterstützen, den Überblick über die zahlreichen Instrumente zu behalten. Ein weiteres Ziel war es, VR in der Telerobotik einzusetzen, um Roboter z.B. bei der Reparatur von Satelliten zu steuern. Auch das Potential für die Medizin wurde nach Bormann (1994) früh erkannt. Hier wurde VR sowohl zur Visualisierung als auch in Form von Telerobotik verwendet. So lassen sich mit Hilfe von VR komplexe volumetrische Daten, wie sie z.B. mit Kernspintomographen aquiriert werden, übersichtlich darstellen. Bei der minimal-invasiven Chirurgie soll die präzise Steuerung von Instrumenten im Raum ermöglicht werden. Ein weiteres Einsatzgebiet in der Medizin ist das Trainieren von Operationen und die Operationsplanung, da diese nicht am realen Körper durchgeführt werden kann. Erste Versuche zur Verwendung der erweiterten Realität (Augmented Reality) wurden in der Medizin mit Hilfe von Durchsicht HMD's durchgeführt, welche es erlauben, den Patienten und ein virtuelles Modell der ihn betreffenden 3D-Daten miteinander zu überlagern.

Im technisch-wissenschaftlichen Umfeld werden immersive Visualisierungssysteme hauptsächlich eingesetzt, um 3D-Daten darzustellen, wie die Ausgabe von Strömungssimulationen oder Informationen in Kontrollräumen. Im technischen Umfeld, zum Beispiel in der Architektur, läßt sich VR einsetzen, um die Verständigung zwischen den Auftraggebern, welche oft keinen technischen Hintergrund haben, und

den Konstrukteuren zu erleichtern und um Mißverständnisse zwischen den beiden Parteien möglichst früh auszuräumen. In Dai (1997) werden mehrere Anwendungen beschrieben, welche im industriellen Umfeld, meist in der Autoindustrie, VR verwenden, um Studien zur Produktion durchzuführen, bevor bereits das eigentliche Produkt real erstellt wird.

Die Verwendung von immersiven Visualisierungssystemen in den Geowissenschaften fand erst relativ spät statt und dann vor allem in der Erdölindustrie. Einer der ersten Artikel, der zu diesem Thema veröffentlicht wurde, stammt von Williams (1994). Dieser hat untersucht, ob die Verwendung von VR für die Visualisierung von Daten aus der Reservoir-Simulation als Ersatz oder Ergänzung für Standard-Workstation-Systeme sinnvoll ist. Als Hardware kamen eine SGI Onyx RealityEngine2 und ein CRT-basiertes Head-Mounted-Display mit einer Auflösung von 400 x 200 Pixel zum Einsatz. Die Bewegung des Benutzers wurde vom System verfolgt und ihm stand ein 3D-Zeiger zum Arbeiten im Raum zur Verfügung. Die Visualisierung wurde mit einer selbst entwickelten Software durchgeführt und mit der von Standard-Workstation-Arbeitsplätzen verglichen. Probleme traten mit "Motion Sickness" und der Ermüdung der Augen auf, der zu niedrigen Auflösung, welche keine Darstellung von Details erlaubte, und der Begrenzung der möglichen Modellkomplexität, damit die erforderliche Bildrate von 15 Bildern/sek gehalten werden konnte. Aus diesem Grund hielt Williams damals die Arbeit mit der vorgestellten Ausrüstung für weniger effektiv als die Arbeit an einem Standard-Bildschirm-Arbeitsplatz.

Am Lawrence Berkeley National Laboratory wurde in Zusammenarbeit mit BP-Exploration und Landmark Graphics eine Software entwickelt, um Reservoir-Ingenieure bei der Planung von Bohrlöchern zu unterstützen (Bethel et al., 1996). Hierzu wurde ein Programm zur Reservoir-Simulation mit der Visualisierungssoftware AVS (Advanced Visual Systems, 2001) gekoppelt, so daß die Simulationsausgaben unmittelbar in die Darstellung übernommen wurden. Die Visualisierung erfolgte in Stereo auf einer Desktop-Workstation unter Einbindung von 3D-Eingabegeräten (Stylus, Spaceball), um eine interaktive Reservoirplanung zu ermöglichen. Der Benutzer konnte seine Blickrichtung bestimmen, die Bohrlöcher frei im Raum positionieren und so die Visualisierung von Strömungslinien steuern. Da AVS seit neuestem auch als Multipipe-Version erhältlich ist, müßte das erstellte Programm jetzt auch in der CAVE ausführbar sein und voll immersives Arbeiten erlauben.

Der australische Konzern BHP hat in Zusammenarbeit mit dem CSIRO For-

schungszentrum in Canberra/Australien eine Demonstrationssoftware für die Interpretation seismischer Daten auf der "Haptic Workbench" entwickelt (Nesbitt et al., 1997; McLaughlin & Orenstein, 1997), die dem im Kapitel über Hardware vorgestellten Reachin-System sehr ähnlich ist. Mit Hilfe dieser Software sollten Geowissenschaftlern die Konzepte der haptischen und multimodalen Interaktion vermittelt werden, um danach mit ihnen das Potential der "Haptic Workbench" für die Aufgaben in ihren Bereichen zu diskutieren. Die Software visualisiert Schnitte durch einen seismischen 3D-Datensatz, während das Phantom die Bewegungsfreiheit entlang dieser Schnitte einschränkt. So wurde z.B. der Bewegung entlang von Reflektoren weniger Kraft entgegengesetzt als senkrecht zu ihnen, was eine Unterstützung bei der Digitalisierung der Reflektoren darstellt und Verwerfungen als Barrieren fühlbar macht. Zusätzlich lassen sich Markierungen im Datensatz plazieren und zu Dreiecken verbinden, so daß das System sich nutzen läßt, um aufbauend auf den Datensatz ein einfaches geometrisches Modell zu konstruieren. Eine weitere Arbeit am CSIRO beschäftigt sich mit der Planung von Bergwerken auf der Haptic-Workbench (Stevenson & Bogsanyi, 2000). Bei dieser Arbeit können die Schächte frei im Raum geplant werden. Das Phantom schränkt aber unter vorgegebenen Randbedingungen die Bewegungsfreiheit ein. So können zum Beispiel neue Schächte nur in einem bestimmten Abstand von anderen Anlagen geplant werden oder die mögliche Krümmung ist vorgegeben.

Hybertsen (1998) beschreibt die Visualisierungsinitiative der Firma Statoil. Hier wurden zuerst in einem Visualisation-Center mehrere große Bildschirme nebeneinander verwendet, auf denen jeweils Daten (Seismik, Geologische Daten, Daten des Reservoirmodells) abgebildet waren. Auf diese Weise waren die Daten auf einen Blick sichtbar, sie waren aber noch nicht miteinander integriert. Zusammen mit dem Virtual Environment Technology Laboratory (VETL) der Universität Houston wurde zu Demonstrationszwecken eine erste Software entwickelt, die seismische Daten, interpretierte Horizonte, Bohrlochdaten und das Gitter für die Reservoir-Simulation darstellen kann. Mit dieser Software wurde die Visualisierung in Stereo und unter Einbindung von Tracking auf die Wand projiziert und 600 Mitarbeitern vorgeführt, welche laut Hybertsen (1998) überwiegend der Auffassung waren, daß die Verwendung dieser Technologie Vorteile bringt. Mittlerweile betreibt Statoil zwei zylindrische Projektionen, eine in Stavanger und eine weitere in Trondheim.

Ab den Jahren 1999/2000 wurden von den Erdölfirmen in zunehmenden Maße Artikel veröffentlicht, welche ihre Aktivitäten bei der Forschung für die Anwendung von VR im Bereich Aufschluß und Gewinnung beschreiben. So wurde sowohl bei

Norsk-Hydro eine CAVE installiert und mangels kommerzieller Software die notwendige Software entwickelt, um Explorationsdaten in der CAVE zu untersuchen (Midttun et al., 2000), als auch bei der Firma ARCO (Stark et al., 2000a; Stark et al., 2000b). Die erste CAVE bei ARCO wurde 1997/98 errichtet. Sie wurde zunächst hauptsächlich für Forschung und Entwicklung genutzt und um das Potential der Technologie zu demonstrieren. In zunehmenden Maße und ab 1999 überwiegend wurde diese dann kommerziell eingesetzt zur Interpretation, Visualisierung von Volumendaten, Bohrlochplanung und bei Zusammenarbeit mit ARCO's Partnern. 1999 wurde dann in Zusammenarbeit mit der Firma MechDyne eine CAVE entwickelt, bei der sich die Seitenwände ausklappen lassen, so daß diese sich auch als Visionarium mit Rückprojektion nutzen läßt (MD-Flex Projektionssystem). Im Bereich Software-Entwicklung arbeitet ARCO mit dem Gocad Team zusammen. Die CAVE bei Norsk Hydro wurde zum Ende 1997 installiert, und es wurde in Eigenregie von Grund auf eine Software entwickelt, die an die speziellen Bedürfnisse bei der Verwendung von VR angepaßt ist. Diese erlaubt die Bedienung von der virtuellen Umgebung aus, so daß kein externes Operating notwendig ist, und stellt Seismik, interpretierte Verwerfungen und Horizonte, Bohrlöcher samt Daten und die Reservoirmodelle dar.

Für das Programm Gocad, das in den Geowissenschaften für die geometrische Modellierung sehr verbreitet ist, wurde eine Erweiterung entwickelt, die es ermöglicht, Szenen im Multipipe-Betrieb in Stereo darzustellen und so immersive Visualisierungssysteme für die Ausgabe zu nutzen. Zur Interaktion wurde das Menü von Gocad auf ein virtuelles Tablett in der virtuellen Umgebung abgebildet. Auf diese Weise läßt sich die Funktionalität, die Gocad für den Desktop-Bereich hat, auch in der virtuellen Umgebung nutzen (Gocad, 2001).

Die in der Erdölindustrie am häufigsten eingesetzten großformatigen Visualisierungssysteme außerhalb der Forschungs- und Entwicklungsabteilungen sind bisher Visionarien und Walls, welche mittlerweile bei einer Vielzahl von Erdölfirmen eingesetzt werden, um ihren Asset-Teams eine effektivere Zusammenarbeit zu ermöglichen. So läßt Shell von der Firma Trimension neun Visionarien errichten, welche rund um die Welt verteilt liegen und zum einen für die Interpretation genutzt werden sollen, zum anderen um geographisch verteilten Teams das Zusammenarbeiten zu ermöglichen (Real Time Graphics, 2001a). Bei Texaco wird schon seit längerem ein Visionarium genutzt, um die Volumendaten, die bei der Verwendung von 3D-Seismik entstehen, zu analysieren (Sheffield et al., 1999). Die Firma Mechdyne hat drei Visualisierungssysteme bei der venezuelanischen Erdölfirma Petrolera Ameriven installiert. Eines dieser Systeme wird in einem zentralen Visualisierungslabor



errichtet, in dem Experten für spezielle Fragestellungen arbeiten. Die beiden anderen Systeme werden direkt an den Bohrlokalationen eingesetzt und benutzt, um die Bohrlochdaten, die im Logging-Wire-Drilling Verfahren akquiriert werden, zu visualisieren. Diese Daten werden direkt per Satellit an das zentrale Visualisierungslabor übermittelt, so daß die Experten im Zentrallabor mit den Mitarbeitern vor Ort zusammenarbeiten können (Real Time Graphics, 2001b).

Um die Forschung auf dem Gebiet der Anwendung von virtuellen Umgebungen in der Erdölexploration voranzutreiben, wurde 1998 von 10 Firmen aus dem Erdöl- und Erdöl-service-Sektor, dem Virtual Environment Technology Laboratory (VETL) der Universität Houston und dem Institut für Medienkommunikation der Gesellschaft für Mathematik und Datenverarbeitung in Sankt Augustin (GMD, jetzt Teil der Fraunhofer Gesellschaft) ein Konsortium gegründet. Im Rahmen dieses Konsortiums wurde an der GMD unter der Leitung von Dr. Bernd Fröhlich bis Mitte 1999 eine Demonstrationssoftware entwickelt (Fröhlich et al., 1999; Zehner et al., 1999), welche auf dem SGI Visualization Summit 1999 in Galveston/Texas zusammen mit der Firma TAN Projektionstechnologie GmbH & Co. KG in einer CAVE, in einem Visionarium und auf einer Holobench präsentiert wurde. Parallel hierzu wurde am VETL ein Demonstrator entwickelt, der in Harding & Loftin (2000) beschrieben wird. Ferner wurde am VETL eine Dissertation angefertigt, welche sich vertieft mit der Verwendung haptischer und akustischer Techniken bei der Exploration geowissenschaftlicher Daten beschäftigt (Harding, 2001). Die zunehmende Verbreitung von immersiver Visualisierung in der Erdölindustrie, die Erfahrungsberichte in den Artikeln und das positive Feedback, das sowohl das Institut für Medienkommunikation als auch das VETL für ihre Arbeiten erhalten haben, weist darauf hin, daß die Technologie ein großes Potential für diese Branche hat.

Außerhalb der Erdölbranche liegen hinsichtlich der Vorteile, welche die Verwendung von VR mit sich bringt, nur sehr wenige Untersuchungen vor. Dies ist wahrscheinlich darauf zurückzuführen, daß die Kosten für die notwendige Technik sehr hoch sind und die notwendige Software-Entwicklung sehr aufwendig und kompliziert ist. Die Arbeiten, die durchgeführt wurden, konzentrieren sich in der Regel auf die relativ kostengünstig zu beschaffenden Systeme, wie Stereo-Visualisierung auf Monitoren oder die Verwendung von kostengünstigen Head-Mounted-Displays.

In Opriessnig (1999) wird eine Software beschrieben, welche die Visualisierung von Simulationsergebnissen und Meßdaten beim Tunnelbau ermöglicht. Die Deformation des Tunnels und die Veränderung von anderen Größen, wie zum Beispiel

die Spannungen im Gestein, lassen sich dynamisch darstellen. Zusätzlich lassen sich beliebige Schnitte durch den Tunnel legen. Als Hardware kommt eine SGI Octane Workstation mit Shutterbrillen zum Einsatz und die Interaktion erfolgt mit Hilfe der Maus und des Spaceballs.

Im Rahmen des Projektes BAGIS (Baugeologisches Geotechnisches Informations System) an den Universitäten Hamburg und Clausthal wurde ein Informationssystem entwickelt, das die Visualisierung geologischer, baugeologischer und bautechnischer Informationen gestattet (Reik et al., 1999). Für die Darstellung von komplexen 3D-Daten kann zur Visualisierung ein Head-Mounted-Display verwendet und die Interaktion über Datenhandschuhe vorgenommen werden.

Von der technischen Ausführung her ähnlich zu dieser Arbeit sind am ehesten die Arbeiten, die am Institut für Geoinformatik der Universität Münster durchgeführt werden. Im Rahmen der SpatialVisC Initiative wird ein Barco-Projektionstisch (einseitige Workbench) mit Shutterbrillen und einem optischen Trackingsystem verwendet (Münster, 2001). Da für den Betrieb dieser Workbench nur ein Graphikausgang benötigt wird, läßt sie sich mit einer SGI Octane Workstation antreiben. Das System wird im Rahmen mehrerer Projekte eingesetzt, um zum Beispiel Geländemodelle, atmosphärische Daten oder einen dreidimensionalen Sedimentkörper darzustellen.

# Kapitel 6

## Beispielanwendungen

Um Untersuchungen zur Verwendung von virtuellen Umgebungen für die Exploration geophysikalischer Daten durchzuführen, wurden aufbauend auf AVANGO prototypische Applikationen für eine Reihe unterschiedlicher Datentypen entwickelt. Diese Anwendungen dienten dazu, einen Einblick zu bekommen, welche Möglichkeiten sich für die Verwendung dieser Visualisierungstechnik in den Geowissenschaften für verschiedene Aufgaben bieten. Ferner war es das Ziel, durch die Erstellung der Applikationen Erfahrung mit der Entwicklung von Anwendungen für solche Umgebungen aus der Sicht eines Geowissenschaftlers zu sammeln. Um den Aufwand für die Implementierung in einem vertretbaren Rahmen zu halten, können diese Anwendungen immer nur einen exemplarischen Charakter haben. Bei der Erstellung prototypischer Applikationen wurde soweit es ging versucht, ein möglichst breites Feld an verschiedenen Anwendungstypen zu realisieren. Die meisten Applikationen beziehen sich allerdings auf die Visualisierung der “reinen“ Daten zum Zwecke der Datenexploration. Bei diesen könnte aber jederzeit ergänzend ein geometrisches Modell aus einer Datei mit VRML, Inventor oder DXF-Format geladen werden, was dann die Validierung der Ursprungsdaten gegen dieses Modell ermöglicht. Für die Darstellung wurden “echte“ Datensätze verwendet, welche freundlicherweise von Mitarbeitern der Universitäten in Köln und Bonn, der BGR in Hannover und dem Forschungszentrum Jülich zur Verfügung gestellt wurden. In diesem Kapitel sollen die verschiedenen Datensätze und ihre Visualisierung beschrieben werden.

- Als Beispiel für eine Anwendung in der Erdölindustrie dient der Demonstrator des VRGeo-Projektes, das am Institut für Medienkommunikation von einer

größeren Arbeitsgruppe durchgeführt wurde. Der Demonstrator visualisiert seismische Daten, Bohrlochdaten und die von Erdölfirmer modellierten Horizonte.

- Um die geologische Struktur des Untergrundes ohne vorhergehende Interpretation sichtbar zu machen, wurde eine 3D-Georadarvermessung visualisiert, welche aus mehreren, in zwei orthogonalen Richtungen eingemessenen vertikalen Schnitten besteht. Zusätzlich wurde die Möglichkeit geschaffen, die in den Radargrammen sichtbaren Schichtgrenzen in der virtuellen Umgebung zu markieren und festzulegen. Daher kann diese Anwendung dazu dienen, die Interpretationsmöglichkeiten in einer virtuellen Umgebung zu demonstrieren.
- Um einen Eindruck zu bekommen, wie gut sich immersive Visualisierungsumgebungen für die visuelle Fusion verschiedener geophysikalischer Datensätze eignen, wurden verschiedene Messungen, die auf einem hydrogeologischen Versuchsfeld durchgeführt wurden, gemeinsam dargestellt. Dies sind die tomographische Bestimmung des elektrischen Widerstandes, Bohrlochmessungen ( $K_f$ -Wert und natürliche Strahlungsaktivität) und Bohrprotokolle.
- Als Beispiel für eine 4D-Darstellung (räumlich-zeitlich) dient die zeitabhängige Visualisierung der Herde von ca. 14.000 Mikrobeben, die bei der hydraulischen Stimulation eines geothermischen Reservoirs induziert wurden. Anhand der räumlich-zeitlichen Ausbreitung dieser Bebenherde sollen Aussagen über die Bewegung des injizierten Wassers im Untergrund gemacht werden.
- Um die Visualisierung von geometrischen Modellen zu gestatten, wurde ein Prototyp entwickelt, der solche Modelle aus Dateien einliest und einfaches interaktives Arbeiten gestattet. Dieser Software-Prototyp wurde so ausgelegt, daß er verteilte Visualisierung ermöglicht. Das heißt, daß zwei Benutzer jeweils an einer Workbench stehen, dasselbe Modell sehen und jeweils die Interaktionen des anderen mit dem Modell verfolgen können. Als Beispiel-Datensatz kam das geometrische Modell eines Bergwerkes zum Einsatz.

## 6.1 Das VRGeo-Projekt am Institut für Medienkommunikation

### 6.1.1 Übersicht und verwendete Daten

Das VRGeo-Projekt wurde am Fraunhofer Institut für Medienkommunikation unter der Leitung von Dr. Bernd Fröhlich durchgeführt und ist Teil eines Konsortiums von Erdölfirmen, Erdölservicefirmen, des Instituts für Medienkommunikation und des Virtual Environment Technology Laboratories (VETL) an der Universität Houston. Ziel für den ersten Zeitabschnitt des Projektes war die Entwicklung eines Software-Prototypen für die Visualisierung der in der Erdölindustrie typischen Daten. Mit Hilfe des Prototypen sollten dann verschiedene Reservoir-Datensätze visualisiert werden, um einen Einblick zu bekommen, welches Potential und welche Anwendungsmöglichkeiten sich für spezifische Fragestellungen innerhalb der Erdölindustrie ergeben. Der Anwendungsprototyp sollte die integrierte Darstellung der im Feld gemessenen Daten mit den interpretierten geologischen Modellen ermöglichen und so bei der Validierung der Interpretation gegen die ihr zugrunde liegenden Messungen helfen. Ein weiterer für die Erdölfirmen interessanter Schwerpunkt war die Planung von Bohrungen im dreidimensionalen Raum. Der im Rahmen des Projektes entwickelte Software-Prototyp soll in dieser Arbeit als Beispiel für die Anwendung von VR in der Erdölindustrie dienen, da er im wesentlichen den meisten anderen entwickelten Systemen ähnlich ist, was die Darstellung der Daten betrifft. Der Schwerpunkt des Autors innerhalb dieses Projektes lag auf der Visualisierung der Bohrlochdaten.

Im Bereich der Exploration und der Charakterisierung und des Managements von Reservoirs sind die 3D-Seismik und die Bohrlochmessungen mit Abstand die wichtigsten Methoden. Die 3D-Seismik-Daten wurden von den Erdölfirmen als 3D-Bild-Dateien oder als Stapel von horizontalen 2D-Bildern (time-slices) zur Verfügung gestellt. Die Bohrlochdaten lagen für jede Bohrung als ASCII-Tabelle vor, welche für eine hohe Anzahl an Meßpunkten entlang des Bohrlochs die horizontale Abweichung vom Ausgangspunkt der Bohrung, die Tiefe unter dem Ausgangspunkt und eine Reihe von verschiedenen physikalischen Parametern in dieser Tiefe enthielt. Die von den Erdölfirmen bereits interpretierten Verwerfungen und stratigrafischen Grenzflächen lagen als Geometrien vor und werden aus Dateien im VRML oder Open Inventor Format eingelesen.

Im Rahmen des Projektes wurden zwei Demonstrationsmodelle entwickelt. Das erste Modell stellt einen Ausschnitt aus dem Gullfaks-Reservoir dar, welches nordwestlich von Bergen (Norwegen) am Westrand des Viking-Grabens liegt. Die Strukturgeologie des Reservoirs wird in Fossen & Hesthammer (1998) beschrieben. Das Modell diente dazu, verschiedene Visualisierungsmöglichkeiten für 3D-Seismik (als Schnitte und als Volumen) und verschiedene Interaktionsmöglichkeiten, z.B. mit der Cubic-Mouse, zu zeigen. Die Daten stammen von der norwegischen Firma Statoil. Das zweite Modell verwendet Daten von der australischen Firma BHP Petroleum und diente eher dazu, die interaktive Exploration von Bohrlochdaten mit Hilfe akustischer und visueller Methoden aufzuzeigen. Wo das Reservoir liegt, von dem diese Daten stammen, ist im Rahmen des Projektes nicht bekannt. Da die Firma BHP Petroleum jedoch schon sehr früh aus dem Konsortium ausgetreten ist, wurde dieses Modell des Anwendungsprototypen bei den meisten Präsentationen des Projektes nicht weiter verwendet.

## 6.1.2 Visualisierung der verschiedenen Datentypen

### 3D-Seismik

Die Seismik wird zur Echtzeitvisualisierung in einer 3D-Textur gespeichert und der Signalstärke eine Farbtabelle zugeordnet, so daß z.B. stark negative Ausschläge rot erscheinen und stark positive blau. Abbildung 6.1 zeigt drei senkrecht zueinander stehende Schnitte durch einen auf diese Weise kolorierten seismischen Datensatz. Die Orientierung des Datensatzes im Raum ist an die Orientierung der Cubic-Mouse gebunden. Die Schnitte können in Echtzeit mit Hilfe der korrespondierenden Stäbe der Cubic-Mouse durch den Datensatz bewegt werden. Zusätzlich zu den am Koordinatensystem ausgerichteten Schnittebenen ist eine weitere Ebene frei positionierbar. Durch das Modell läßt sich eine "magic lens" bewegen, welche die Seismik in ihrem Inneren durch direkte Volumenvisualisierung darstellt, wie sie zum Beispiel in Kidd (1999) beschrieben wird (Abb. 6.2). Innerhalb des rechteckigen Bereiches sieht man nur die Extremwerte der Seismik, welche die Reflektoren kennzeichnen. Der dazwischen liegende Bereich bleibt transparent. Auf diese Weise gewinnt der Betrachter sehr schnell einen Eindruck vom räumlichen Verlauf der Grenzflächen.

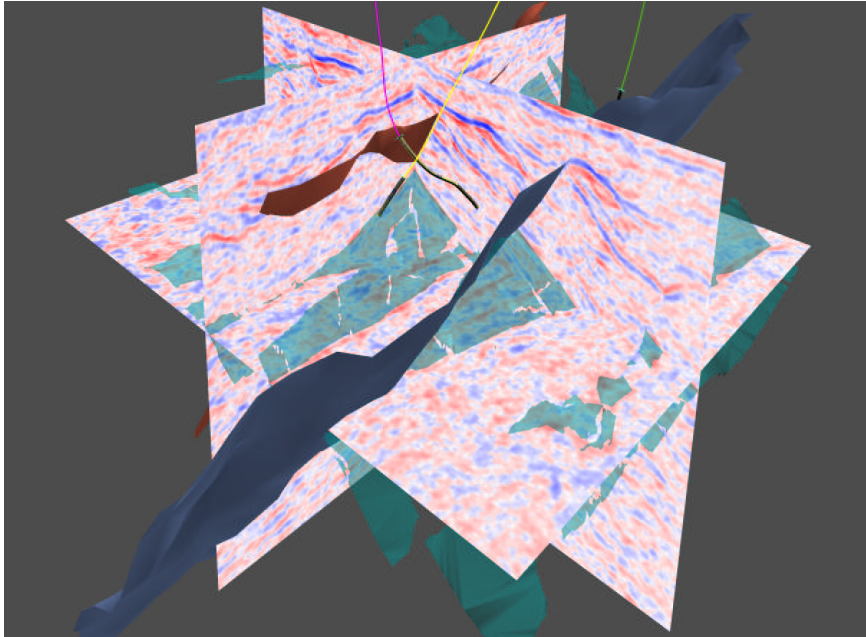


Abbildung 6.1: Visualisierung seismischer Daten mit drei senkrecht zueinander stehenden Schnittebenen, den sogenannten Crossline-, Inline- und Timeslices.

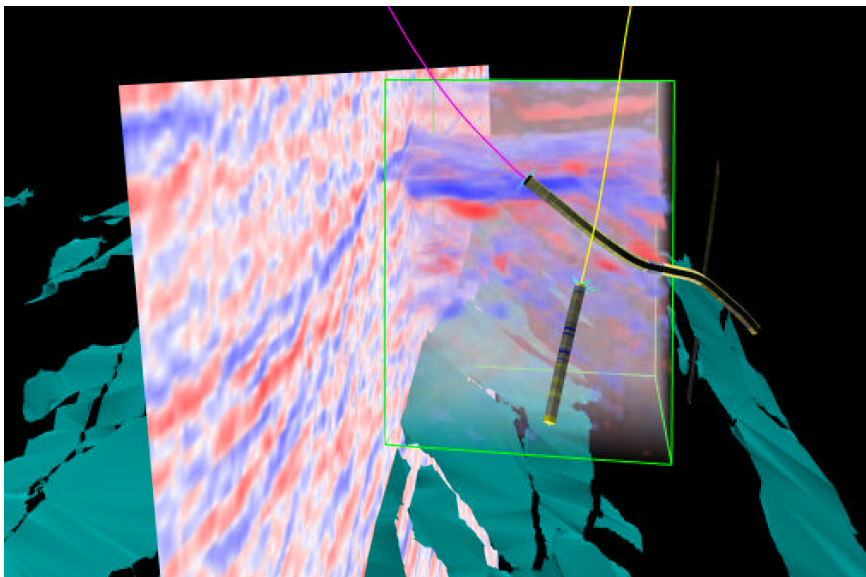


Abbildung 6.2: Die sogenannte Volume-Rendering-Lupe. Innerhalb eines würfelförmigen Bereiches erfolgt eine Volumendarstellung auf Basis der seismischen Daten. Man erkennt deutlich den Verlauf der Reflektoren im Untergrund. Im Vordergrund verlaufen zwei Bohrlöcher, auf denen farbkodierte Logdaten zu sehen sind.

## Bohrlochmessungen

In Bohrlöchern werden oft mehrere Messungen verschiedener petrophysikalischer Parameter durchgeführt, die dann entlang des Bohrloches in hoher Dichte vorliegen. Generell muß man bei deren Darstellung zwischen skalaren Daten (z.B. Dichte, elektrische Leitfähigkeit und natürliche Strahlungsaktivität des Gesteins), vektoriellen Daten (z.B. Streßrichtung, Einfallsrichtung von Strukturen) und flächenhaften Daten (z.B. Lage sedimentärer Flächen um das Bohrloch herum) unterscheiden. Vektorielle und flächenhafte Daten werden meist durch das Verarbeiten der Daten von bildgebenden Bohrlochmeßverfahren gewonnen.

In virtuellen Umgebungen wird der Verlauf des Bohrlochpfades als sechseckige Röhre dargestellt, deren Querschnittsflächen senkrecht zur Trajektorie stehen. Auf diese Weise lassen sich beliebige Bohrlochverläufe im Raum darstellen. Die Darstellung skalarer Daten erfolgt durch Farbkodierung entlang dieses Bohrlochpfades. Wie in Abbildung 6.3 zu sehen ist, lassen sich durch eine bivariate Farbkodierung zwei Parameter gleichzeitig darstellen. Im RGB-Farbmodell (siehe z.B. Foley et al., 1990) wird für die normalisierten Parameter jeweils die Farbintensität linear zwischen null für minimale Werte und eins für maximale Werte interpoliert. Ein Parameter korrespondiert hierbei mit dem Blauwert, der zweite mit dem Rot- und Grünwert (gelb). Bei linearer Korrelation der Meßwerte ergibt sich ein Grauton, sonst dominiert der Farbton des jeweils höheren Wertes. Zur Betrachtung einer Logkurve oder zur Darstellung eines weiteren Parameters besteht die Möglichkeit, ein neben dem Bohrloch befindliches kurzes Panel auf und ab zu bewegen. Seine Position wurde zuerst mit Hilfe des Zeigestrahls gesteuert, mit dem man auf das entsprechende Bohrloch zeigt. Da sich diese Art der Steuerung als zu ungenau erwies, erfolgte die Steuerung später mit Hilfe eines Potentiometers, mit dem die Geschwindigkeit der Bewegung entlang des Bohrlochs gesteuert wurde.

Um die Anwendung der bivariaten Farbtabelle zur Korrelation zweier Logs zu illustrieren, zeigt Abbildung 6.3 auf der linken Seite ein Lehrbuchbeispiel aus der Erdölgeologie für drei Bohrlochmessungen aus einem Kalksteinreservoir (Selley, 1998), links als klassischer Logplot und rechts als korrespondierende 3D-Visualisierung. Die hier dargestellten Parameter sind Gamma-Ray (G-API), als Kurve auf dem Panel zu sehen, Neutron-Porosity (NPHI), hier gelb gefärbt, und Gesamtdichte (RHOB), hier blau gefärbt. Der gelbe Bereich auf den Bohrlöchern entspricht Regionen mit hohen NPHI-Werten und niedrigen RHOB-Werten, was auf Gas hindeuten würde. Der Bereich darunter ist grau, da beide Parameter mittlere Werte aufweisen,



was entweder Öl oder Wasser anzeigt. In der schematischen Zeichnung ist ein Schnitt durch ein Reservoir mit variabel mächtiger Gaskappe dargestellt, welches mit den in der 3D-Visualisierung dargestellten Bohrlöchern korrespondieren könnte. Auf der rechten Seite der Abb. 6.3 ist eine mögliche Darstellung vektorieller und flächenartiger Daten zu sehen, welche mit Hilfe von Pfeilen und Flächen als Symbole entlang des Bohrloches erfolgt.

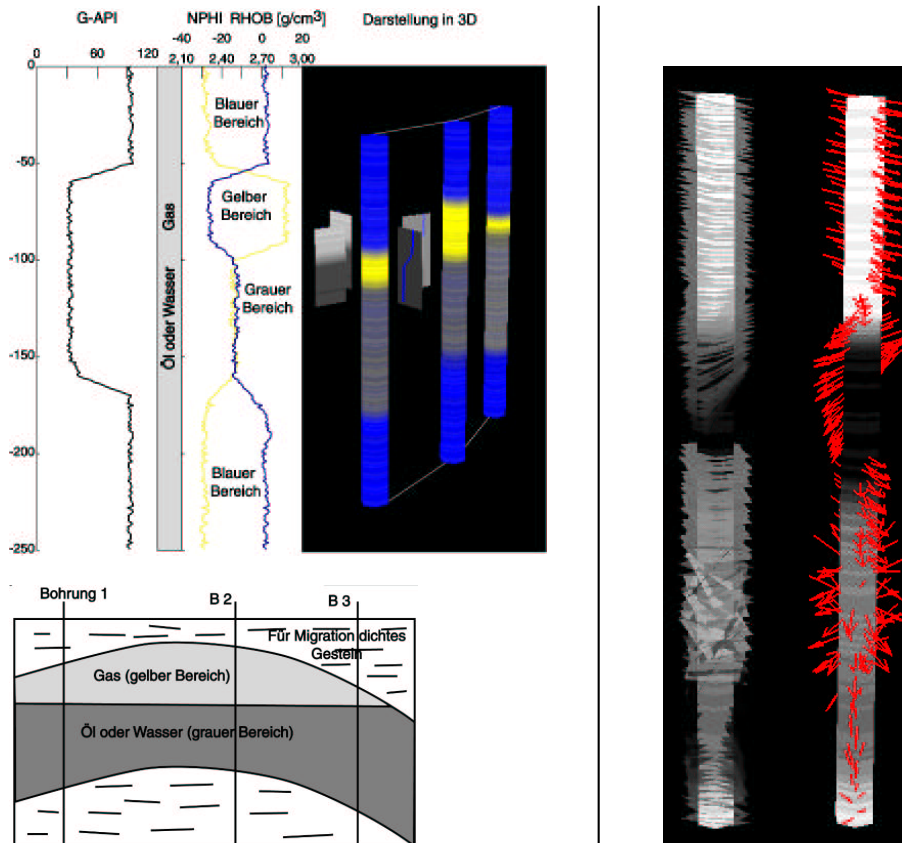


Abbildung 6.3: Logplot vs. 3D-Darstellung. Auf der linken Seite sieht man den klassischen Logplot mit Gamma-Ray (G-API), Neutron-Porosity (NPHI) und Dichte (RHOB). Rechts daneben die korrespondierende 3D-Visualisierung. NPHI ist gelb eingefärbt, und die Dichte blau. Der gelbe Bereich auf den Bohrlöchern entspricht den hohen NPHI-Werten und zeigt evtl. Gas an. Der Bereich darunter ist grau, da NPHI und RHOB beide mittlere Werte haben, was entweder Öl oder Wasser anzeigt. In der schematischen Zeichnung links unten ist die Mächtigkeitsveränderung des gas- bzw. fluidhaltigen Bereichs angezeigt. Diese ist auch in der 3D-Darstellung gut zu erkennen. Auf der rechten Seite ist eine mögliche Darstellung von Vektor- und Flächendaten zu sehen. Das Beispiel wurde aus Selley (1998) übernommen.

Um zusätzlich zu den beiden mit der Farbtabelle visualisierten Daten weitere

darstellen zu können, hat Dr. Stephen Barrass angekoppelt an die Visualisierung versucht akustische Techniken zu verwenden (Barrass & Zehner, 2000). Die Sonifizierung basiert auf dem Konzept des Geiger-Zählers und ist an das bewegliche Panel gekoppelt. Wird dieses in einem Bereich des Bohrloches bewegt, an dem niedrige Werte vorliegen, so ist ein tiefes langsames Klicken zu hören. Sind die Werte hoch, hört man ein hohes schnelles Klicken.

### **Geologischen Einheiten**

Die geologischen Störungen und stratigrafischen Horizonte werden als Dreiecksnetze dargestellt. Die Interaktion mit diesen Objekten erfolgt mit Hilfe von virtuellen Werkzeugen, mit denen man sie bewegen und rotieren kann, z.B. wenn sie im Weg sind, oder mit denen die Materialeigenschaften wie Farbe und Transparenz geändert werden kann. Die von den Erdölfirmen bereitgestellten interpretierten Horizonte waren sehr detailliert, so daß der Rechner sie nicht interaktiv darstellen konnte. Sie werden daher mit einer niedrigeren Auflösung gezeigt und der Benutzer hat die Möglichkeit temporär einzelne Teilbereiche in der ursprünglichen hohen Auflösung anzuzeigen.

### **6.1.3 Erfahrungen mit dem Software-Prototypen**

Der Software-Prototyp zeigt, daß sich virtuelle Umgebungen gut eignen, um die zugrunde liegenden Daten gegen die Interpretation zu validieren und um über den gesamten Datensatz zu diskutieren. Insbesondere die 3D-Seismik und die Interpretation eignen sich gut für eine Präsentation in einem solchen Visualisierungssystem. Ist das dargestellte Volumen sehr groß, so tritt die Darstellung der Bohrlöcher immer weiter in den Hintergrund und einzelne Features auf den Bohrlöchern sind nur noch schwer oder gar nicht mehr zu erkennen. Hier wäre eine Veränderung der Repräsentation, zum Beispiel anhand von Level-of-Detail (LOD) Techniken notwendig. Beide Prototypen wurden auf mehreren Steering-Treffen des Konsortiums und auf dem SGI Visualization Summit 1999 in Galveston/Texas vorgeführt. Das Feedback war jeweils positiv, obwohl nur ein Bruchteil der Funktionalität zur Verfügung steht, welche von den Standard-Desktop-Workstationanwendungen bereitgestellt werden. Die meisten kommerziellen Visualisierungssysteme aus der Erdölindustrie, wie sie auf dem SGI Visualization Summit vorgestellt wurden, ermöglichen zwar die großräumi-

ge Darstellung der Daten in virtuellen Umgebungen und damit das kooperative Arbeiten größerer Teams, müssen aber von einem Desktop aus gesteuert werden. Die Demonstrationssoftware des VRGeo-Projektes hob sich insofern von den anderen vorgestellten Anwendungen ab, als sie die direkte Interaktion mit den Objekten im Raum gestattet und somit das kooperative Arbeiten erleichtert.

Während der Steering-Treffen kristallisierten sich einige Schwerpunkte für weitere Forschungsmöglichkeiten heraus, an denen die Firmen Interesse haben. Dementsprechend liegen die weiteren Schwerpunkte des Projektes auf der Entwicklung einer verteilten Anwendung und vor allem auf der Entwicklung einer Software, die die Volumenvisualisierung auch mit sehr großen Datensätzen ermöglicht.



## 6.2 Visualisierung von Georadardaten

### 6.2.1 3D-Erkundung des Untergrundes mit Bodenradar

Um geologische Strukturen im flachen Untergrund zu erfassen, wird immer häufiger Georadar eingesetzt. Das Verarbeiten dieser Daten wird in der Regel mit Standard-Processing-Programmen für die Seismik durchgeführt. Für den zweidimensionalen Fall ist dies auch für die Interpretation und Visualisierung der Daten sehr gut möglich. Bei der Bestimmung dreidimensionaler Strukturen bekommt man hier jedoch Probleme, was an den unterschiedlichen Meß- und Processingverfahren liegt, die zu unterschiedlichen Datenformaten führen. In der Abbildung 6.4 sind drei mögliche Meßanordnungen aufgezeigt, die geeignet sind, mit dem Georadar geologische Strukturen im Untergrund zu erfassen.

Bei der 3D-Seismik führt die Datenverarbeitung zu einem 3D-Voxelgitter, welches in allen drei Raumdimensionen relativ dicht ist. Durch einen solchen Datensatz lassen sich beliebige Schnittebenen legen und direkte Volumenvisualisierung, wie sie zum Beispiel für 3D-Seismik in Kidd (1999) oder im Rahmen des VRGeo-Projektes am Fraunhofer Institut für Medienkommunikation (siehe vorherigen Abschnitt) beschrieben wird, eignet sich gut für eine übersichtliche Visualisierung. Ferner kann man auf solche Datensätze automatische Extraktionsmethoden anwenden, die auf der Konnektivität der einzelnen Voxel beruhen (z.B. Lohmann, 1998). Derartige Daten, die im Rahmen des VRGeo-Projektes als Volumen visualisiert wurden, hatten Dimensionen von 256 x 256 x 256 bis 512 x 512 x 512 Voxel.

Bei der Vermessung mit dem Georadar werden immer zweidimensionale Schnitte durch den Datenraum erfaßt und diese meist auch als solche interpretiert. Würden viele parallele Linien ausreichend dicht beieinander gemessen, so ließe sich auch mit dem Georadar ein entsprechend dichtes dreidimensionales Voxelgitter erzeugen (siehe Abbildung 6.4a). Die hierfür notwendige Anzahl an Radargrammen (mindestens 64 - 128) stellt jedoch meist einen zu hohen Aufwand dar. Statt dessen werden meist zwei senkrecht zueinander liegende Serien an Radargrammen gemessen, wie es in Abbildung 6.4b gezeigt wird. Auch Vermessungen wie die in Abbildung 6.4c dargestellte können sinnvoll sein. Die beim Messen einzelner Profile entstehenden Datensätze sind entlang der Profillinien sehr dicht, weisen aber dazwischen große Lücken auf. Daher können sie nicht zur Volumenvisualisierung verwendet werden. Die einzige Möglichkeit wäre, die Daten auf ein regelmäßiges Gitter zu interpolieren,

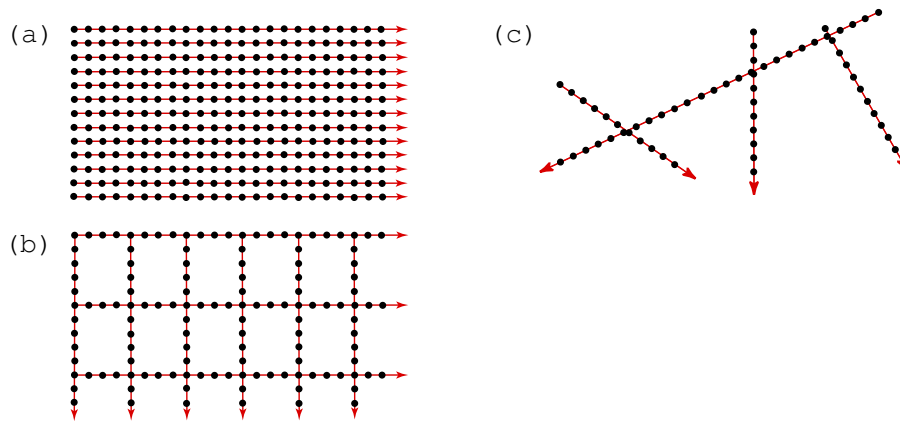


Abbildung 6.4: Skizze von drei verschiedenen Möglichkeiten, mit dem Georadar dreidimensionale Strukturen im Untergrund zu erfassen. Die Pfeile zeigen die gemessenen Linien an, die schwarzen Punkte die daraus resultierenden CMP's (Common Mid Points). Messungen vom Typ (a) ergeben ein dreidimensionales Voxelgitter, wie es auch bei der 3D-Seismik gewonnen wird, und lassen sich daher gut mittels Volumenvisualisierung darstellen. Für Messungen der Typen (b) und (c) müssen andere Lösungen gefunden werden.

wobei aber die Qualität des hieraus resultierenden Datensatzes schwer zu kontrollieren ist.

In diesem Beispiel soll mit Hilfe von VR-Technik versucht werden, exemplarisch eine Darstellungs- und Interpretationsmöglichkeit zu schaffen, welche es ermöglicht, die einzelnen Radargramme zu interpretieren, während man die umliegenden Schnitte als zusätzliche Information zur Verfügung hat. Die Darstellung sollte hierbei unabhängig von der Anlage des Meßnetzes sein. Der für den Versuch verwendete Datensatz wurde von Carsten Alteköster (Geologisches Institut der Universität Bonn) auf einem Sportplatz in Lägerdorf/Norddeutschland gemessen. Die Datenaufnahme erfolgte mit einem GSSI SIR-10+ Bodenradar in Common-Offset-Technik. Das Prozessieren erfolgte mit Hilfe der REFLEXW-Software von Sandmeier Scientific Software (2001). Die Untersuchung wurde durchgeführt, da sich auf dem Sportplatz ein trichterförmiger Hohlraum von ca. 1 m Durchmesser gebildet hatte. Zum einen sollte die Ursache hierfür geklärt werden, und zum anderen sollte sichergestellt werden, daß sich nicht noch weitere Hohlräume im Untergrund befinden. Die Vermessung erfolgte auf einer 9 m x 9 m großen Fläche, wobei zwei senkrecht zueinander gelegte Serien von parallelen Profilen mit 0,5 m Abstand gemessen wurden. In allen Radargrammen war die obere Grenze des sich hier in geringer Tiefe befin-

denden Kalkgesteins gut zu erkennen. Diese wurde von Alteköster mit REFLEXW auf allen Schnitten interpretiert und mit Hilfe des Programmes SURFER (Scientific Software Group, 2001) als Konturplot dargestellt. Wie in der Abbildung 6.5 zu sehen ist, zeigt die Obergrenze des Kalksteins im Norden eine talförmige Einbuchtung, die nach Süden hin abflacht.

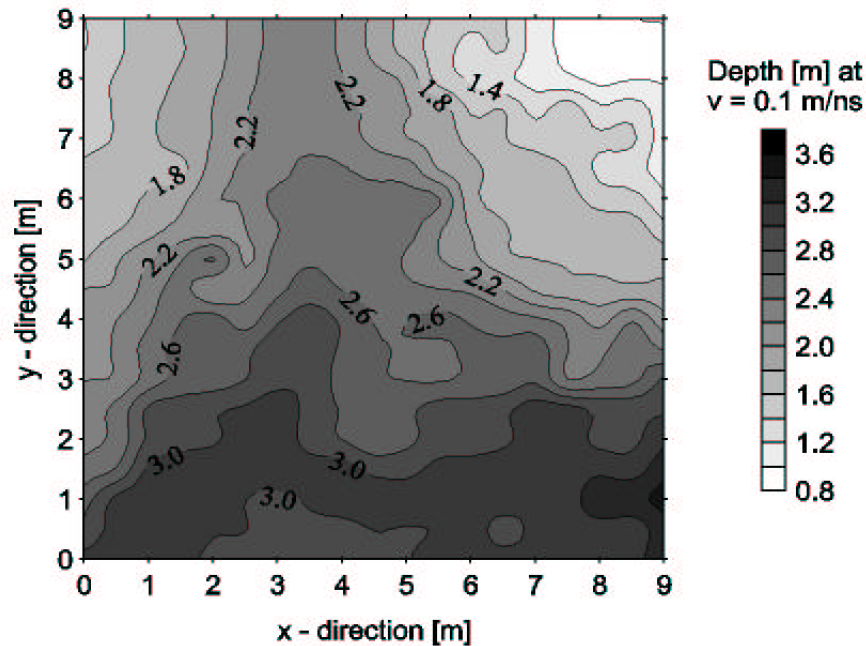


Abbildung 6.5: Konturplot des in den Radargrammen sichtbaren Reflektors, so wie er aus den beiden senkrecht zueinander gemessenen Serien von Schnitten bestimmt wurde (Abstand zwischen den einzelnen Linien war 0,5 m). Nach Alteköster (2000), persönliche Mitteilung.

### 6.2.2 Eine Bodenradar-Vermessung als interaktives virtuelles Modell

Da die Radargramme zweidimensionale Schnitte durch den Datenraum darstellen, werden sie in Form eines sogenannten Fence-Diagramms als im Raum stehende Bilder visualisiert. Die Umwandlung vom Radar- bzw. Seismik-Datenformat in 8bit-Graustufenbilder im IRIS-RGB-Format erfolgt mit Hilfe eines Zusatzprogrammes, welches für das Seismik-Unix Programmpaket von Cohen & Stockwell (1999) geschrieben wurde. Hierbei wird zuerst die nach dem Betrag maximale Amplitude

ermittelt und dann der Bereich zwischen maximaler negativer und maximaler positiver Amplitude auf den Bereich  $[0:255]$  abgebildet. Die negativen Amplituden werden dabei in schwarz dargestellt und die positiven in weiß. Bei der Verwendung von Bildern in 3D-Graphikbibliotheken sollte die Größe der Bilder nach Möglichkeit in allen drei Raumrichtungen eine ganzzahlige Zweierpotenz betragen. Die notwendige Umwandlung wurde mit der ImageVision Library von Silicon Graphics durchgeführt.

In einer virtuellen Umgebung wird der Verlauf aller Messungen durch Linien angezeigt. Am Ende jeder Linie befinden sich "Griffe", mit deren Hilfe die Visualisierung der einzelnen Radargramme ein- und ausgeschaltet werden kann. Diese sind im ausgeschalteten Zustand grün und im angeschalteten Zustand rot, was die Orientierung erleichtert, wenn, wie z.B. oft am Bildschirm, keine Stereodarstellung möglich ist. Abbildung 6.6 zeigt die Darstellung einiger Radargramme als klassisches Graustufenbild. Bei gleichzeitiger Betrachtung mehrerer Schnitte verdecken sich diese gegenseitig, weswegen der dreidimensionale Zusammenhang der Daten auf diese Weise nicht vollständig zu sehen ist. Daher wurde eine weitere Form der Visualisierung implementiert, die nur die starken Reflektoren zeigt. Hierzu wird auf das Bild eine Farbtabelle angewendet, wie sie in Abbildung 6.7 zu sehen ist. Die niedrigen Grauwerte (entsprechen starken negativen Amplituden) werden hierbei auf die Farbe Blau abgebildet, die hohen (entsprechen starken positiven Amplituden) auf die Farbe Rot. Der mittlere Bereich der Amplituden (mittlere Grauwerte) wird als durchsichtig behandelt. Dies ermöglicht es, den Verlauf eines Reflektors auch auf hintereinander liegenden Schnitten zu erfassen. Die Cut-Off Werte, die bestimmen, welche Bildbereiche als Reflektor gelten und welche nicht, lassen sich auch während der Laufzeit spezifizieren. Mit Hilfe eines zweiten "Griffes", der sich jeweils an den Linienenden befindet, kann zwischen den beiden Visualisierungsmodi hin- und hergeschaltet werden.

Um nicht nur die Visualisierung, sondern auch die Interpretation von Bodenradar-Daten in einer virtuellen Umgebung zu testen, wurde ein sehr rudimentärer Auto-tracker implementiert. Dieser wird ausgelöst, indem man mit dem Strahl des entsprechenden Werkzeuges den Reflektor auf einem Radargramm auswählt. Handelt es sich bei dem Schnittpunkt um den Teil eines Reflektors, so wird automatisch dessen Fortsetzung gesucht. Wie in Abbildung 6.8 zu sehen ist, wird hierzu in Abhängigkeit von den gesetzten Grenzwerten aus der Farbtabelle nach oben und nach unten evaluiert, wie weit sich der Reflektor in der Vertikalen erstreckt. Hiernach wird jeweils, ausgehend von den benachbarten Bildpunkten in der vorigen und der nächsten Spalte, die gleiche Suche durchgeführt. Der Reflektor selber wird jeweils als genau



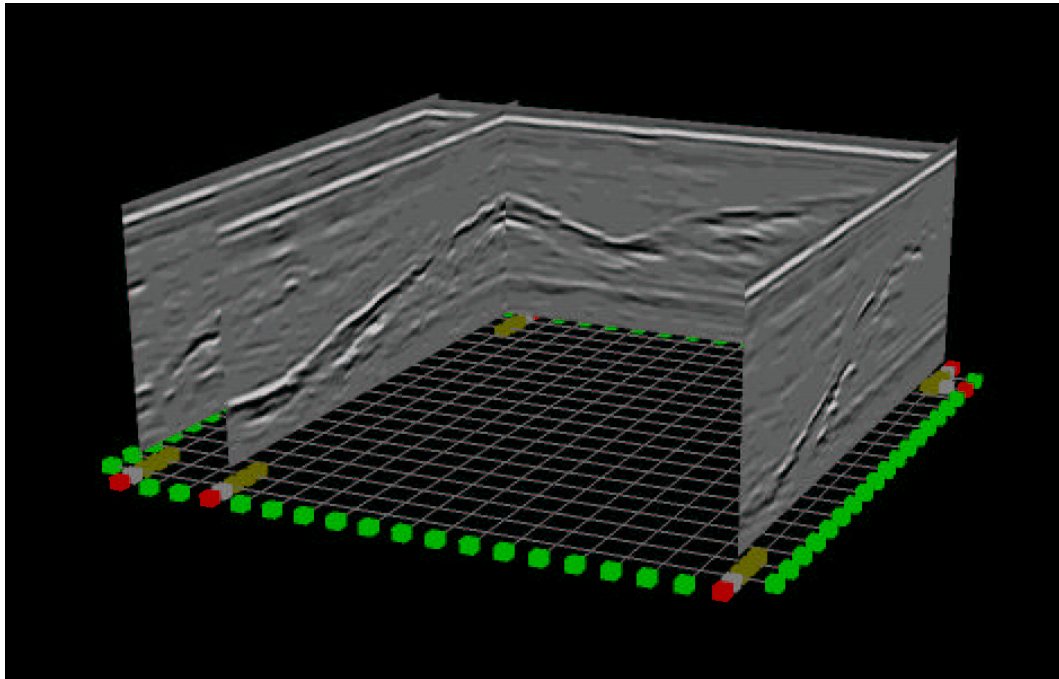


Abbildung 6.6: Standard-Darstellung von vier Radargrammen als sogenanntes Fence-Diagramm. Das Liniengitter auf der Unterseite zeigt den Verlauf der gemessenen Schnitte an. Mit Hilfe der “Griffe“ an den Enden können diese jeweils an- und ausgeschaltet werden. Im angeschalteten Zustand erscheint ein zweiter Griff (hier weiß), mit dem zwischen zwei verschiedenen Darstellungsmodi hin und hergewechselt werden kann.

in der vertikalen Mitte verlaufend angenommen. Diese Methode liefert, wie auch in der Abbildung zu sehen, nicht den exakten Verlauf, wie man ihn von Hand zeichnen würde. Sie ist jedoch für Versuchszwecke ausreichend und stellt durch die Kopplung mit der Farbtabelle eine Art “What You See Is What You Get“-System dar. Die gefundenen Reflektorverläufe werden an ein weiteres Objekt übergeben, das sie speichert und sofort als Linien darstellt. Jede gefundene Linie hat hierbei einen Griff, der benutzt werden kann, um sie auch wieder zu löschen. Alle gefundenen Linien können als Gocad-Datei abgespeichert werden. Auf diese Weise besteht die Möglichkeit, den Reflektor mit Gocad anschließend als Dreiecksnetz zu konstruieren.

Um die gesamten Messungen zu betrachten, werden alle Radargramme sichtbar gemacht und die Visualisierung in den Modus geschaltet, in dem nur die Reflektoren zu sehen sind. Eine Gesamtansicht des Modells ist in Abbildung 6.9 zu sehen. Der Verlauf des Reflektors ist klar erkennbar, so wie er auch schon aus dem Kon-

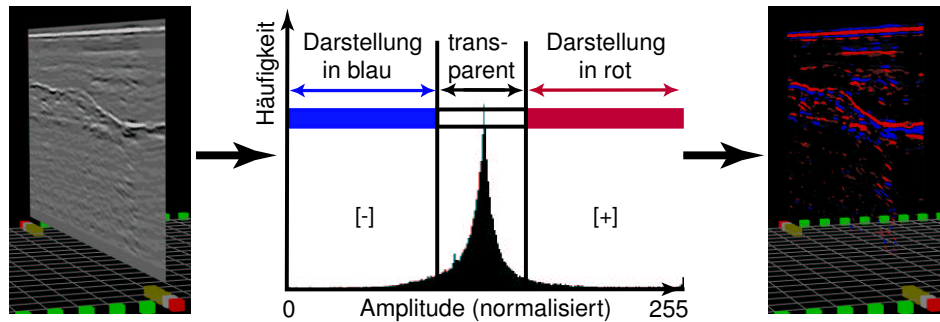


Abbildung 6.7: Anwendung einer Farbtabelle (in der Mitte) auf die Graustufendarstellung (links) welche die Sichtbarkeit der Reflektoren im Raum verbessert (siehe rechts). Die starken positiven Amplituden werden in rot dargestellt, die stark negativen in blau. Der Rest des Radargramms ist transparent.

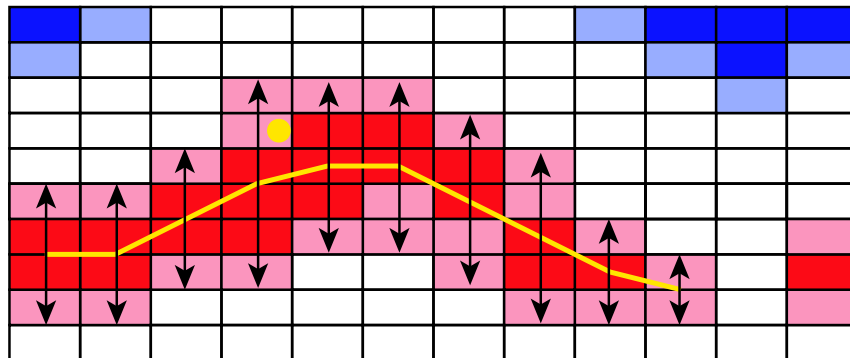


Abbildung 6.8: Funktionsweise des Autotrackers. Der gelbe Kreis zeigt den Punkt, von welchem die Suche startet; die einzelnen Kästchen repräsentieren jeweils Texel (Bildpunkte) im Radargramm. Die gelbe Linie zeigt den sich ergebenden Verlauf des Reflektors an. Weitere Erklärungen im Text.

turplot der Abbildung 6.5 zu ersehen ist. Zur Manipulation stehen dem Betrachter drei Werkzeuge zur Verfügung. Ein Werkzeug, um das Modell zu rotieren, ein Werkzeug zur Transformation und ein Werkzeug zum Auswählen der Reflektoren. Zum Auswechseln der Werkzeuge wurde der von Wesche & Droske (2000) implementierte virtuelle Werkzeugkasten verwendet. Ist kein Werkzeug ausgewählt, erscheint dieser vor dem Stylus, sobald mit dem Zeigestrahl ein Objekt des Modells getroffen wird.

Mit gedrücktem Knopf kann dann durch die Drehung des Stylus das entsprechende Werkzeug ausgewählt werden.

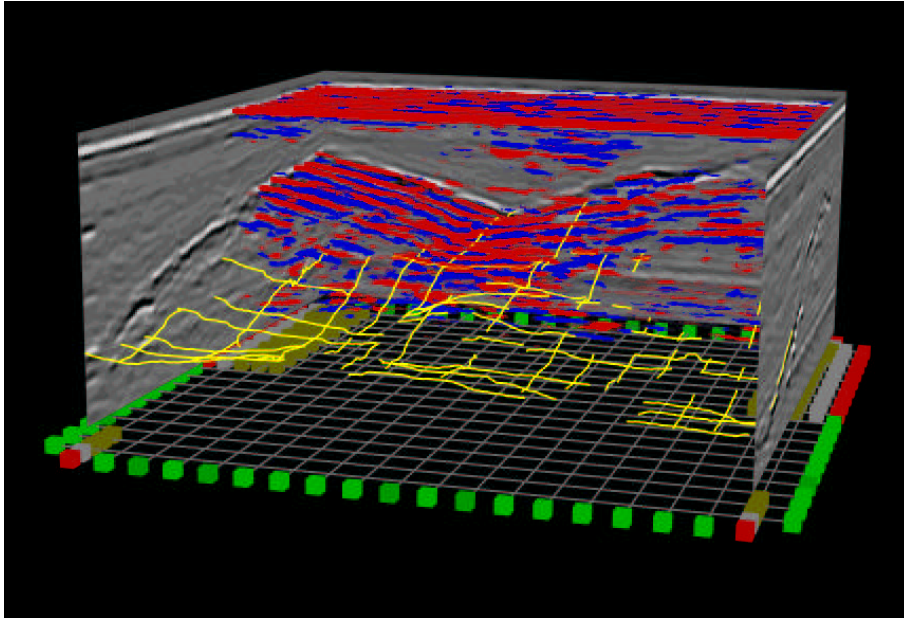


Abbildung 6.9: Darstellung des gesamten Datensatzes. Die gelben Linien stellen bereits interpretierte Reflektoren dar.

Obwohl das vorgestellte Interpretationssystem nur eine recht rudimentäre Funktionalität aufweist, kann man es benutzen, um jeweils in der CyberStage, auf der Workbench und am Monitor auszuprobieren, wie die Interpretation in einer solchen Umgebung aussehen könnte, um so ein Gefühl dafür zu entwickeln, wie praktikabel die Verwendung eines Standard-VR-Systems für derartige Aufgaben ist.

Da das An- und Umschalten aller einzelnen Schnitte im Testbetrieb als umständlich empfunden wurde, wurde ein weiteres System implementiert, das alle Radargramme gleichzeitig darstellt. Mit Hilfe einer zu den Achsen parallelen Box, wie sie in Kapitel 4 beschrieben wurde, läßt sich ein Subvolumen spezifizieren, in dem die Schnitte sichtbar sind. Die “Griffe“ können nach wie vor dazu verwendet werden, um zwischen den beiden Visualisierungsmodi hin- und herzuschalten. Dieser Visualisierungsprototyp implementiert allerdings keinen Autotracker.

### 6.2.3 Erfahrung mit der Verwendung des Systems

Der vorgestellte Prototyp wurde zum Test exemplarisch für die Interpretation am Bildschirm auf der Workbench und in der CyberStage eingesetzt. In der CyberStage wurde er zudem noch häufig als Beispiel verwendet, um Geowissenschaftlern, die am Institut für Medienkommunikation zu Besuch waren, diese Arbeitsumgebung vorzuführen. Die räumlichen Zusammenhänge der Daten, hier also der Verlauf der Kalkgesteins-Obergrenze, ist gut zu sehen. Die Sichtbarkeit ist aufgrund der Stereo-Visualisierung wesentlich unabhängiger vom Standpunkt des Betrachters als am Bildschirm. Dies wird in der Abbildung 6.10 noch einmal verdeutlicht, wo dieselben drei Radargramme aus verschiedenen Perspektiven dargestellt werden. Jedoch kann man den Verlauf des Reflektors auch am Bildschirm recht gut sehen, so daß die Verwendung der heute noch sehr teuren VR-Technik für einen Datensatz dieses Umfangs eigentlich übertrieben erscheint. Es wird aber auch deutlich, daß man mit dieser Technik wesentlich komplexere Datensätze übersichtlich darstellen kann als am Bildschirm.

Um die eigentliche Interpretation durchzuführen, daß heißt Reflektoren auszuwählen und zu markieren, scheinen die immersiven Visualisierungssysteme in der standardisierten Form, wie sie uns zur Verfügung standen, nicht so gut geeignet zu sein. Obwohl der räumliche Eindruck am Bildschirm fehlt und daher der Bezug zu den benachbarten und kreuzenden Radargrammen eingeschränkt war, schien die Interpretation am Bildschirm am praktikabelsten und angenehmsten zu sein. Am Bildschirm erfolgt das Markieren eines Reflektors, indem die Maus einfach vor diesem plaziert und dann der Autotracker ausgelöst wird. Hierbei wird der Zeigestrahle als senkrecht zur Bildschirmenebene angenommen, so daß keine echte 3D-Interaktion notwendig ist. Diese Aufgabe läßt sich mit Hilfe der Maus als 2D-Interaktionsgerät wesentlich schneller und präziser ausführen. In den virtuellen Umgebungen hat man hingegen Probleme mit der Zeitverzögerung der Interaktion. Ferner muß man den Zeigestrahle aus der freien Hand in allen drei Dimensionen steuern, was eine sehr ruhige Hand erfordert. Auch das in Kapitel 4 bereits erwähnte Problem, daß die Augen nur die Hand mit dem Stylus oder den (virtuellen) Reflektor scharfstellen können, macht sich hier bemerkbar, da man als Benutzer versucht ist, mit dem Zeiger unmittelbar auf dem Radargramm zu operieren, um den Zeigestrahle möglichst kurz zu halten. Dies ist in der CyberStage besonders kritisch, da die Entfernungsdiskrepanzen hier höher sind.

Da man am Bildschirm nicht den räumlichen Eindruck hat, ist man darauf an-

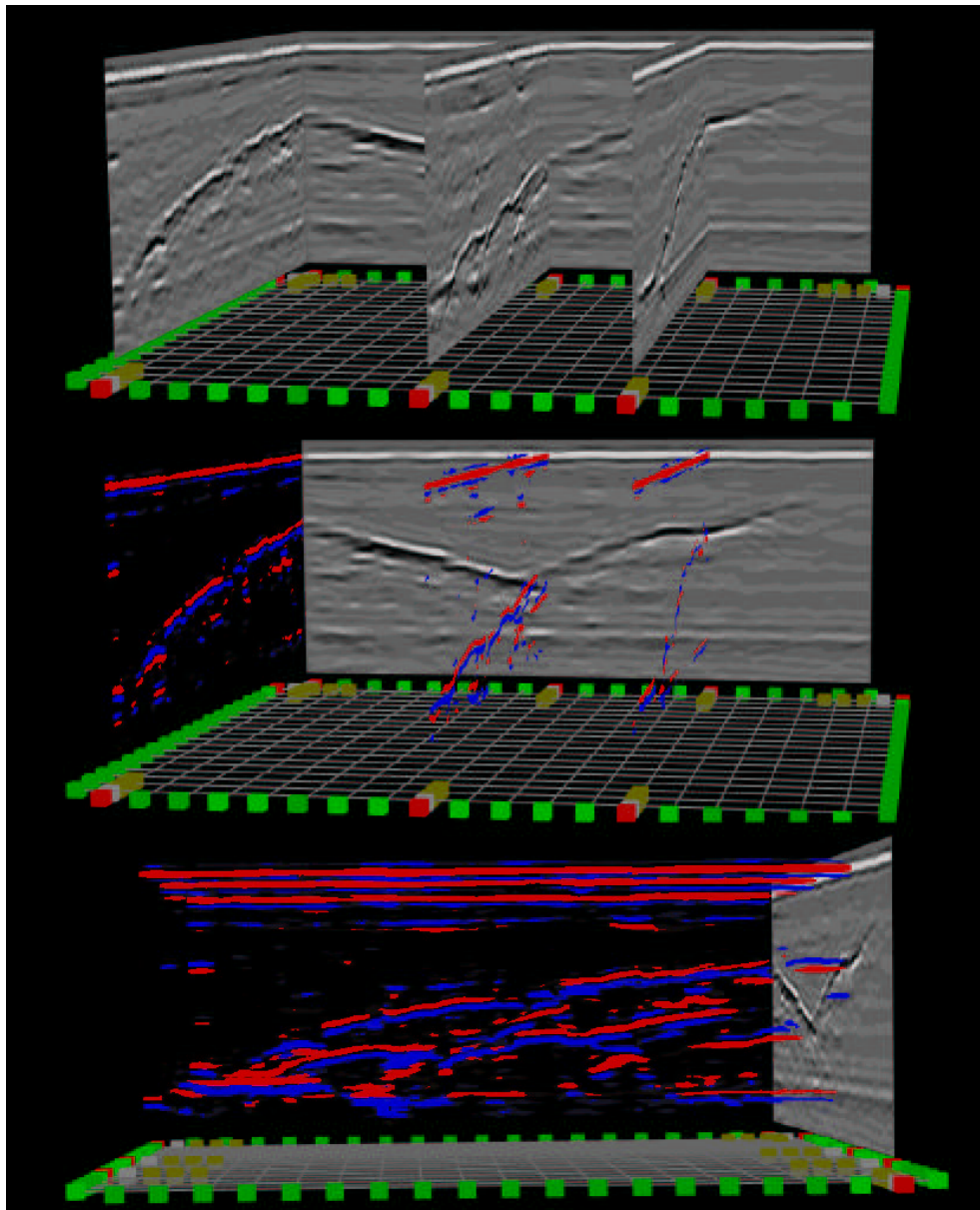


Abbildung 6.10: Drei verschiedene Darstellungen derselben vier Radargramme mit unterschiedlichem Darstellungsmodus der Schnitte oder unterschiedlicher Blickrichtung. Oben: Standard-Fence-Diagram, das die Radargramme als Graustufenbilder zeigt. Der Verlauf des Reflektors im Raum ist sichtbar, aber da sich die Schnitte gegenseitig verdecken, kann nur eine geringe Anzahl an Radargrammen gleichzeitig betrachtet werden. Mitte: Drei der Schnitte werden unter Verwendung einer Farbtabelle dargestellt, wie sie in Abbildung 6.7 beschrieben wird. Die Schnitte verdecken sich nicht gegenseitig, so daß eine größere Anzahl von Radargrammen gleichzeitig betrachtet werden kann. Unten: Die gleiche Darstellung wie in der Mitte, aber von einer anderen Richtung. Durch die Projektion auf die Papierebene ist es nicht mehr möglich, die räumlichen Zusammenhänge zu erkennen. Die Darstellung in Stereo würde dieses Problem lösen.

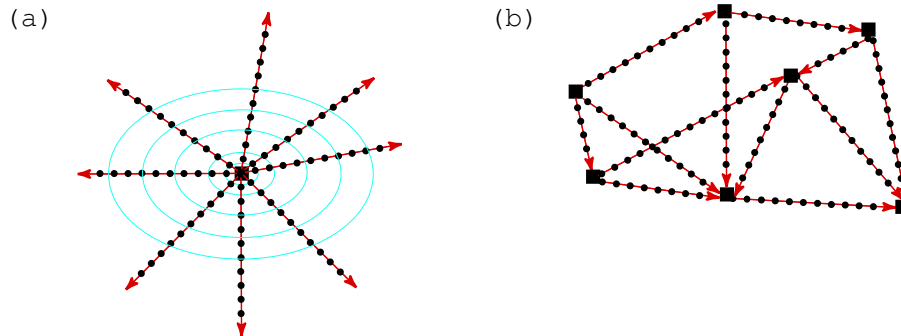


Abbildung 6.11: Weitere Meßanordnungen, die mit Hilfe von Stereo-Visualisierung möglich sind, um mit dem Georadar 3D-Strukturen im Untergrund zu erfassen. Die Pfeile zeigen die gemessenen Linien an, die schwarzen Punkte die daraus resultierenden CMP's (Common Mid Points). (a) Möglichkeit für eine Meßanordnung, mit der ein Absenkungstrichter mit Hilfe von Radargrammen erfaßt werden könnte. Auf diese Weise würden geringere Probleme mit der Migration auftreten als mit der klassischen Methode mit zwei Serien von Schnitten, welche senkrecht zueinander angeordnet sind. (b) Anordnung der Radargramme zwischen verschiedenen Bohrlöchern. In den Bohrlöchern erkennbare Schichtgrenzen lassen sich so unmittelbar in den Radargrammen verfolgen. Die Beeinflussung der Radargramme durch die Bohrungen und deren Casing sind so wahrscheinlich einfacher zu kontrollieren.

gewiesen, hin und wieder die Perspektive zu ändern, um sich über die räumlichen Zusammenhänge klar zu werden. Als einfacher Interpretationsfluß bot es sich hier an, den Reflektor auf 3 bis 4 Schnitten in eine Richtung zu markieren und dann der Reihe nach senkrecht zu diesen vorzugehen und einen Schnitt nach dem anderen abzuarbeiten. Diese Vorgehensweise funktioniert jedoch nur, wenn die Vermessung aus zwei senkrecht zueinander stehenden Gruppen von 2D-Schnitten besteht. Sind die Radargramme in mehreren Richtungen schief zueinander gemessen, so ist man mit zunehmender Anzahl an Schnitten auf die Stereo-Visualisierung angewiesen. Die Schnitte schief zueinander zu messen, kann zum Beispiel sinnvoll sein, wenn man bei einem Pumpversuch den Absenkungstrichter mit GPR erfassen möchte. Um hierbei einer 3D-Migration der Daten aus dem Weg zu gehen, kann man jeweils orthogonal zu den Isobathen messen, das heißt konzentrisch vom Förderbrunnen weg (Abb. 6.11a). Ein anderes Anwendungsbeispiel wäre das Messen von Radargrammen zwischen jeweils benachbarten Bohrungen, was bedeutet, daß Strukturen, die in den Radargrammen zu sehen sind, auf ihre Äquivalente in den Bohrungen zulaufen und der Seiteneinfluß der Bohrungen minimiert wird (Abb. 6.11b).

Dank der Benutzung der Farbtabelle läßt sich vorhersagen, ob und wie weit der Tracker den Reflektor findet. Durch Verschieben der Grenzwerte läßt sich dies gut beeinflussen. Bei der Benutzung eines Bildschirms läßt sich das Interpretationssystem auch auf einer normalen SGI-O2 Desktop Workstation bequem benutzen. Die Hauptanforderung ist, daß der Rechner Bilddaten (Texturen) schnell verarbeiten kann.





## 6.3 Hydrogeologisches Testfeld Krauthausen

### 6.3.1 Das Testfeld und die verwendeten Daten

Bei vielen geophysikalischen Untersuchungen des flachen Untergrundes werden verschiedene Meßmethoden verwendet, deren Ergebnisse anschließend miteinander integriert werden müssen. Beispiele hierfür sind archäologische Untersuchungen (z.B. Sambuelli et al., 1999) oder die Erkundung von Abfalldeponien (z.B. Green et al., 1999). Da bisher noch kaum Verfahren für die automatische Datenfusion verfügbar sind, muß die Integration der verschiedenen Daten in der Regel mit Hilfe visueller Methoden stattfinden. Hierbei kann die Anwendung immersiver Visualisierungstechniken unter Umständen sinnvoll sein.

Als Beispiel für die Visualisierung einer solchen Untersuchung werden in dieser Arbeit die Daten der geophysikalischen Messungen auf einem hydrogeologischen Versuchsfeld dargestellt. Dieses wird vom Institut für Chemie und Dynamik der Geosphäre (ICG) des Forschungszentrums Jülich betrieben, um experimentelle Untersuchungen zur Grundwasserströmung, zum Stofftransport und zur Messung von Aquiferparametern zu machen. Das Versuchsfeld liegt 7 km südöstlich von Jülich zwischen den Ortschaften Krauthausen und Niederzier und hat Ausmaße von ca. 230 m x 50 m. Das Feld weist nur geringe Reliefunterschiede auf, weshalb diese bei der Visualisierung auch vernachlässigt werden. Der flache Untergrund bis ca. 25 m Tiefe besteht aus quartären Sedimenten und weist mehrere Grundwasserstockwerke auf, wobei für die dort durchgeführten Tracerversuche hauptsächlich das obere Stockwerk relevant ist. In Englert (1998) ist ein aus Döring (1995) entnommenes generalisiertes Profil angegeben. Hiernach liegen in den obersten 1,3 m Löß und Auenlehm mit Pseudogley als Bodenbildung vor. Darunter folgen bis in eine Tiefe von ca. 10 m Kiese mit sandigen Lagen, welche zur Niederterrasse und jüngeren Hauptterrasse gehören. An der Basis werden diese tonhaltiger, was den obersten Grundwasserleiter von dem darunter liegenden trennt. Darunter folgen bis zur Endteufe der Bohrungen in ca. 21 m Tiefe Sande und Feinsande mit Tonlagen.

Das gesamte Gebiet ist mit insgesamt 72 Bohrungen erschlossen, die als Pegel und zur Entnahme von Wasserproben dienen. Bei einem Teil der Bohrungen wurde durch den Bohrmeister eine Feldaufnahme der Sedimenteinheiten gemacht, welche aus Englert (1998) entnommen werden konnten. Detaillierte Profile liegen allerdings nur für die Bohrungen mit den Kennnummern 1 bis 4 vor. Die Ergebnisse weiterer

Untersuchungen zu den hydraulischen Eigenschaften des Aquifers sind ebenfalls in Englert (1998) beschrieben. Um das Versuchsgelände möglichst umfassend zu beschreiben, wurden mehrere geophysikalische Untersuchungen durchgeführt, deren Daten im Rahmen dieser Arbeit für die Visualisierung zur Verfügung standen.

Von Schulze (1996) wurde im Rahmen einer Diplomarbeit an der Universität Bonn mit geoelektrischen Untersuchungen in Multielektrodenanordnung die Verteilung des elektrischen Widerstands im Untergrund bestimmt. Als Elektrodenkonfiguration kamen sowohl die Wenner Anordnung als auch die Dipol-Dipol Anordnung zum Einsatz. Hiernach zeichnet sich der Aquifer durch hohe spezifische Widerstände von 150 - 350  $\Omega m$  aus, während der Boden und der unterliegende Grundwasserstauer einen geringeren Widerstand von 40 - 60  $\Omega m$  aufweist. Die aufgenommenen Daten wurden im Rahmen der Diplomarbeit mit dem Programm IP3D invertiert, das an der TU Braunschweig entwickelt wurde (Seichter, 1998). Das Programm arbeitet nach dem Verfahren der finiten Differenzen, wobei die Diskretisierung mit Elementarvolumina erfolgt und so die Leitfähigkeit den Gitterelementen zugeordnet wird und nicht den Gitterpunkten (Seichter, 1998). Bei der Diskretisierung nahm der Gitterabstand in z-Richtung mit der Tiefe zu, und auch die Diskretisierung in x-Richtung wies unregelmäßige Gitterabstände auf. Für diese Arbeit standen die Daten für die verschiedenen Tiefenstufen in einzelnen Dateien als Schnitte parallel der xy-Ebene zur Verfügung. Insgesamt gab es 10 Schnitte mit den Zellgrenzen in z-Richtung: 0-1,0-2,2-3,6-5,3-7,4-9,9-12,9-16,5-20,8-26,0 m. In den Dateien sind jeweils die xy-Koordinaten des Zellenmittelpunktes und der dazugehörige spezifische Widerstand angegeben. Diese wurden in ein anderes Format, wie es nachfolgend bei der Visualisierung beschrieben wird, umgewandelt. Die Anzahl der Zellen und der abgedeckte Bereich des Gebietes waren für die invertierten Dipol-Dipol und die invertierten Wenner Daten unterschiedlich:

Dipol-Dipol:	17 x 152 x 10 Zellen auf dem Gebiet [2:35][22:175][-26:0] m
Wenner:	20 x 83 x 10 Zellen auf dem Gebiet [0:39][13:177][-26:0] m

Wie aus den Histogrammen in Abbildung 6.12 ersichtlich, liegen die ermittelten Werte für den elektrischen Widerstand bei der Invertierung der nach den Dipol-Dipol Meßverfahren gewonnenen Daten unter denen der nach dem Wenner Meßverfahren gewonnenen.

Von Seuffert (1997) wurden im Rahmen einer Diplomarbeit in 59 der auf dem Gelände vorhandenen Bohrungen Messungen mit Hilfe einer Gamma-Ray Sonde

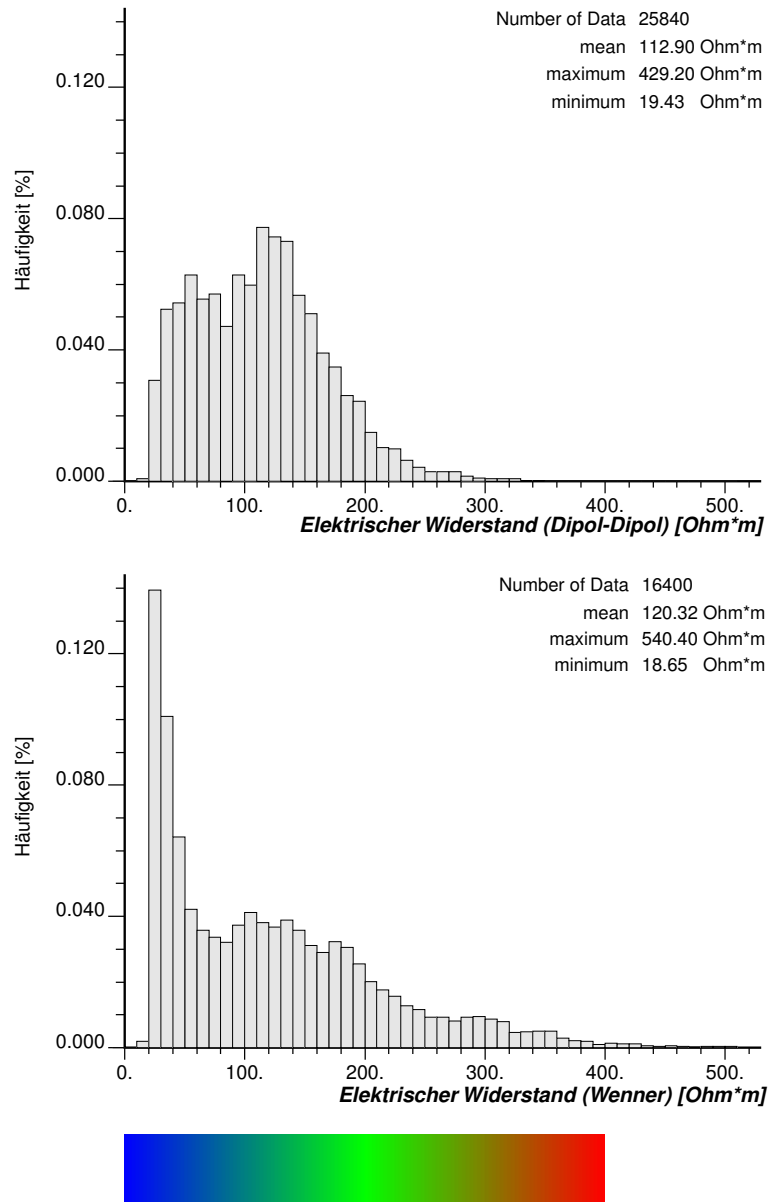


Abbildung 6.12: Mit der Programmbibliothek GSLib erstellte Histogramme für den elektrischen Widerstand der einzelnen Zellen, jeweils für die Messungen mit Wenner und Dipol-Dipol Anordnung. Unten wird die in späteren Abbildungen verwendete Farbkodierung für die verschiedenen Werte angezeigt.

vorgenommen. Da der Durchmesser der Sonde nur wenig geringer war als der Innendurchmesser der Verrohrung, kam es an den Übergangsstellen der einzelnen Rohrab-schnitte häufig zu Problemen, so daß die Bohrungen dann nicht bis zur Endtiefe son-diert werden konnten. Nach dem Abteufen der Bohrungen war ein PVC-Filterrohr eingesetzt und der Ringraum mit Filterkies und Tonsperren (Kompaktonit) verfüllt

worden. Da die Eindringtiefe der Sonde in die Formation sehr begrenzt ist, werden die gemessenen Daten nach Seuffert (1997) maßgeblich vom im nachhinein in den Ringraum eingefüllten Material beeinflusst. Im Bereich der Tonsperren wurden die Meßwerte wegen deren höheren Strahlungsaktivität verfälscht. Da die Tiefenlage der Tonsperren bekannt war, konnte ihr Einfluß jedoch korrigiert werden (Seuffert, 1997). Im oberflächennahen Bereich, wo die Strahlungsaktivität aufgrund der Verlehmung des Bodens erhöht ist, war die Lage der Kompaktonite nicht bekannt, so daß hier auf eine Korrektur verzichtet wurde. Aus den vorgenannten Gründen hängt der Verlauf der Gamma-Ray Kurve stark von der Verfüllung ab und spiegelt die ursprüngliche Schichtung nur bedingt wider. Wie weiter unten in Abb. 6.14 zu sehen ist, stimmen die Messungen der natürlichen Strahlungsaktivität dennoch weitgehend mit den beim Bohren aufgenommenen Profilen überein. So zeigen sich im Bereich der beiden Tonlagen in 12 m Tiefe deutlich erhöhte Gamma-Ray-Werte.

Nach Seuffert (1997) haben alle Log-Kurven den in der Abbildung 6.14 zu sehenden typischen Verlauf. Nahe der Oberfläche zeigen die Logs hohe Werte (größer 70 CPS), was dem lehmigen Boden entspricht. Im Bereich von 1 - 4 m fällt die Strahlungsaktivität auf ein niedriges Niveau von ca. 20 - 30 CPS ab, auf dem es bis in eine Tiefe von ca. 10 m bleibt. Dieser Bereich entspricht den schwach strahlenden Kiesen und Sanden. Bei den Logs, die bis in eine Tiefe von mehr als 10 m gemessen werden konnten, zeigt sich wieder ein Anstieg der Strahlungsaktivität, da an der Basis des oberen Grundwasserleiters der Tongehalt wieder steigt. Die Daten lagen mit einer Meßdichte von 5 cm digital vor.

In einem Teil der Bohrungen (21) wurde von Schneider (1995) mit Hilfe eines Flowmeters die Permeabilität ( $K_f$ -Wert) entlang der Bohrungen bestimmt. Die Messungen verlaufen in einem Tiefenbereich von ca. 3 bis 10 m unter der Geländeoberkante, liegen also alle im Bereich des oberen Grundwasserleiters. Die Daten weisen eine variable Meßdichte auf und liegen ebenfalls digital vor. Die Flowmetermessungen erfolgten nur im Aquifer in Bereichen, in denen der Wert der Gamma-Ray Messungen niedrig war (kleiner 30 CPS). Da außerdem die natürliche Strahlungsaktivität oft nicht bis zur Endteufe gemessen werden konnte, gibt es nur eine geringe räumliche Überschneidung der beiden Messungen. Daher erscheint ein Vergleich der Werte für die natürliche Strahlungsaktivität mit den Werten für die  $K_f$ -Werte nicht sinnvoll.

In der Abbildung 6.13 ist eine Übersicht des Versuchsgeländes mit den Bohrlokationen und den dort vorhandenen Messungen, sowie mit der Abgrenzung des

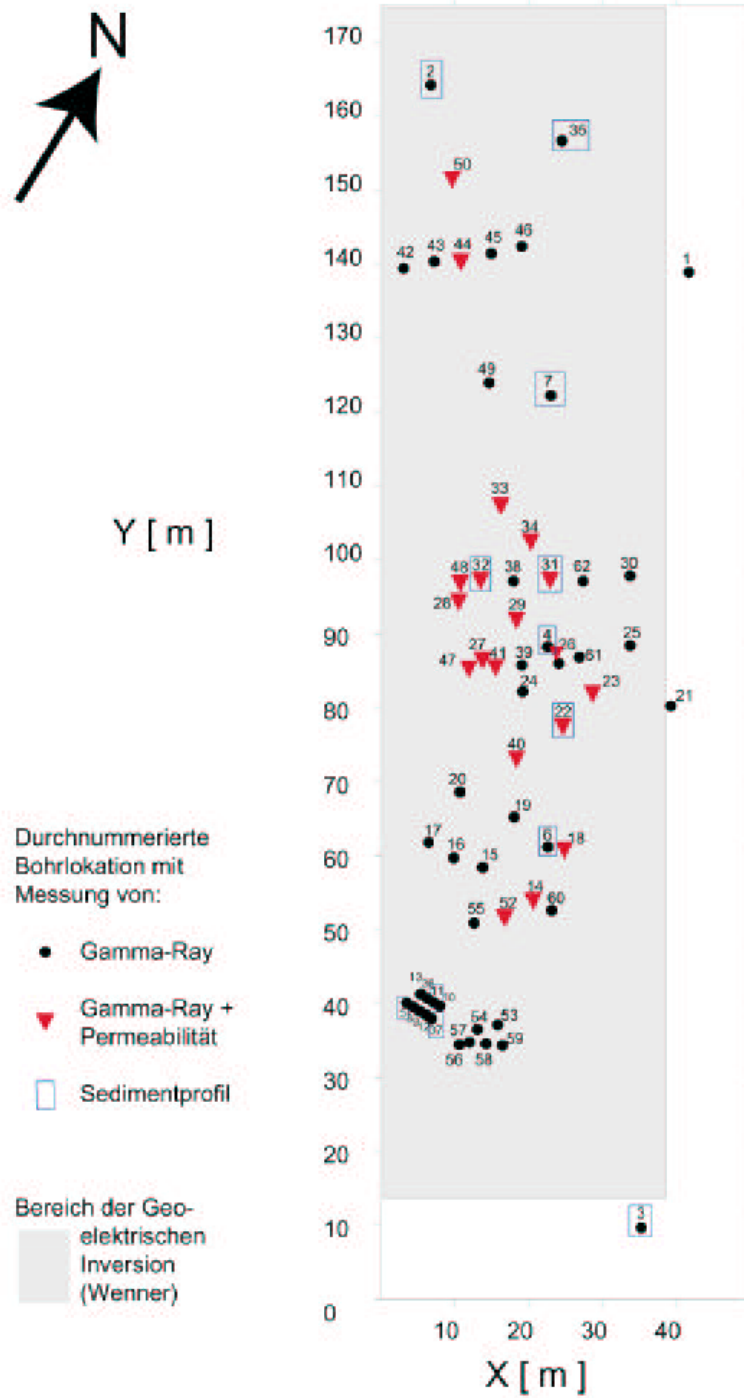


Abbildung 6.13: Übersicht über das Versuchsgelände Krauthausen mit seinen Bohrlokationen und den verschiedenen durchgeführten Messungen (nach Schulze, 1996).

durch geoelektrische Messungen abgedeckten Bereichs, zu sehen. Die erhobenen Daten liegen für die einzelnen Meßpunkte getrennt in Dateien vor und wurden für jeden einzelnen Meßpunkt getrennt in verschiedenen Diplomarbeiten dargestellt. Aus diesen vielen Einzeldiagrammen lassen sich die räumlichen Zusammenhänge jedoch nur schwer extrahieren, so daß sich eine 3D-Visualisierung anbietet. Wie bereits in Abbildung 4.1 aufgezeigt läßt sich eine hohe Anzahl an Bohrlöchern mit 3D-Visualisierung am Bildschirm nur schlecht übersichtlich darstellen. Die Verwendung von VR vereinfacht die Auswertung des Datensatzes und hilft dabei, räumliche Beziehungen zwischen den in den Bohrlochdaten sichtbaren Features und den in den Geoelektrik-Daten sichtbaren herzustellen. Da keine Standardprogramme für die Darstellung der im Versuchsfeld gemessenen Datentypen zur Verfügung standen, mußte, aufbauend auf die AVANGO Rahmenapplikation, ein entsprechender Software-Prototyp entwickelt werden. Im folgenden sollen die vorhandenen Daten und ihre Visualisierung vorgestellt werden.

### 6.3.2 Visualisierung der verschiedenen Datentypen

#### Bohrlochdaten und Sedimentprofile

Die Visualisierung der Bohrlochdaten erfolgt mit Hilfe der Implementierungen, welche zu diesem Zweck bereits im Rahmen des VRGeo-Projektes entwickelt worden sind. Im Falle der Messungen in Krauthausen sollten im Idealfall hohe Werte der natürlichen Strahlungsaktivität (hoher Tongehalt) mit niedrigen Werten für die hydraulische Permeabilität korrespondieren und umgekehrt. Das bedeutet, daß sich die sandig-tonigen Schichten durch die Farbkodierung deutlich von den kiesigen Schichten unterscheiden lassen müßten. Wegen der bereits erwähnten geringen räumlichen Überschneidung der beiden Meßwerte läßt sich dies jedoch in der Visualisierung nicht zeigen.

Zur Visualisierung der Bohrprofile läßt sich neben der Bohrung jeweils eine Art Meßplatte anzeigen, auf der über eine Farbkodierung jeweils die Schichtung zu sehen ist. In Abbildung 6.14 sind einmal zusammengefaßt die vorliegenden Daten der Bohrung Nummer zwei zu sehen (links) und die korrespondierende 3D-Darstellung (rechts). An dieser Lokation wurde die Permeabilität nicht gemessen. Wie in der Abbildung zu sehen, zeigt die natürliche Strahlungsaktivität trotz Messungen durch den Ringraum die Tonlagen deutlich an. Sie sind sowohl im Log-Plot als auch in der

3D-Darstellung gut zu erkennen.

Die Meßdaten werden aus Dateien eingelesen, die in der ersten Zeile jeweils für jede Spalte einen Kurznamen haben, der angibt, welche Daten in der jeweiligen Spalte gehalten werden. Um den Verlauf des Bohrloches relativ zu einem bestimmten Koordinatensystem korrekt darstellen zu können, sind für jeden Meßwert entlang des Bohrloches mindestens die Tiefe (TVDS = True Vertical Depth) und die x- und y-Koordinaten (XOFF und YOFF) anzugeben. Um gerichtete Daten darzustellen, müssen noch Streichen und Einfallen angegeben werden (AZI und DIP). Um die Profilaufnahme darzustellen, werden jeweils zwei Dateien eingelesen. Eine enthält die Farbzunordnung zu den verschiedenen Sedimenttypen, die andere enthält das eigentliche Profil mit den Schichtgrenzen und dem jeweiligen Sediment.

### **Geoelektrische Messungen**

Der elektrische Widerstand des Untergrundes liegt als volumetrischer Datensatz vor. Die Daten werden aus einer Datei gelesen, welche die Zellgrenzen in x-y-z-Richtung enthält und dann die Widerstandswerte der einzelnen Zellen. Diese werden sequentiell zuerst entlang der x-Achse, dann entlang der y-Achse und zuletzt entlang der z-Achse eingelesen. Die Inversion der Dipol-Dipol Messungen und der Messungen nach dem Wenner Verfahren fand auf unterschiedlichen Gittern statt, weswegen die Widerstandsdaten in zwei verschiedenen Dateien gespeichert sind. Die Diskretisierung der Gitter ist nicht äquidistant, weswegen die C++ Klassen, die im Rahmen des VRGeo-Projektes für die Darstellung von volumetrischen Datensätzen implementiert wurden, sich für die Visualisierung der hier vorliegenden Daten nicht eignen und eine neue Klasse entwickelt werden mußte.

Die Kodierung der Widerstandswerte wird über Farben vorgenommen, die wie bei der Darstellung von Karten und Schnitten üblich, den einzelnen Widerstandswerten zugeordnet werden. Da die Inversion mit einem Finite-Differenzen Verfahren erfolgte, welches auf der Diskretisierung mit Elementarvolumina beruht (Seichter, 1998), werden diese Farben jeweils den gesamten Zellen zugeordnet, und es wird keine Interpolation vorgenommen. Das hierbei erscheinende Bild erscheint zwar "blockig", spiegelt aber das vorhandene Wissen über den Untergund korrekt wider. Um das Farbspektrum gut auszunutzen, muß die Farbtabelle auf die vorhandenen Widerstandswerte abgestimmt werden. In Abhängigkeit von der Verteilung der Widerstandswerte kann sie mit Standardsoftware erstellt werden und wird als Bild

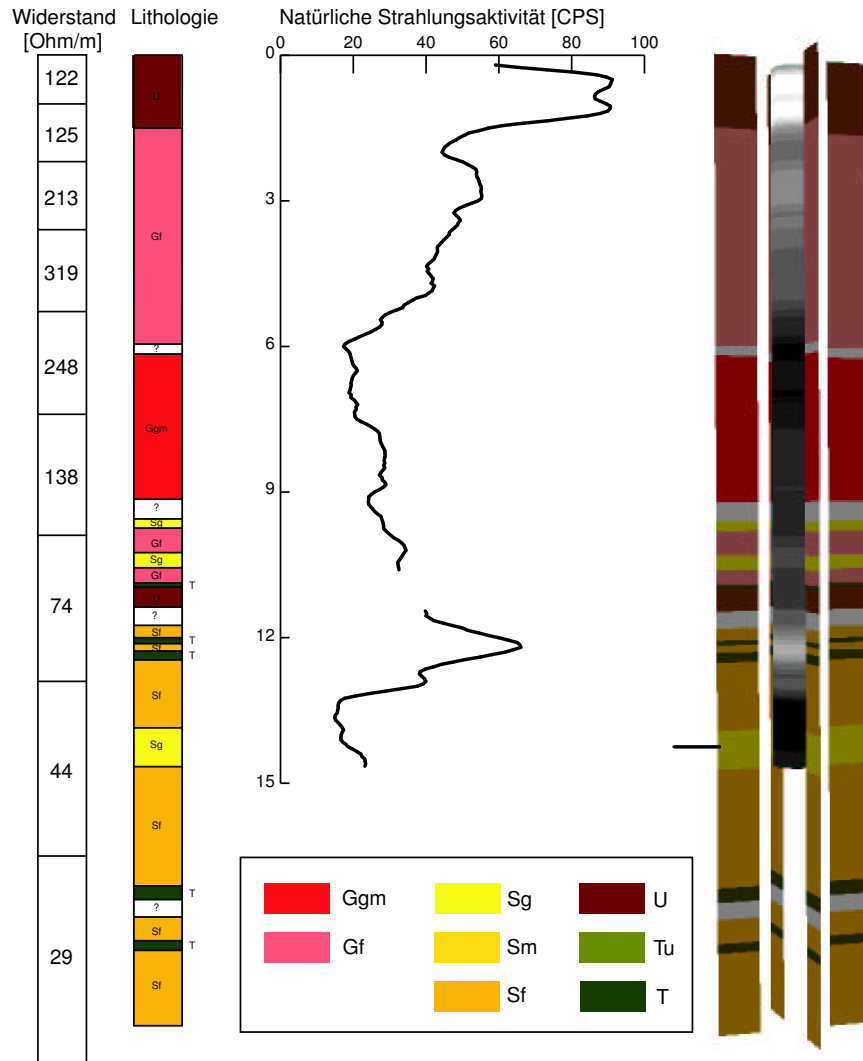


Abbildung 6.14: Darstellung der Daten an der Bohrung Nummer zwei. Auf der linken Seite sind das während der Bohrung aufgenommene geologische Profil und die durch Inversion der geoelektrischen Messungen im Wenner-Verfahren ermittelten elektrischen Widerstände zu sehen, in der Mitte die mit der Gamma-Ray Sonde gemessene natürliche Strahlungsaktivität, darunter die Legende für die Farbkodierung der Sedimentpetrographie. Wie zu sehen, zeichnen sich die Tonlagen in der Meßkurve deutlich ab, obwohl durch den verfüllten Ringraum gemessen werden mußte. Auf der rechten Seite ist die Darstellung des Sedimentprofils (außen) und des Gamma-Ray Logs (innen) als Visualisierung in einer virtuellen Umgebung zu sehen. Auch in der 3D-Visualisierung lassen sich die Tonlagen gut erkennen (helle Bereiche). Deutlich wird auch, daß die Inversion der geoelektrischen Daten den Untergrund nur sehr generalisiert wiedergibt.



geladen. In Abbildung 6.12 ist die verwendete Farbtabelle unter den Histogrammen angezeigt. Bei der Visualisierung wird das Spektrum der elektrischen Widerstände zwischen einem wählbaren minimalen und maximalen Wert in 255 Bereiche unterteilt, denen jeweils ein Farbwert zugeordnet wird. Alle Zellen mit Werten unterhalb des Minimums erhalten den Farbwert für das Minimum, und alle Zellen mit Werten oberhalb des Maximums erhalten den Farbwert für das Maximum.

Für die Visualisierung der Daten gibt es zwei Möglichkeiten: die Darstellung der Daten auf Schnittebenen durch den Raum und die Darstellung als volumetrisches Voxelmodell. Bei der Visualisierung mittels Schnittebenen (Abb. 6.15) stehen drei senkrecht zueinander stehende Schnitte zur Verfügung, welche interaktiv durch den Datensatz bewegt werden können.

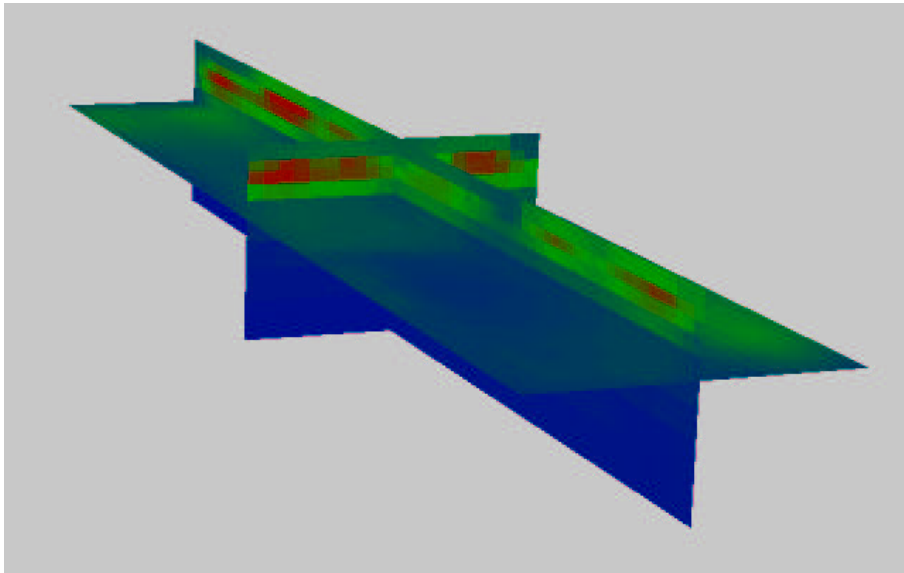


Abbildung 6.15: Darstellung des elektrischen Widerstandes mittels dreier Schnittebenen durch das Datenvolumen. Für die Farbkodierung siehe Abbildung 6.12. Die räumliche Ausdehnung beträgt 39 m x 164 m x 26 m.

Auf diesen Schnitten sind die Ausmaße der geschnittenen Zellen und - über die Farbkodierung - die jeweiligen Widerstandswerte zu sehen. Die Steuerung der Schnitte erfolgt über die Cubic-Mouse. In einem früheren Modell sollte die Steuerung über einen Käfig, der das Modell umgab, erfolgen, um nach Möglichkeit von speziellen Eingabegeräten unabhängig zu sein. Die Schnittebenen folgten hierbei dem Schnittpunkt des Zeigestrahls mit den Käfigachsen, die parallel zu den Koordinatenachsen verliefen. Am Bildschirm war diese Form der Bedienung gut durchführbar. In der CyberStage oder auf der Holobench war jedoch die Präzision, mit welcher die Be-

nutzer den Zeigestrahl steuern konnte, nicht ausreichend. Außerdem störte der Käfig den Blick auf Teile des Modells.

Bei der Darstellung als volumetrisches Voxelmodell (Abb. 6.16) kann der Benutzer einen oberen und unteren Grenzwert eingeben, der bestimmt, welche Zellen dargestellt werden sollen und welche nicht. Auf diese Weise werden innerhalb des Datensatzes zum Beispiel die Regionen, die einen hohen elektrischen Widerstand haben, als dreidimensionale Körper sichtbar. Um dem Benutzer die gleichzeitige Beobachtung des elektrischen Widerstandes als Volumen und der Bohrlochdaten zu ermöglichen, läßt sich das Voxelmodell transparent darstellen. Dies erwies sich im Testbetrieb jedoch als zu verwirrend und unübersichtlich für den Betrachter.

### 6.3.3 Gesamtmodell und Interaktion

Das hydrogeologische Testfeld ist ungefähr 160 m lang, 40 m breit und wurde bis zu einer Tiefe von ca. 25 m untersucht. Aufgrund des für 3D-Darstellungen in VE's ungünstigen Länge-Breite-Höhe-Verhältnisses läßt sich der zur Verfügung stehende Visualisierungsraum auf der Holobench nicht ohne weiteres vollständig nutzen. Hinzu kommt, daß die räumliche Auflösung der elektrischen Widerstandsmessungen sehr viel niedriger ist als die Auflösung der Bohrlochmessungen. Aus diesem Grund erschien es notwendig, das Gebiet in einzelnen Teilmodellen betrachten zu können. Durch das Verschieben von acht Ecksteinen läßt sich ein quaderförmiger Unterbereich definieren, welcher genauer betrachtet werden soll. Für diesen wird die Einhüllende berechnet und das Zentrum des Modells und seine Skalierung so gesetzt, daß der Interaktionsraum der Holobench von diesem Unterbereich optimal ausgenutzt wird. Ferner besteht die Möglichkeit, alle Geometrien außerhalb dieses Bereiches wegzuschneiden. Die Orientierung des Modells und die Position der drei senkrechten Schnittebenen werden mit Hilfe der Cubic-Mouse gesteuert.

Im Vordergrund befinden sich eine Reihe von (virtuellen) Knöpfen, mit denen der Benutzer zwischen verschiedenen Visualisierungsoptionen auswählen kann. So kann man bei den Bohrlöchern wählen, ob die Permeabilität oder die natürliche Strahlungsaktivität angezeigt werden soll. Von der Möglichkeit, über eine bivariate Farbtabelle beide Werte gleichzeitig anzuzeigen, wurde aus Gründen der geringen räumlichen Überschneidung kein Gebrauch gemacht. Bei der Geoelektrik kann zwischen den Dipol-Dipol Daten und den Wenner Daten hin- und hergeschaltet werden. Ferner lassen sich die Visualisierung als Schnitte durch den Datenraum und die Vi-

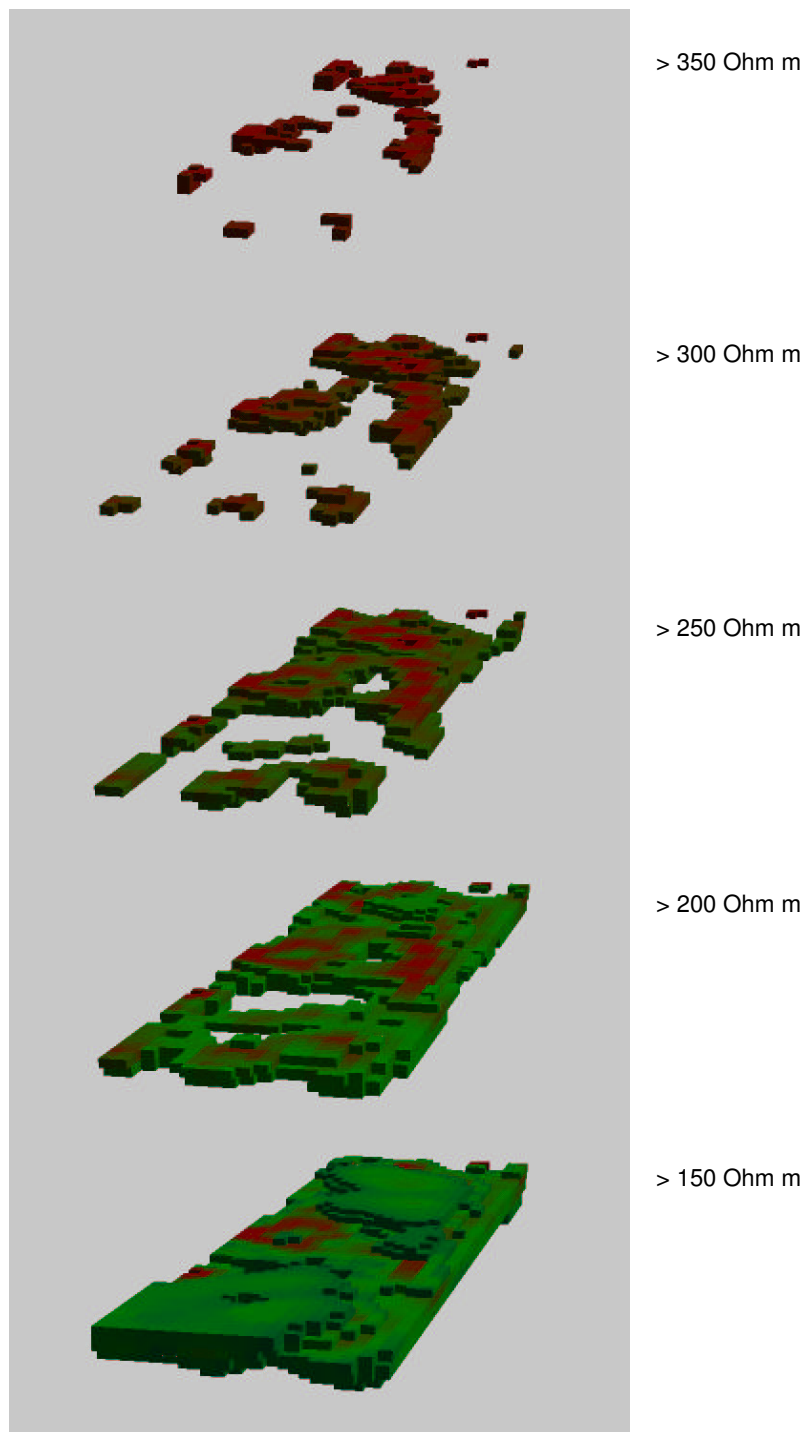


Abbildung 6.16: Darstellung der elektrischen Daten als Volumendatensatz mittels Visualisierung als Voxel. Hier sind jeweils nur diejenigen Zellen sichtbar, deren Widerstand sich über einem vorgegebenen Schwellenwert befindet. Die Farben der Zellen zeigen auch hier die Werte für den elektrischen Widerstand an.

sualisierung als Voxellmodell jeweils an- und ausschalten. Weitere Knöpfe lösen die Anpassung der jeweiligen Modellregion an den Interaktionsraum aus oder setzen das Modell in seinen Anfangszustand zurück.

Zusätzliche Interaktionen sind mit Hilfe von virtuellen Werkzeugen möglich. So kann z.B. ein Info-Werkzeug dazu benutzt werden, auf verschiedene Objekte zu zeigen und sich anzeigen zu lassen, um welches Bohrloch es sich handelt und welche Messungen dort vorliegen, oder um sich zu informieren, welchen geoelektrischen Datensatz man gerade betrachtet und an welcher Position sich der jeweilige Schnitt befindet. Ein weiteres Werkzeug dient dazu, den Datensatz achsenparallel im Raum zu bewegen. Auf diese Weise kann man beidhändig arbeiten: Mit der linken Hand bestimmt man mittels der Cubic-Mouse die Orientierung und mit der rechten Hand mit Hilfe des Stylus die Positionierung. Die Position durch Bewegung der Cubic-Mouse zu bestimmen, wie es bei den Demonstratoren des VRGeo Projektes gemacht wurde, ist ungünstiger, da das Gerät schwerer und unhandlicher ist als der Stylus.

### 6.3.4 Erfahrung beim Arbeiten mit dem System

Das vorgestellte Modell eignet sich gut, um mit den Daten zu arbeiten und um über sie zu diskutieren. Die Stereodarstellung hilft, bei der großen Menge von Daten den Überblick zu behalten. Während die Darstellung der Schnittebenen durch die Geoelektrik auch ohne die Stereo-Darstellung ohne Probleme möglich ist, wäre dies bei der hohen Anzahl der Bohrlöcher schwieriger und würde andere Unterstützung, zum Beispiel die Verwendung von Hilfslinien, benötigen. Da die Kodierung (Abbildung) der Meßwerte über Farben vom Benutzer aus Wahrnehmungsgründen nur sehr ungenau umgekehrt werden kann, wäre ein Info-Tool, mit dem sich die Meßwerte direkt von den Objekten abfragen lassen, hilfreich. Da das Untersuchungsgelände ein recht großes Länge-Breite-Verhältnis hat, eignet es sich von der Geometrie her nicht besonders gut zur Darstellung in einer virtuellen Umgebung. Die Darstellung auf der Workbench erfolgt am besten nahe der Projektionsfläche, so daß das Modell auch ohne Probleme gedreht werden kann. Große Teile des Modells liegen dann aber außerhalb des Interaktionsraumes.

Bei der Betrachtung in der VR-Umgebung werden auch die unterschiedlichen Größenordnungen deutlich, mit welchen die verschiedenen Messungen den Untergrund beschreiben. Während in den Bohrlöchern Messungen im Abstand von 5 cm vorliegen, sind die Zellen für die elektrischen Daten an der Oberfläche 1 m und

im tiefsten Bereich 5 m mächtig und haben eine Querschnitt von 2 m x 2 m. Dies ist besonders auffällig, da die Zellen direkt dargestellt werden und nicht wie meist üblich eine Konturdarstellung interpoliert wird. So wird besonders deutlich, daß die Inversion der geoelektrischen Daten nur eine sehr generalisierte Darstellung des Untergrunds liefert. Auf der anderen Seite lassen sich bei Betrachtung des gesamten Untersuchungsgebietes in den Bohrlöchern keine Details mehr erkennen, da deren Darstellung dann zu klein ist. Was den Datensatz betrifft, sind die in ihm sichtbaren geologischen Strukturen relativ einfach. Daher wäre es in diesem Fall auch ohne Probleme möglich, die Daten mit einem Standard-Visualisierungssystem ohne stereoskopische Darstellung zu untersuchen. Die folgenden Abbildungen 6.17 und 6.18 sollen eine Vorstellung vermitteln, wie das Arbeiten mit dem Modell an der Holobench aussieht. Wie man erkennen kann, ist die Orientierung des Modells an die Cubic-Mouse gekoppelt. Mit Hilfe des Stylus kann man das Modell bewegen oder Abfragen an die verschiedenen Objekte durchführen (siehe Abbildung 6.17).

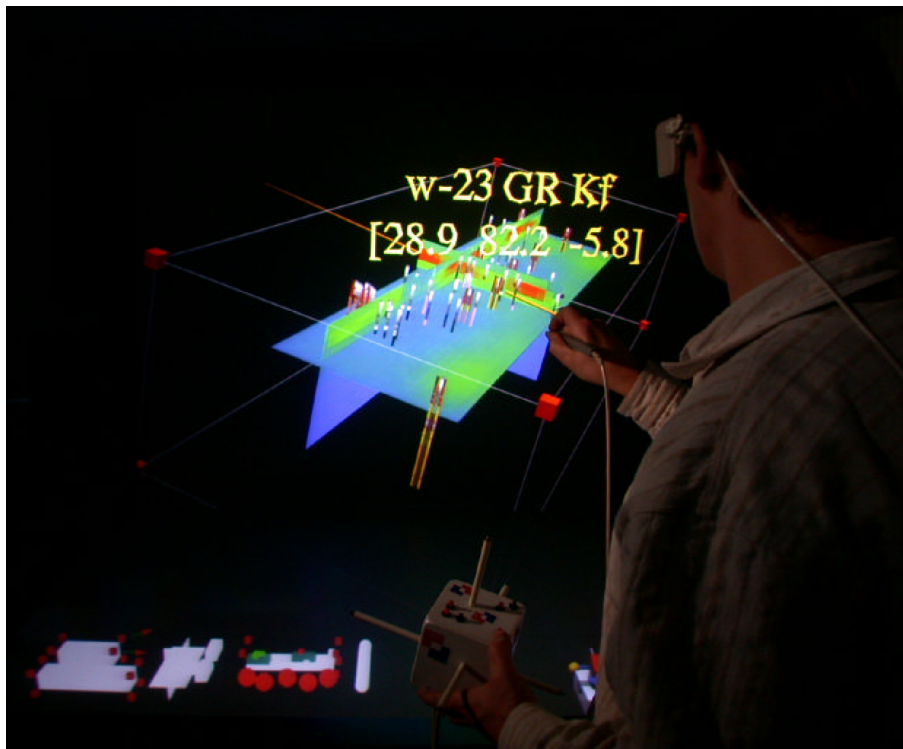


Abbildung 6.17: Visualisierung des Krauthausen Datensatzes an der Holobench. Der Benutzer hat mit dem Stylus ein Bohrloch ausgewählt und bekommt angezeigt, um welches Bohrloch es sich handelt und daß dort Messungen der natürlichen Strahlungsaktivität (GR) und der hydraulischen Permeabilität (Kf) vorliegen.

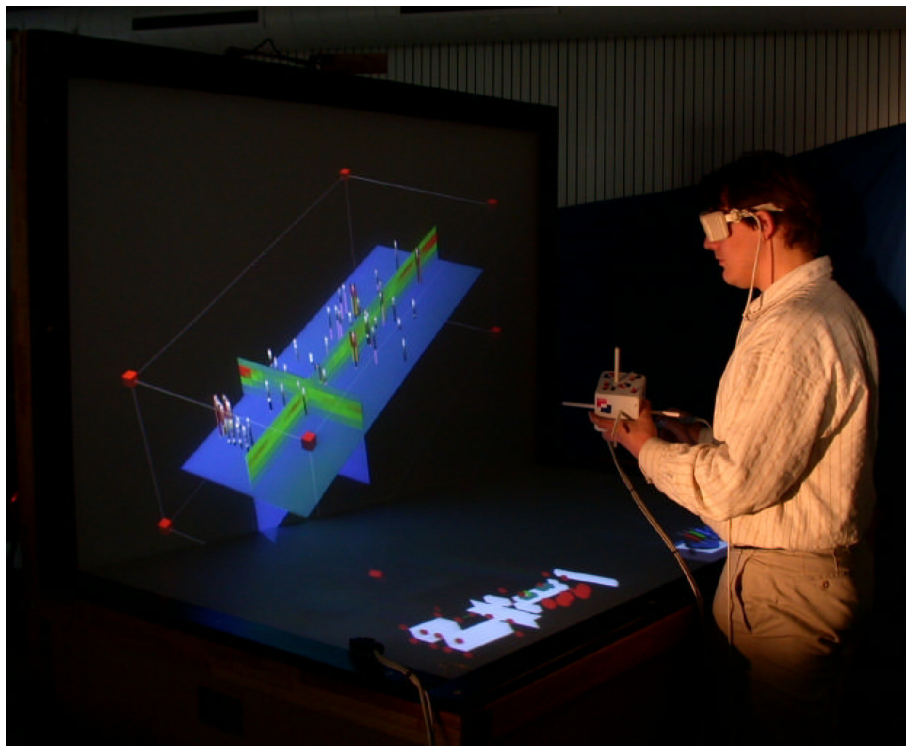


Abbildung 6.18: Visualisierung des Krauthausen Datensatzes an der Holobench.

## 6.4 4D-Darstellung seismologischer Daten

### 6.4.1 Geothermie in Soultz-sous-Forêts

Wie bereits erwähnt, ist es bei der Darstellung räumlich-zeitlicher Abläufe besonders wichtig, die dreidimensionalen Zusammenhänge einfach und schnell zu erfassen. Bei statischen 3D-Modellen am Bildschirm hat der Benutzer die Möglichkeit, die Modelle zu bewegen, um sich die räumlichen Zusammenhänge durch die Betrachtung aus unterschiedlicher Perspektive klar zu machen. Bei dynamischen Modellen hat er hierzu nicht die Zeit, da sich die Szenen kontinuierlich ändern. Oft ist es daher notwendig, Bewegungen immer wieder ablaufen zu lassen und diese aus verschiedenen Richtungen zu betrachten, um die räumlichen Bewegungsabläufe zu begreifen. Da die Verwendung von Stereo-Visualisierung die Bewegung im Raum sichtbar macht, lassen sich solche 4D-Modelle in immersiven Visualisierungssystemen besser darstellen als klassisch am Bildschirm. Durch die direkte Interaktion unterstützen sie zudem das schnelle Arbeiten mit den Modellen.

Als Beispiel für eine solche Darstellung räumlich-zeitlicher Daten soll hier die Visualisierung von seismologischen Messungen dienen, welche bei hydraulischen Tests eines geothermischen "Hot-Dry-Rock"-Reservoirs in Soultz-sous-Forêts aufgenommen wurden (Baria et al., 1999). Beim "Hot-Dry-Rock"-Verfahren werden in der Tiefe liegende heiße Gesteine mit Bohrungen erschlossen. Zwischen den Bohrungen werden mittels hydraulischer Stimulation Fließwege geschaffen bzw. bestehende Wege ausgeweitet. Auf diese Weise wird ein unterirdischer Wärmetauscher erzeugt, in dem von der Oberfläche eingepreßtes Wasser erhitzt wird. Wieder nach oben gefördert, wird dieses Wasser durch einen Wärmetauscher geleitet und so die gewonnene Wärme nutzbar gemacht (Bad Urach, 2001).

Als Standort für die Erforschung dieser Technologie wurde ein Gebiet in der Nähe von Soultz-sous-Forêts bei Straßburg ausgewählt. Hier ist der geothermische Wärmefluß besonders hoch und das Gebiet durch mehrere Erdölbohrungen im sedimentären Deckbereich bereits erschlossen (Baria et al., 1999). Ziel ist es, die Wärme aus einem Granitkörper zu fördern, der unterhalb von 1400 m Sedimentbedeckung liegt. Die Temperatur steigt hier von ca. 100° Celsius an der Oberkante auf 160° - 170° Celsius in 4000 m Tiefe an. Zu diesem Zweck wurde 1993 die Bohrung GPK1 bis in 3590 m Tiefe vorgetrieben, wobei die letzten 780 m unverrohrt blieben. Eine zweite Bohrung (GPK2) wurde Ende 1994 bis in eine Tiefe von 3890 m vorgetrieben,

die letzten 650 m davon blieben ebenfalls unverrohrt. In den Modellen für die Visualisierung sind jeweils die Trajektorien dieser beiden Bohrungen dargestellt. Nach Baria et al. (1999) weisen Messungen in den Bohrlöchern auf zwei Hauptrichtungen für die Klüfte hin. Sie streichen jeweils mit 170 Grad und fallen mit 65 Grad nach Westen bzw. mit 70 Grad nach Osten ein.

### 6.4.2 Seismologie zur Beobachtung hydraulischer Tests

Um die Bewegung des Wassers im Untergrund zu bestimmen, werden hydraulische Tests durchgeführt, bei denen unter hohem Druck eine große Menge Wasser in die Formation eingepreßt wird. Dadurch werden bereits vorhandene Schwächezonen des Gesteins reaktiviert, so daß es zum Aufbrechen und zu Bewegungen innerhalb des Gesteins kommt, wobei jeweils Schallsignale generiert werden. Durch Lokalisierung der Herde dieser Schallsignale lassen sich also Rückschlüsse über die Bewegung des Wassers im Untergrund ziehen. Um die seismischen Signale aufzuzeichnen, wurden daher in vier älteren Erdölbohrungen in Soultz-sous-Forêts Seismometer installiert.

Eine der Meßkampagnen für hydraulische Tests in der Bohrung GPK1 lief von Mai bis Oktober 1993. Dabei wurde die Mikroseismizität beobachtet und die Bebenherde und die Stärke der verschiedenen Mikrobeben bestimmt. Die Ergebnisse dieser Tests sind in Jones et al. (1995) veröffentlicht. Die für diese Arbeit zur Verfügung stehenden Daten wurden im Zeitraum vom 02. September bis 28. Dezember 1993 aufgenommen und bestehen aus insgesamt 12.875 Mikrobeben. Nach Jones et al. (1995) wurden in diesem Zeitraum drei verschiedene Injektionstests durchgeführt, bei denen jeweils verschiedene Bereiche der Bohrung mit Packern isoliert wurden. Im September wurde bei der Bohrung GPK1 in einer Tiefe von 2850 - 3400 m Wasser eingepreßt. Für diesen Zeitraum standen für die Visualisierung die Lokation und Magnitude von ca. 10.800 seismischen Ereignissen zur Verfügung. Im Oktober und November 1993 wurde zuerst in den Tiefenbereich von 3457 - 3507 m Wasser injiziert und danach in den gesamten Bereich des offenen Bohrloches von 2850 m bis in 3600 m Tiefe.

Bei der Auswertung der Daten bereiten die hohe Anzahl der Datenpunkte und die Tatsache, daß diese unregelmäßig im Raum verteilt sind, Probleme. Zudem kommt noch die Zeit als zusätzliche Dimension hinzu. Standard 3D-Visualisierung läßt sich auf diese Daten nur sehr schwer anwenden, da der räumliche Bezug der einzelnen Datenpunkte zueinander kaum herzustellen ist. Eine Möglichkeit der Auswertung



ist das Legen von Schnitten durch den Raum, wobei alle Bebenherde, die innerhalb einer definierten Entfernung der Fläche liegen, entlang der Flächennormalen auf die Fläche projiziert werden. So werden in Jones et al. (1995) jeweils Tiefenschnitte gezeigt, bei denen alle Bebenherde innerhalb eines 200 m mächtigen Tiefenbereiches auf die Horizontale projiziert werden. Auf diese Weise wird sichtbar, wie sich zur Tiefe hin das Streichen der Vorzugsrichtung bei der Ausbreitung des Wassers ändert. Was die dreidimensionale Ausbreitung betrifft, ist der Benutzer wieder darauf angewiesen, sich aus den verschiedenen Schnitten ein mentales Modell zu generieren. Ein weiterer Nachteil dieser Darstellungsmethode ist, daß man keinen Überblick über die zeitabhängigen Vorgänge bekommt. Um den räumlich-zeitlichen Verlauf der Ausbreitung des injizierten Wassers direkt sichtbar zu machen, wurde im Rahmen dieser Arbeit ein Software-Prototyp entwickelt, welcher in der Lage ist, für die Darstellung der Daten immersive Visualisierungssysteme zu nutzen.

### 6.4.3 Verwendung von VR zur Betrachtung der Messungen

Um einen ersten Überblick zu bekommen, wurde ein statisches Modell erstellt, das heißt alle Bebenherde werden gleichzeitig als Marker im Raum dargestellt. Die Farbe der Marker zeigt hierbei entweder den Zeitpunkt des seismischen Ereignisses oder seine Magnitude an. Diese Darstellungsweise verschafft aber nur einen ersten Eindruck, in welchem Bereich die Beben stattfinden, wo z.B. Kluster von Beben auftreten und ob es regionale Unterschiede bezüglich der Magnitude gibt. Verwendet man bei der Darstellung des September-Tests die Zeitachse zur Farbkodierung, so kann man z.B. erkennen, wie sich die Beben vom Bohrloch weg immer weiter ausbreiten. Da der Benutzer nicht in der Lage ist, die Farben exakt einzuordnen und wieder zurück auf die Zeitachse abzubilden, erlaubt die Darstellung der Zeit mittels Farbkodierung nur eine oberflächliche zeitliche Einordnung. Der räumlich-zeitliche Ablauf kann daher nur in groben Zügen erfaßt werden. Verwendet man die Magnitude für die Farbkodierung, so läßt sich ein genereller Trend zu schwächeren Beben nach oben hin beobachten. Ferner wird sichtbar, daß das Wasser sich bevorzugt auf einer Fläche ausbreitet, welche mehr oder weniger saiger steht und mit ca. 170 Grad streicht. Das Streichen entspricht also dem der Hauptkluftrichtungen. Wie in Jones et al. (1995) dargestellt, ändert sich das Streichen jedoch geringfügig mit der Tiefe. Dies läßt sich aus der Gesamtdarstellung nur sehr schwer ersehen, da der Benutzer nicht alle Beben gleichzeitig in ihrer räumlichen Tiefe erfassen kann und die für ihn vorne liegenden Informationen die dahinter liegenden verdecken. Um derartige Eigenschaf-

ten des Datensatzes zu erkennen, ist es hilfreich, die Bebenherde nur innerhalb einer z.B. 100 oder 200 m dicken Schicht anzuzeigen, die mit Hilfe eines Griffes interaktiv hoch und runter bewegt werden kann. Um kleinräumige Strukturen, wie zum Beispiel die Richtung einer reaktivierten Kluftfläche, sichtbar zu machen, kann man einzelne Bereiche des Datensatzes ausschneiden und vergrößern. Diese Subvolumen werden wie in Kapitel 4 beschrieben festgelegt. Wie Phillips (2000) gezeigt hat, wäre es für die Erkennung kleinräumiger Strukturen in solchen Datensätzen jedoch notwendig, die einzelnen Mikrobeben mit Hilfe manueller Techniken zu bestimmen, da die auf diesen Datensatz angewendeten automatischen Algorithmen bisher noch keine ausreichende Präzision aufweisen. Die manuelle Bestimmung wäre für einen Datensatz dieser Größe jedoch zu aufwendig.

Um den zeitlichen Ablauf zu analysieren, wurde ein dynamisches Modell entwickelt. Die Zeitspanne des hydraulischen Tests wird dabei im Zeitraffer abgespielt, und die Markierungen für die verschiedenen Bebenherde sind jeweils von dem Zeitpunkt an, zu dem der seismische Event eintrat, für eine vorher festgelegte Zeitspanne zu sehen. Als Standardeinstellung repräsentiert bei der Visualisierung eine Sekunde einen Zeitraum von sechs Stunden beim Experiment, wobei die Beben für jeweils 36 Stunden angezeigt werden. Die Farben der Marker zeigen hierbei die Magnitude der Erdbeben an. Um sich verschiedene Abläufe wiederholt ansehen zu können, kann eine Zwischenzeit gespeichert werden, zu der dann zurückgesprungen wird. Ferner besteht die Möglichkeit, jederzeit die Zeit anzuhalten und wieder zu starten. Um die Position von einzelnen Features im Datensatz zu bestimmen, kann ein Marker frei im Raum bewegt werden, dessen Position im Bereich der Benutzerschnittstelle angezeigt wird. In diesem Modus ist es möglich, den räumlich-zeitlichen Ablauf genauer zu analysieren. Nimmt man zum Beispiel die Daten des Septemberexperiments, so kann man sehen, wie sich in einer Tiefe von 2910 m unmittelbar um das Bohrloch die ersten Beben bemerkbar machen. Nach ca. 4 Tagen beginnen sich diese dann nach unten bis in eine Tiefe von 3100 m auszubreiten. Für weitere vier Tage sind die Mikrobeben hauptsächlich nördlich des Bohrloches zu beobachten, erst danach erfolgt eine Bewegung des Bebenschwarmes in südliche Richtung und nach oben. Nach ca. 20 Tagen treten kaum noch Ereignisse ein. Abbildung 6.19 zeigt auf der linken Seite die Mikrobeben, die im Zeitraum von ca. 10 - 11,5 Tagen, und auf der rechten Seite jene, die im Zeitraum zwischen ca. 13 und 14,5 Tagen stattgefunden haben. Es ist deutlich zu erkennen, daß die Beben sich in einzelnen Bereichen konzentrieren, die ihre Position verändern.

Als Benutzerschnittstelle steht eine Reihe von Knöpfen zur Verfügung, mit de-

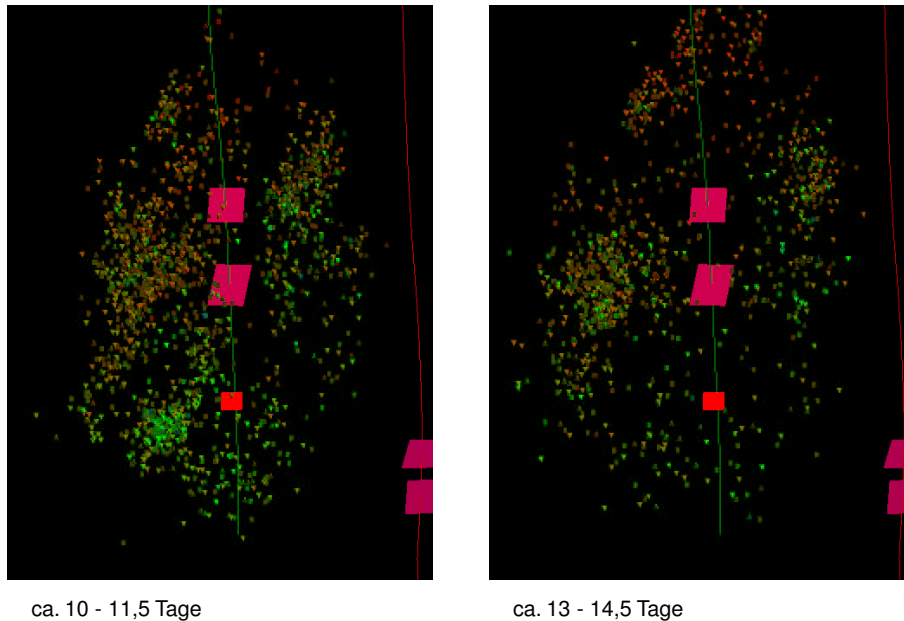


Abbildung 6.19: Herdlokationen von Beben, welche in der Zeit von ca. 10 - 11,5 Tagen nach Start des Experiments (02. September 1993) stattgefunden haben (links) und von Beben, welche sich in der Zeit von ca. 13 - 14,5 Tagen ereignet haben (rechts).

nen sich die verschiedenen Optionen der Visualisierung einstellen lassen. Eine erste Gruppe von Knöpfen ist für die Auswahl der verschiedenen Visualisierungsmodi zuständig, d.h. für den Wechsel zwischen der dynamischen Visualisierung, der statischen Visualisierung der Bebenherde in einem gegebenen Tiefenbereich und der gleichzeitigen Darstellung aller Bebenherde. Mit einer zweiten Gruppe von Knöpfen läßt sich auswählen, welcher Parameter für die Farbkodierung verwendet werden soll. Eine dritte Gruppe steuert die von der Zeit abhängige Funktionalität, wie das Stoppen und Zwischenspeichern der Zeit und das Zurückspringen an den Anfang. Zwei weitere Knöpfe lösen die Größenanpassung des Modells in Abhängigkeit vom gewählten Ausschnitt aus oder setzen das Modell in den Anfangszustand zurück. Ein frei beweglicher Marker kann innerhalb des Modells positioniert und orientiert werden, um räumliche Parameter abzufragen. Seine Position, die Orientierung als Streichen und Einfallen und der Zeitpunkt, zu dem er das letzte mal verschoben wurde, werden neben der Benutzerschnittstelle angezeigt.

Das Modell läßt sich sowohl auf der Workbench als auch in der CyberStage und auf dem Monitor starten. Allerdings wurde, anders als bei der Visualisierung von Radar-Daten, keine Benutzerschnittstelle für den Monitor entwickelt. Aus geometrischen Aspekten (die vertikale Ausdehnung des Untersuchungsgebietes ist wesentlich

höher als die horizontale) und da die Darstellung aus einer Vielzahl kleiner Objekte besteht, wurde die wesentlich größere Darstellung in der CyberStage vorgezogen.

#### 6.4.4 Erfahrung mit der Verwendung des Modells

Bei der Betrachtung des Modells in einem immersiven Visualisierungssystem wird einerseits schnell deutlich, daß die Verwendung sehr großer Mengen von Daten nur einen begrenzten Sinn macht. Auch wenn die Markierungen sich nicht gegenseitig verdecken, so hat der Benutzer im virtuellen Modell nicht die Möglichkeit, sich mit Hilfe der Schärfentiefe auf einen bestimmten Bereich im inneren des Bebenschwarmes zu konzentrieren.

Die hier vorgestellte zeitabhängige Visualisierung läßt sich andererseits gut nutzen, um schnell einen ersten Überblick von der zeitlichen Ausbreitung der Mikroben zu bekommen. Die Stereodarstellung ist hierfür wichtig, da es sonst schwierig ist, die räumliche Tiefe der Datenpunkte zu kontrollieren; denn bedingt durch die relativ kleine Darstellung der Markierungen fallen weitere Kontrollfaktoren für die räumliche Tiefe, wie z.B. das Verdecken der hinteren Objekte durch die vorderen, aus. Eine zu kleine Darstellung der Markierungen für die Bebenlokationen erschwert zudem die Wahrnehmung der Farbkodierung.

In der CyberStage ergab sich das Problem, daß die Benutzerschnittstelle an einem festen Platz dargestellt werden mußte. Die Verwendung der virtuellen Toolbar, welche schon bei der Darstellung der Radargramme zum Einsatz kam, war hier nicht möglich, da nicht nur Werkzeuge ausgewählt, sondern auch Schalter zur Verfügung gestellt werden sollten. Die Benutzerschnittstelle wurde am Eingang der CyberStage auf dem Boden dargestellt. Die Bedienung erfolgte vom jeweiligen Standort aus mit dem Zeigestrahls, was umständlich ist und die Geschwindigkeit der Bedienung bremst. Besser wäre es, wenn die Schnittstelle direkt am Standpunkt zur Verfügung stehen würde, zum Beispiel mit Hilfe eines Palm-Computers.

## 6.5 Visualisierung eines Silberbergwerkes

### 6.5.1 Allgemeines

Wie in Abschnitt 3.2 über die Rahmenapplikation AVANGO bereits erläutert, ist das Laden von geometrischen Modellen sehr einfach zu bewerkstelligen. Der Benutzer hat in diesem Fall aber nur ein statisches Modell zur Verfügung, das im Raum fixiert ist und mit dem keinerlei Interaktionen möglich sind. Um das interaktive Betrachten von geometrischen Modellen zu ermöglichen, wurde im Rahmen dieser Arbeit eine Anwendung entwickelt, welche die Geometrien aus Dateien einliest und darstellt. Der Benutzer kann das Modell dann mit Hilfe der Cubic-Mouse rotieren und gleichzeitig mit einem Werkzeug in der anderen Hand bewegen. Außerdem kann er auch hier, wie bereits im Abschnitt 5.3 über das hydrogeologische Versuchsfeld Krauthausen beschrieben, mit Hilfe von acht Ecksteinen einen Unterbereich definieren, der dann entsprechend vergrößert wird, so daß er den Visualisierungsraum der Workbench möglichst optimal nutzt. Allgemein wurde diese Anwendung zur Visualisierung von geometrischen Modellen unter der Maßgabe entwickelt, daß sie exemplarisch die verteilte Visualisierung und die Kooperation zwischen geographisch verteilten Geowissenschaftlern ermöglichen soll.

### 6.5.2 Der Beispiel-Datensatz Schauinsland-Bergwerk

Als Beispiel-Datensatz kam das geometrische Modell eines Bergwerkes zum Einsatz, welches von Ganter (1995) erstellt wurde und nahe Freiburg im Berg "Schauinsland" im Schwarzwald liegt. An dieser Lokation wurde bereits vor fast 800 Jahren mit dem Abbau von Silber begonnen. Danach wurden Blei und bis zur Grubenstillegung Zink abgebaut. Die Erzgänge entstanden durch hydrothermale Ablagerung entlang von aufgerissenen Spalten, stehen nahezu saiger und weisen ein Streichen von 10 - 30 bzw. 50 - 60 Grad auf.

In jüngerer Zeit wurde das Bergwerk von der Freiburger Energie- und Wasserversorgung (FEW) für die Trinkwassergewinnung genutzt. Das Sickerwasser tritt aus Klüften und Störungen in die Feldstrecken und Schächte ein, von wo aus es umgeleitet und gewonnen werden kann. Da teilweise Schwermetalle aus den Erzen in Lösung gehen, können diese Wässer jedoch kontaminiert sein. Somit ist die Kenntnis des

Fließweges sehr wichtig, damit verhindert werden kann, daß sich das kontaminierte Wasser mit dem reinen, als Trinkwasser geeigneten, mischt. Um für die Abschätzung der Fließwege einen Überblick über den recht komplexen dreidimensionalen Aufbau der Grube zu bekommen, wurde am Geologischen Institut der Universität Freiburg aus einer Vielzahl von Rissen und Plänen ein geometrisches Modell des Bergwerkes erstellt (Ganter, 1995). In Abbildung 6.20 ist das Modell der Grube zu sehen, wie es auf der Workbench erscheint. Die Feldstrecken verlaufen entlang des Streichens der Erzkörper (NNE/SSW und NE/SW), welche an zwei Stellen innerhalb des Bergwerkes zu sehen sind. Der Abbau erfolgt dann entlang der Erzkörper von Feldstrecke zu Feldstrecke. Entlang der Feldstrecken sind als kleine Flächen die Richtungen von Störungen eingetragen, welche dort gemessen wurden. Entlang großer Bewegungsflächen haben sich sogenannte Ruschelzonen gebildet, die mehrere Meter mächtig sind und in denen das Gestein stark zerrieben und daher wenig standfest ist. Die Wichtigste von ihnen ist die bis zu 20 m mächtige Schumacher-Ruschelzone, die ebenfalls in der Abbildung zu sehen ist.

Das Modell wurde ursprünglich für die Darstellung mit Hilfe der Visualisierungs-umgebung Geo3View (Klein & Ramshorn, 1991; Lindenbeck & Ulmer, 1995) erstellt und ist danach von Christoph Lindenbeck und Heike Ulmer in das VRML Format umgewandelt worden. Als VRML-Modell kann es von der Homepage des Geologischen Instituts Freiburg heruntergeladen werden (Freiburg, 2001). Um das Modell für diese Arbeit zu verwenden, wurde mit Hilfe des Programms Creator von der Firma Multigen-Paradigm der Szenengraph modifiziert und für die einzelnen Polygone die Flächennormale berechnet und gesetzt. Die in Abbildung 6.20 gezeigten Stollen decken einen Bereich von 550 m x 900 m Grundfläche und eine Höhe von 600 m ab.

### 6.5.3 Zum Thema verteiltes Arbeiten

Eine Anwendung, welche die Möglichkeiten von AVANGO nutzt, den Szenengraphen an mehreren Lokationen zu spiegeln, wurde bereits für die Erdöl-Datensätze im Rahmen des VRGeo-Projektes entwickelt (Springer et al., 2000). Auf diese Weise haben die Benutzer jeweils die Möglichkeit, das Modell in derselben Orientierung zu betrachten, während die Schnittebenen durch die 3D-Seismik synchron gehalten werden. Um auf einzelne Objekte zu zeigen, kann der Zeigestab verwendet werden, dessen Strahl jeweils auch an der anderen Lokation sichtbar ist. Um wirklich die geographisch verteilte Kooperation zu ermöglichen, sind aber einige weitere Featu-

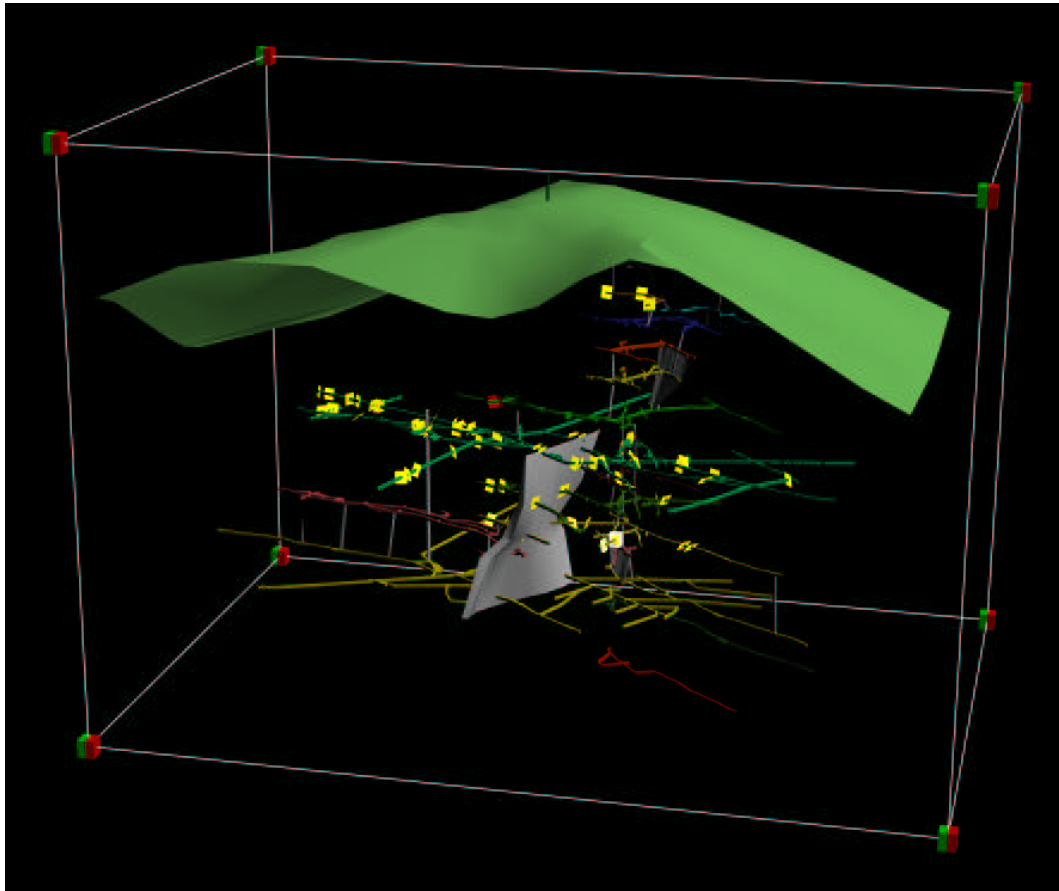


Abbildung 6.20: Das Modell des Schauinsland-Bergwerkes. Die Dimensionen des Bergwerkes sind in Realität ungefähr 550 m x 900 m x 600 m. Die grüne Fläche stellt die Topographie des Schauinslands dar, die graue Fläche die Schumacher-Ruschelzone. Die kleinen gelben Flächen entlang der Schächte zeigen die Richtung der dort gemessenen Störungen an.

res notwendig. Geophysikalische Messungen und geologische Geländedaten sind oft nicht eindeutig, so daß dem Interpretier eine gewisse Freiheit bei der Modellierung des Untergrundes bleibt. Daher ist es notwendig, daß nicht nur die zugrunde liegenden Daten als Modell verteilt werden, sondern daß die Benutzer auch die Möglichkeit haben, sich über das mentale Modell, welches sie sich jeweils anhand der Daten gebildet haben, austauschen zu können. Hierfür müssen Werkzeuge bereitgestellt werden, um zum Beispiel die Gestik zu ersetzen, der man sich bei Diskussionen über real vorhandene Sachverhalte bedient. Will man seinem Gesprächspartner zum Beispiel die eigene Auffassung zur Lage einer Fläche erklären, so verwendet man in der realen Welt meist die Handfläche, um die Richtung anzudeuten, oder man zeigt mit dem Finger auf bestimmte Objekte, um zu erklären, was gemeint ist. Diese Metho-

de der Wissensvermittlung funktioniert in einer virtuellen Umgebung in der Regel nicht. Dies gilt auch bei der Benutzung von VR für lokale Diskussionen, da mehrere Benutzer, welche vor einem Display stehen, durch die perspektivische Verzerrung die Objekte an verschiedenen Orten wahrnehmen. Um die verteilte Kooperation zu ermöglichen, sind also einige Grundanforderungen zu stellen:

- Das Modell und sein Zustand müssen verteilt visualisiert werden.
- Es muß möglich sein, einzelne Informationsträger im Datensatz zu markieren, einzugrenzen oder schnell auf sie zu zeigen, um den anderen Personen klarzumachen, von welchen Objekten man gerade redet.
- Man muß geologische Sachverhalte schnell andeuten können, also zum Beispiel die Lage von Schichtflächen und Verwerfungen oder Position und Verlauf einer Faltenachse.
- Haben die Benutzer die Möglichkeit, das Modell aus unterschiedlicher Richtung zu betrachten, so müssen sie wissen, welche Blickrichtung die anderen Benutzer jeweils einnehmen.

Bei der Anwendung, wie sie hier realisiert wurde, kann nur einer der Benutzer die Orientierung des Modells über die Cubic-Mouse steuern. Werden also zwei Workbenches miteinander gekoppelt, betrachten beide Benutzer das Modell ungefähr aus derselben Richtung, können aber durch ihre Bewegung ihre Perspektive selbst bestimmen. Um Objekte zu markieren, läßt sich ihre Visualisierung verändern. Wie in Abbildung 6.21 zu sehen, werden bei ausgewählten Objekten die einzelnen Polygone als Gitter dem eigentlichen Objekt überlagert und die Bounding-Box des Objekts angezeigt. Um die Polygongitter möglichst gut sichtbar zu machen, werden die Farben jeweils in das HSV-Farbmodell (siehe z.B. Foley et al., 1990) umgewandelt und das Gitter je nach Helligkeit in Schwarz oder Weiß visualisiert. Wird die Bounding-Box nicht angezeigt, was vorkommen kann, wenn diese außerhalb des betrachteten Unterbereiches zum Liegen kommt, so ist das Gitter allerdings der Erfahrung nach trotzdem schwer zu erkennen. Um die Diskussion über Gedanken zum Untergroundaufbau zu unterstützen, hat der Benutzer die Möglichkeit, einzelne Marker wie hier als Beispiel eine Fläche mit Normale, einen Pfeil oder andere Zeichen in das Modell einzufügen und zu bewegen. Wesentlich besser wäre jedoch, wenn der Benutzer in dieser Umgebung auch noch eine schnelle Methode zur Freiflächenmodellierung hätte. Insbesondere wenn die Netztopologie variabel und interaktiv veränderbar sein



soll, wäre der hiermit verbundene Entwicklungsaufwand im Rahmen dieser Arbeit nicht zu bewerkstelligen gewesen.

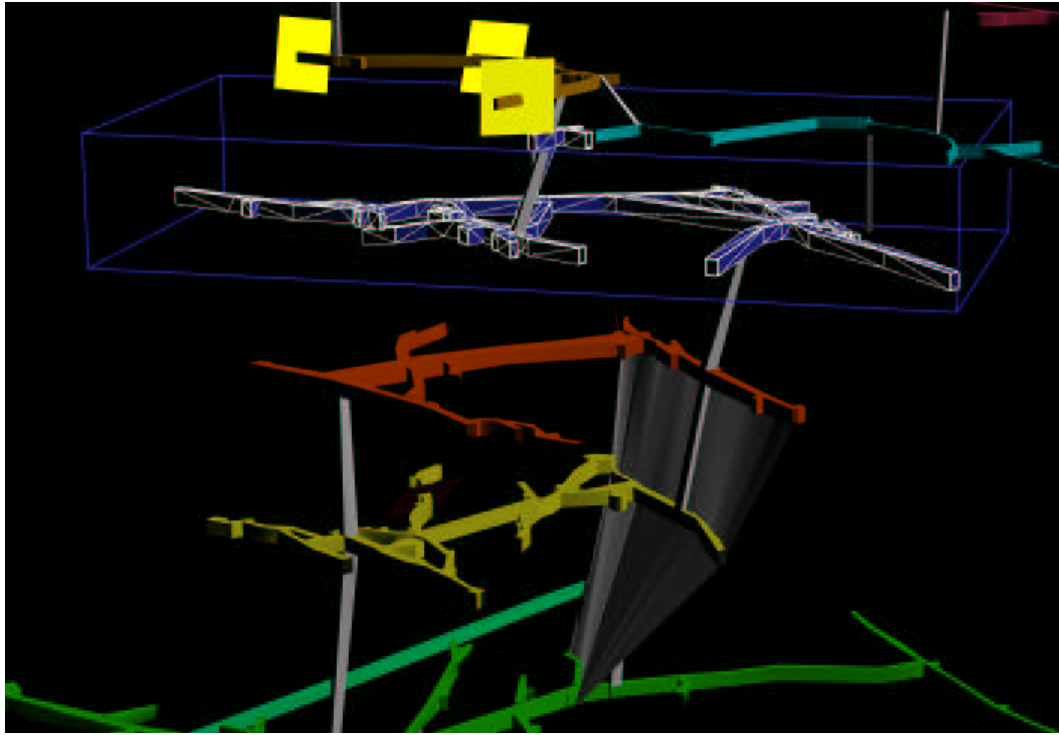


Abbildung 6.21: Die Abbildung zeigt anhand der oberen blauen Strecke, wie in dem hier vorgestellten System ein markiertes Objekt hervorgehoben wird, um die Aufmerksamkeit darauf zu lenken. In diesem Fall sind sowohl das einhüllende Rechteck als auch das Gitter angezeigt. Darunter kann man noch einmal sehen, wie der bergmännische Abbau erfolgte. Die graue Fläche markiert den Verlauf eines Erzkörpers. Die Feldstrecken verlaufen entlang von dessen Streichen. Der eigentliche Abbau erfolgte dann zwischen den Feldstrecken entlang des Erzkörpers.

#### 6.5.4 Erfahrung mit dem Modell

Die Anwendung stellt nur sehr rudimentäre Interaktionsmöglichkeiten bereit, welche sich aber bei Bedarf ohne weiteres erweitern ließen. In diesem Fall wurde das gesamte geometrische Modell aus einer Datei eingelesen, so daß der Szenengraph nicht aus AVANGO-Objekten aufgebaut ist und die Belegung der Felder den Zustand des Modells definiert. Daher ist es nur schwer möglich, Veränderungen in dem Modell durchzuführen, welche sich dann am anderen Ort widerspiegeln. Die Flexibilität diesbezüglich kann aber sehr einfach gesteigert werden, indem die einzelnen

Feldstrecken in verschiedenen Dateien gespeichert werden und der Szenengraph über das Scheme Interface von AVANGO aufgebaut wird.

Insgesamt eignet sich das Schauinsland-Bergwerk sehr gut für die Visualisierung in einer virtuellen Umgebung. Es hält die an ein Modell zu stellenden geometrischen Randbedingungen ein und ist ausreichend komplex, um die Vorteile einer echt räumlichen Visualisierung zu zeigen. Die Anwendung von VR erlaubt hier eine übersichtlichere Darstellung der Daten als sie am Bildschirm möglich wäre und könnte zum Beispiel insbesondere dann von Vorteil sein, wenn geophysikalische Untersuchungen innerhalb des Bergwerkes durchgeführt würden.

# Kapitel 7

## Diskussion

### 7.1 Möglichkeiten der Anwendung

Im vorhergehenden Kapitel wurden verschiedene Beispielanwendungen vorgestellt, die im Rahmen dieser Arbeit erstellt wurden. Während ihrer Implementierung konnten Erfahrungen mit der Entwicklung und Anwendung von Visualisierungssoftware zur Darstellung geophysikalischer Daten und geologischer Modelle gewonnen werden, welche das interaktive Arbeiten mit diesen Daten unter Anwendung der Methode der Virtuellen Realität erlaubt. Andere Software-Prototypen, die von Mitarbeitern am Institut für Medienkommunikation erstellt wurden, verschafften zusätzliche Einblicke in verschiedene Anwendungsbereiche wie Visualisierung von Strömungssimulationen und Crashtests. Ferner bestand die Möglichkeit, die VR-Laboratorien am Virtual Environment Technology Laboratory der Universität Houston, bei der Atlantik Richfield Company (ARCO) in Plano/Texas, am Institut für Geoinformatik in Münster und bereits 1997 am National Center for Super Computer Applications (NCSA) der University of Illinois zu besuchen, wo jeweils die Möglichkeit gegeben war, weitere Beispielanwendungen auf den jeweiligen Displaysystemen zu betrachten. Daher konnte bei der Ausarbeitung dieser Arbeit auf Kenntnisse von am Institut für Medienkommunikation nicht vorhandenen Displaysystemen zurückgegriffen werden.

Insgesamt hat sich während dieser Arbeit gezeigt, daß es verschiedene Arbeitsbereiche in der Geologie und Geophysik gibt, wo sich VR einsetzen läßt. Es stellt sich aber die Frage, wo und in welcher Form dieser Einsatz sinnvoll ist. Baut man auf der Erfahrung auf, die mit dem Einsatz der Software-Prototypen bei dieser Arbeit

gesammelt wurden, so ist zu beachten, daß die in dieser Arbeit eingesetzte Visualisierungshardware, welche den am Markt üblichen Standardsystemen entspricht, ursprünglich für andere Zwecke entwickelt worden ist. Die Standardapplikationen für VR waren lange Zeit Entertainment und Designstudien (z.B. Architektur und Automobilbau). Bei diesen Anwendungen geht es darum, für den Betrachter die virtuelle Umwelt oder einzelne Objekte in Lebensgröße "erfahrbar" zu machen, und es war in der Regel ausreichend, wenn der Benutzer in der Lage war, sich durch das Modell zu navigieren. Bei in jüngerer Zeit hinzugekommenen Anwendungen werden weitere spezielle Interaktionen benötigt, die aktuelle Forschungsthemen sind. Zum Beispiel sollen bei der Produktionsplanung die Bewegungsabläufe der Testpersonen bei der Arbeit mit den virtuellen Produktionsanlagen unter möglichst realistischen Bedingungen nachgestellt werden. Dies erfordert eine möglichst vielseitige direkte Interaktion mit den Objekten im Raum, wie es auch in der realen Welt der Fall wäre.

Im Gegensatz zu den vorgenannten Anwendungen handelt es sich bei der Darstellung geologischer Modelle und geophysikalischer Daten meist um abstrakte Visualisierungen, da die Objekte in Realität so meist nicht sichtbar sind. Ferner werden in den Geowissenschaften oft großräumige Fragestellungen bearbeitet, die sich in den virtuellen Umgebungen nur stark verkleinert darstellen lassen. Es kommt hier also nicht auf eine möglichst "realistische" Darstellung in "Lebensgröße" an, sondern darauf, räumliche Zusammenhänge sichtbar zu machen und den Austausch über die Datensätze und die sich aus ihnen ergebenden Interpretationsmöglichkeiten zu fördern. Für diese Art von Aufgaben ist es nicht zwingend notwendig und je nach Fragestellung oft auch nicht von Vorteil, mit direkter Interaktion im Raum das Arbeiten in einer virtuellen Umwelt nachzustellen. Daher hat man, je nach praktischen Gesichtspunkten, die Möglichkeit, direkte und indirekte Interaktionsformen miteinander zu mischen. Um VR in den Geowissenschaften sinnvoll zu nutzen, wäre es also notwendig, auch die Hardware speziell für die entsprechenden Aufgabenstellungen anzupassen und zu entwickeln. Hinzu kommt, daß die am Fraunhofer Institut für Medienkommunikation vorhandenen Systeme von verschiedenen Arbeitsgruppen für eine Vielzahl von Aufgabenstellungen genutzt werden und daher darauf ausgelegt sind, ein möglichst offenes System bereitzustellen, das an eine Vielzahl von Fragestellungen adaptiert werden kann. Neue Interaktionsformen werden meist über die Software realisiert.

Es gibt viele verschiedene Fragestellungen, bei denen ein Einsatz von immersiven Visualisierungssystemen denkbar erscheint, zum Beispiel in der Forschung oder

in größeren Büros für angewandte Geologie. Um die verschiedenen Einsatzmöglichkeiten für 3D-Visualisierung und VR bei einer umfangreichen dreidimensionalen geologisch-geophysikalischen Untersuchung zu zeigen, ist in Abbildung 7.1 exemplarisch ein möglicher Workflow dargestellt. Die Bereiche, in denen 3D-Visualisierung eingesetzt werden kann, sind gelb hinterlegt und sie stellen jeweils auch potentielle Einsatzgebiete für die Verwendung von VR-Techniken dar. In rot sind jeweils die iterativen Arbeitsabläufe, die bei der Auswertung der Daten oft auftreten, angezeigt.

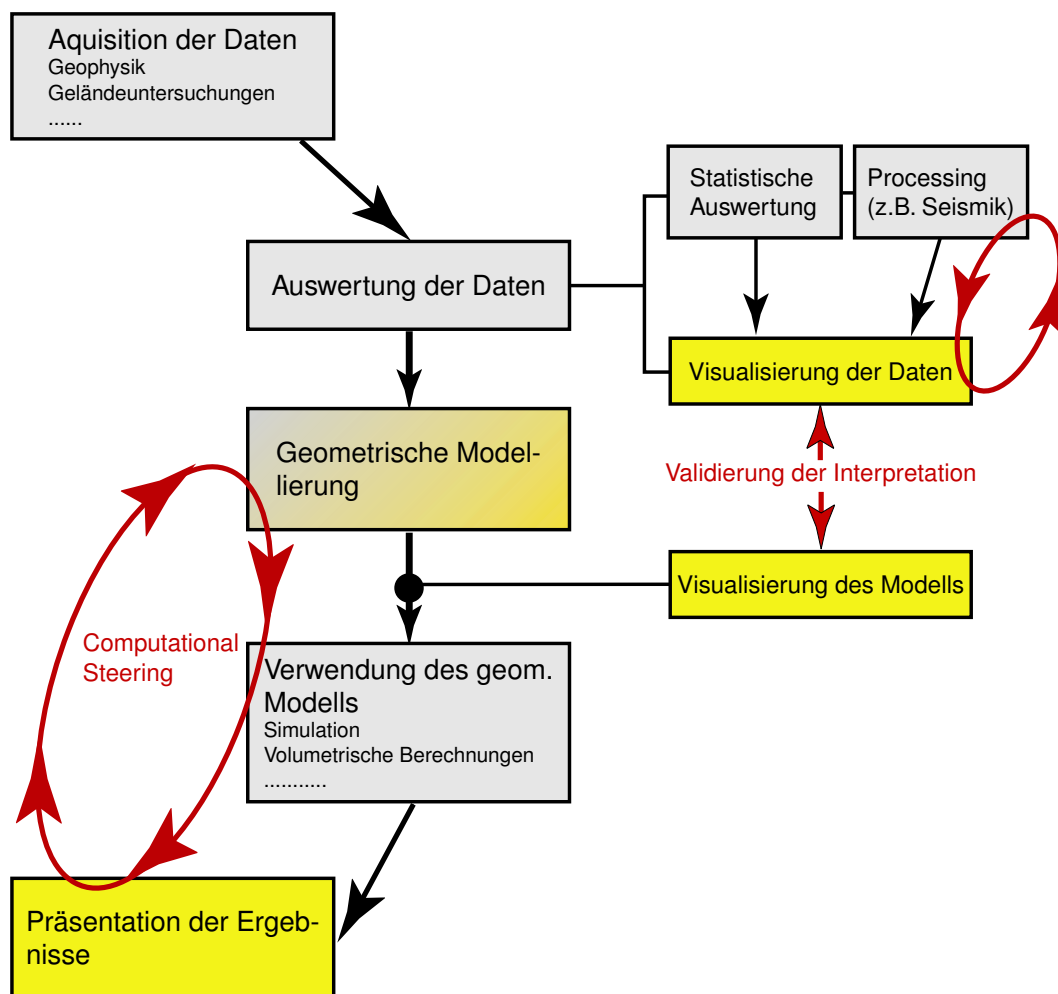


Abbildung 7.1: Möglicher Arbeitsablauf für eine umfangreiche geophysikalische Untersuchung in der angewandten Geologie. Die Einsatzmöglichkeiten für 3D-Visualisierung und VR sind gelb hinterlegt. In rot sind jene Arbeitsabläufe gekennzeichnet, welche häufig iterativ ablaufen.

Die im vorhergehenden Kapitel beschriebenen Beispielanwendungen decken von den in der Abbildung 7.1 aufgezeigten Einsatzbereichen ein weites Spektrum ab:

- Als Beispiele für die Visualisierung der Daten können in dieser Arbeit die Darstellung des dreidimensionalen Datensatzes vom hydrogeologischen Versuchsfeld in Krauthausen und die räumlich-zeitliche Darstellung der seismologischen Daten vom geothermischen Reservoir in Soultz-sous-Forêts dienen.
- Die Validierung der Interpretation gegen die zugrunde liegenden Daten ist durch den Software-Prototyp des VRGeo-Projektes abgedeckt.
- Bei der Darstellung des Modells vom Schauinsland-Bergwerk handelt es sich um die Visualisierung eines komplizierten geometrischen Modells.
- Ein Einblick, wie die direkte Interpretation von Daten in einer solchen Umgebung aussehen kann, läßt sich durch den Prototyp zur Visualisierung und Interpretation von Georadar-Daten vermitteln. Einen Prototypen zu entwickeln, der die vollständige geometrische Modellierung in der virtuellen Umgebung erlaubt, wäre im Rahmen dieser Arbeit bei weitem zu aufwendig gewesen. Die während der Arbeit gesammelten Eindrücke reichen jedoch aus, um hier zu einer Beurteilung zu kommen.

Im folgenden soll aufgeteilt nach verallgemeinerten Anwendungsszenarien diskutiert werden, wo in den Geowissenschaften Displaysysteme für VR sinnvoll eingesetzt werden könnten.

### 7.1.1 Visuelle Datenanalyse

Wie sich bei der Verwendung der Software-Prototypen gezeigt hat, kann die Visualisierung von Datensätzen und Simulationsergebnissen und die Präsentation der daraus generierten Modelle mit den am Markt erhältlichen Systemen ohne Probleme vorgenommen werden. Da die Verwendung von VR eine übersichtlichere Darstellung komplexer Datensätze gestattet, als dies am Bildschirm möglich ist, eignen sie sich somit gut zur visuellen Validierung der Interpretation. Wie bereits erwähnt, müssen sich die Datensätze allerdings auf der einen Seite geometrisch für eine Repräsentation in einer virtuellen Umgebung eignen, auf der anderen Seite aber ausreichend komplex sein, damit sich der Einsatz der zur Zeit noch sehr teuren und aufwendigen Technik lohnt. Die Reservoir-Datensätze aus der Erdölindustrie, die im VRGeo-Projekt Verwendung fanden, halten sich sehr gut an diese Randbedingungen und sind darum eindeutig ein gutes und dankbares Anwendungsbeispiel. Von den anderen

Datensätzen erfüllt das “Digitale Bergwerk“ noch am ehesten diese Randbedingungen.

Insgesamt deuten die Erfahrungen und Beobachtungen, die während der Ausarbeitung dieser Dissertation gemacht wurden, darauf hin, daß vergleichbare Datensätze außerhalb der Erdölindustrie, also zum Beispiel im akademischen Umfeld, eher selten bearbeitet werden. Die Erhebung von so umfangreichen Datenmengen, wie sie bei der Exploration und Reservoir-Charakterisierung vorkommen, ist sehr aufwendig und kommt aus finanziellen Gründen in der akademischen Forschung oder bei Fragestellungen der Ingenieur- und Hydrogeologie fast nicht vor. Bei den hier auftretenden Datenmengen ist für die mentale Modellbildung daher die Unterstützung mit VR nicht erforderlich bzw. es reicht, eine gute 3D-Visualisierung mit einer Standardsoftware wie z.B. Gocad einzusetzen. Aus diesem Umstand folgt aber nicht, daß die Verwendung von VR für derartige Aufgabenstellungen überhaupt keine Vorteile bringt. Sie ist jedoch nicht zwingend notwendig, da sich die Probleme mit den standardmäßig zur Verfügung stehenden Mitteln lösen lassen, was im Prinzip auch für die in dieser Arbeit zur Beispielvisualisierung verwendeten Datensätze gilt. Aus ökonomischer Sicht ist der finanzielle Umsatz bei Untersuchungen in der Ingenieur- und Hydrogeologie oder in der Forschung und Lehre nicht ausreichend hoch, als daß sich mit der zur Zeit noch sehr teuren Technologie Einsparungen erreichen ließen. So wurden denn auch bei der Diskussion mit Geowissenschaftlern, die zu Gast am Institut für Medienkommunikation waren, die hohen mit VR verbundenen Kosten immer wieder als Hauptargument gegen die Verwendung dieser Technologie angeführt. Dieses Argument wird durch den rapiden Preisverfall im Hardwaremarkt jedoch zunehmend entkräftet werden. Da es zudem sehr schwer ist, ein auf alle Fragestellungen anpassbares Softwaresystem zu entwickeln, stellt der Mangel an Spezialisten und Know-How ein weiteres Problem dar.

### 7.1.2 Geometrische Modellierung

Bei der Interpretation eines Datensatzes will man die Informationen, die darin enthalten sind, nicht nur sichtbar machen, sondern nach Möglichkeit auch festhalten können. Ein kritischer Punkt in dem Workflow der Abbildung 7.1 ist daher die geometrische Modellierung, die immer notwendig ist, um von den Daten zu einem Modell zu kommen. Somit ist es wünschenswert, auch die geometrische Modellierung in virtuellen Umgebungen vornehmen zu können. Die Interpretation der verschiede-

nen Datensätze könnte dann zeitgleich mit der Validierung stattfinden und aus dem iterativen Prozeß *Interpretation* → *geometrische Modellierung* → *Validierung* → *Interpretation* → ... würde ein interaktiver Prozeß. Die Anwendung der in dieser Arbeit eingesetzten immersiven Visualisierungssysteme zur Erstellung eines geometrischen Modells ist jedoch kritisch und sie werden hierfür bisher auch kaum benutzt. Theoretisch sollte die Darstellung in Stereo und die schnelle Interaktion im dreidimensionalen Raum hier besonders hilfreich sein, in der Praxis sind diese Systeme aber, wie bereits dargestellt, eher für die Visualisierung im Sinne einer Präsentation ausgelegt. Um ein geometrisches Modell zu konstruieren, müssen an das System und die zur Verfügung stehenden Interaktionsformen weitere Anforderungen gestellt werden, welche natürlich stark von den zur Verfügung stehenden unterstützenden Algorithmen, wie zum Beispiel Hilfsgitter und magnetische Ankerpunkte, abhängen:

- Für die einzelnen Ankerpunkte des Modells muß die Position und die Richtung exakt festlegbar sein. Bei Desktop-Standard-Systemen ist dies durch Eingabe über die Tastatur möglich. Die Präzision, die man mit der Cubic-Mouse oder mit dem Stylus erreicht, genügt hier nicht. Gleichzeitig sollte es möglich sein, sehr schnell Position und Richtung anzudeuten (z.B. für Diskussionen).
- Trotz Stereo-Projektion ist es oft erforderlich, bestimmte Einstellungen aus verschiedenen Blickwinkeln zu prüfen. Der Arbeitsablauf, die Orientierung und Position eines Objektes zu ändern, sich um das Objekt herumzubewegen und danach die Veränderung weiter fortzuführen, muß daher flüssig möglich sein. Dies ist mit direkter Interaktion nicht möglich.
- Die Benutzerschnittstelle sollte trotz höherer Komplexität auch mit wenig Übung “blind“ bedienbar sein, damit der Betrachter sich voll auf den Datensatz konzentrieren kann. Dies ist bei Standard-Windows-Systemen in der Regel nicht der Fall und läßt sich durch spezielle Eingabegeräte, welche bei der Bedienung den Tastsinn ansprechen, wesentlich besser realisieren.
- Tätigkeiten, die sich besser im Zweidimensionalen ausführen lassen, müssen entsprechend unterstützt werden.
- Arbeitsergonomische Aspekte müssen stärker berücksichtigt werden als bisher, um auch das Arbeiten über längere Zeit zu ermöglichen.

Für viele Aufgaben sind externe Eingabegeräte, die indirekt bedient werden, besser geeignet als die direkte Interaktion, wie sie zum Beispiel mit dem Stylus im Raum



erfolgt. Ein für die Modellierung geeignetes System wird also sowohl für die direkte als auch für die indirekte Interaktion Eingabegeräte zur Verfügung stellen. Kommerziell erhältliche Beispiele für Eingabegeräte zur indirekten Interaktion im Raum wären die SpaceMouse und der SpaceBall der Firma 3Dconnexion (siehe Abbildung 7.2). Ein weiteres Beispiel ist die Steuerung, wie sie bei einer Mikrosonde der Firma CAMECA verwendet wird: Mit Hilfe einzelner Rädchen auf einer Steuerkonsole werden die Bewegungen entlang der drei Raumachsen durchgeführt ( $z$ -Änderung bedeutet, an der Mikrosonde in das Bild hineinzuzoomen). Ist die Abbildung von der Drehung an den Einstellrädchen auf die Bewegung nichtlinear, so läßt sich die Position wahlweise sehr schnell oder sehr genau steuern. Auch für das Festlegen der Richtung ließe sich ein spezielles Eingabegerät entwickeln, das die schnelle und intuitive Spezifizierung von Streichen und Fallen erlaubt. Zum Beispiel ließe sich die Cubic-Mouse auf einem professionellen Photostativ befestigen. Ein solches Gerät verfügt in der Regel über zwei Skalen, welche die Drehung und den Neigungswinkel (also im übertragenen Sinne Streichen und Fallen) anzeigen. Für das schnelle Festlegen der Daten, wie man es zum Beispiel bei Diskussionen benötigt, kann dann zusätzlich die direkte Interaktion im Raum verwendet werden. Von Vorteil wäre auch eine getrennte Steuerung der Transformation des Gesamtmodells und von einzelnen Punkten und Werkzeugen innerhalb des Modells. Dies ist in etwa analog zu der Möglichkeit beim Handwerken, das Werkstück auf der Werkbank mit vorgegebener Orientierung einzuspannen (Steuerung des Gesamtmodells) und dann einzelne Teile zu bearbeiten. Die Trennung dieser beiden Steuerungen würde das "blinde" Arbeiten erleichtern. Sehr gut würde sich auch ein kraftrückgekoppeltes Gerät wie das Phantom (SensAble Technologies, 2001) eignen, da dieses erlaubt, die Bewegungsfreiheit des Benutzers bei der Interaktion im Raum einzuschränken. So kann er sich bei der Interpretation eines Schnittes im Raum auch nur auf diesem bewegen, was der Hand einen zusätzlichen Halt gibt und die Präzision der Interaktion erhöht.

Für Aufgaben, die sich besser in 2D erledigen lassen, wie zum Beispiel das Interpretieren einzelner Radargramme, ist es sinnvoll, einen oder zwei zusätzliche Bildschirme als 2D-Interface in das System zu integrieren. Im Falle des Beispiels für die Radardaten könnte der Benutzer dann im 3D-Modell einen bestimmten Schnitt aussuchen, welcher auf einem Bildschirm angezeigt wird. Während er das Radargramm auf dem Bildschirm bearbeitet, werden alle Schritte, wie zum Beispiel gepickte Reflektoren oder die Position der Maus, auch in dem 3D-Modell angezeigt. So kann der Benutzer gleichzeitig die Kontrolle über den dreidimensionalen Zusammenhang behalten. Ein zweiter Monitor ließe sich nutzen, um Eingaben über die Tastatur zu machen oder weitere Interaktionen wie das Definieren der Farbtabelle vorzuneh-



Abbildung 7.2: Der SpaceBall und die SpaceMouse der Firma 3Dconnexion sind speziell für die 3D-Desktop-Interaktion entwickelte Eingabegeräte.

men. Unter Umständen kann es auch sinnvoll sein, eine 2D-Benutzerschnittstelle für zweidimensionale Elemente im Raum nutzbar zu machen. Für das Beispiel zur Interpretation von Georadar-Daten würde das bedeuten, daß der Benutzer im Raum ein einzelnes Radargramm auswählt, das er bearbeiten möchte. Die Steuerung des Trackers beim Digitalisieren der Reflektoren erfolgt dann indirekt über einen Joystick oder eine normale Maus auf dem Radargramm, welches aber nach wie vor im dreidimensionalen Raum dargestellt wird.

Obwohl die Verwendung eines solchen Systems in vielen Fällen die Arbeit sehr erleichtern würde, muß man sich darüber im klaren sein, daß viele Vorgänge bei der geometrischen Modellierung keine stereoskopische Visualisierung oder VR erfordern. Zum Beispiel mußten bei der geometrischen Modellierung des Schauinsland-Bergwerkes vor allen Dingen die Grubenrisse digitalisiert werden. Diese Grundrisse wurden dann mit Hilfe von speziellen Programmen zu 3D-Körpern umgewandelt und die einzelnen Schächte zu einem Gesamtmodell zusammensetzt (Ganter, 1995). Ein anderes Beispiel ist die Erstellung eines geometrischen Modells von der Geologie des Tagebaus Bergheim. Dort lagen insgesamt 19 nebeneinander liegende Schnitte vor, die digitalisiert wurden (Pongratz, 1999). Die Modellierung fand aber jeweils zwischen zweien dieser Schnitte statt, was die dreidimensionale Komplexität der Konstruktion wesentlich vereinfacht. In beiden Fällen war es also bei der Konstruktion des Modells nicht unbedingt notwendig, Stereo-Visualisierung einzusetzen, da man die Komplexität des gesamten Modells auf einen kleinen Unterbereich reduzie-

ren konnte, der gerade bearbeitet wird. Bei der Betrachtung der in beiden Fällen komplizierten Gesamtmodelle würde eine echte räumliche Darstellung jedoch Vorteile bringen.

Ein wichtiger Punkt bezüglich der Arbeitsergonomie ist auch die Tatsache, daß in den hier verwendeten Displaysystemen das Arbeiten nur im Stehen möglich ist. Insbesondere an der Holobench ist es zudem erforderlich, sich oft nach vorne zu beugen, um mit weiter entfernten Teilen des Modells zu arbeiten. Dieses Thema wurde auch von den Mitgliedern des VRGeo-Konsortiums angesprochen, läßt sich aber an der Workbench nicht realisieren, da der Platz unter dem Projektionstisch vom Spiegel benötigt wird. Will man VR aber wirklich zum Arbeiten einsetzen, muß eine Lösung für dieses Problem gefunden werden.

Wie läßt sich VR aber bei der Interpretation sinnvoll einsetzen? Eine sehr einfache Arbeitsplatzgestaltung, die Stereo-Visualisierung bei der Datenauswertung ermöglicht, wäre ein normaler Arbeitstisch mit zwei flach oder leicht schräg eingebauten Flachbildschirmen (noch besser: Touch-Screens) und einer großen Stereo-Projektion auf seiner Rückwand. Die Ausstattung mit einem Tracking-System würde es dem Benutzer erlauben, direkt im Raum auf Objekte zu zeigen. Als zusätzliche Steuerelemente sollten aber 3D-Interaktionsgeräte aus dem Desktop-Bereich, wie die SpaceMouse und der SpaceBall, eingebunden werden. Ein solcher Arbeitsplatz ließe sich kostengünstig realisieren und würde eventuell bei komplizierten Fragestellungen bereits eine Verbesserung des bisherigen Arbeitsablaufes bedeuten.

### 7.1.3 Computational Steering

Mit einer Arbeitsumgebung, wie sie vorstehend beschrieben wurde, ist es einfacher und effektiver, völlig neue Geometrien im Raum zu definieren, die nicht mit der Unterstützung automatisierender Algorithmen aus den vorhandenen Daten generiert werden können, und diese Geometrien im Raum zu verändern. Dies kann zum Beispiel bei der Vorwärtsmodellierung geophysikalischer Prozesse sehr hilfreich sein. Ein Beispiel: Bei der gravimetrischen Modellierung wird der dreidimensionale Aufbau des Untergrunds unter Beachtung von vorliegenden Daten "geraten" und dann berechnet, ob das sich ergebende Schwerefeld an der Oberfläche den im Feld gemessenen Werten entspricht. Ist dies nicht der Fall, wird das Modell modifiziert und die Berechnung erneut durchgeführt. So nähert man sich mit der Zeit einer möglichen Lösung des Problems an. Die immer wiederkehrende Veränderung des 3D-Modells

ließe sich in einer für diese Aufgabe optimierten virtuellen Umgebung wesentlich schneller und einfacher durchführen als am Bildschirm. Hat man eine ausreichend schnelle Simulationssoftware zur Verfügung, so läßt sich die gravimetrische Modellierung als interaktives und spielerisches Arbeiten gestalten. Ähnliche Anwendungen ergäben sich auch in der Hydrogeologie, z.B. bei der Planung von Brunnen, welche interaktiv in der Region bewegt werden können, während simultan der sich jeweils ergebende Grundwasserspiegel berechnet und angezeigt wird.

#### **7.1.4 3D-Informationssysteme**

Viele der im Zweidimensionalen möglichen Operationen auf Karten, welche in einem Geografischen Informationssystem (GIS) wie zum Beispiel ArcView von der Firma ESRI (Environmental Systems Research Institute, Redlands, California) ausgeführt werden können und im entsprechenden Handbuch beschrieben sind (ArcView, 1996), lassen sich ins Dreidimensionale übersetzen. So kann man zum Beispiel Objekte suchen, die sich innerhalb eines Körpers befinden oder alle Objekte, die sich mit einem bestimmten Objekt überschneiden. Genau wie man auf einer Karte den Verlauf einer Pipeline definieren kann, um zu sehen, welche Grundstücke diese berührt (also mit welchen Objekten sie sich überschneidet), kann man auch den Verlauf von Bohrlöchern oder Bergwerks-Schächten im Raum festlegen und das GIS-System anzeigen lassen, welche verschiedenen geologischen Formationen oder Bestandteile eines Bergwerkes geschnitten werden. Hierbei ist die einfache und schnelle Interaktion im Raum hilfreich, um die entsprechenden Geometrien für die Abfrage zu generieren. Auch lassen sich sehr einfach Objekte im Raum markieren, für die dann Informationen angezeigt werden, wobei die angezeigten Informationen häufig aus Tabellen oder Diagrammen bestehen und daher für deren Anzeige dann noch eine 2D-Schnittstelle bereitgestellt werden muß. In der Praxis sind Anwendungen für 3D-Informationssysteme allerdings nicht besonders häufig.

#### **7.1.5 Einsatz für Ausbildung und Training**

Genau wie in der Erdölindustrie würden sich die großflächigen Projektionsdisplays auch in der Lehre gut eignen, um komplizierte dreidimensionale Fragestellungen zu diskutieren. Wegen der hohen Kosten werden VR-Systeme aber in der Lehre bisher kaum eingesetzt. Lediglich an sehr wenigen geowissenschaftlichen Einrichtungen, die

mit der Erdölindustrie zusammenarbeiten und von dieser finanziell unterstützt werden, sind die hierfür notwendigen technischen Einrichtungen vorhanden. So wird am Institut für Petroleum Engineering and Applied Geophysics der Universität Trondheim gerade ein VR-Labor eingerichtet, das dann auch in der Lehre eingesetzt werden kann und von der Firma Norsk Hydro finanziert wird. In den USA haben eine Reihe von universitären Einrichtungen das Geowall Konsortium gegründet, welches das Ziel hat, im Vorlesungssaal oder im Seminarraum Low-Cost VR einzusetzen, um geowissenschaftliche Sachverhalte zu vermitteln (Geowall Konsortium, 2001). Zu diesem Zweck wurden an verschiedenen Instituten einfache Projektionswände aufgestellt, bei denen Passiv-Stereo verwendet wird. Die Kosten pro System liegen je nach Konfiguration bei ca. 30.000 - 50.000 DM (incl. Computer). Eingesetzt werden sollen sie in verschiedenen Bereichen wie Hydrogeologie, Geodynamik und Astronomie.

Neben der besseren Veranschaulichung von komplizierten Vorgängen hat die Verwendung von virtuellen Umgebungen in der Lehre den Vorteil, daß sie ermöglicht, vom klassischen Frontalunterricht weg zu einer Unterrichtsform überzugehen, bei der die Diskussion im Mittelpunkt steht. So lässt sich zum Beispiel mit Hilfe der Interaktion im Raum schnell der Verlauf von Flächen skizzieren, was die Diskussion über den Untergrund vereinfachen würde. Ein entsprechender Software-Prototyp wurde am Institut für Medienkommunikation erstellt (Wesche & Droske, 2000). Der Benutzer zeichnet die Randkurven frei in den Raum, woraus dann die Oberfläche generiert wird. Für geologische Fragestellungen wäre es allerdings gut, wenn neben den Randkurven auch einzelne Ankerpunkte, welche interpoliert werden sollen, samt ihrer Flächennormale festgelegt werden könnten (das entspräche einem Aufschluß, bei dem das Schichtfallen bestimmt werden kann).

Sinnvoll wäre auch ein Einsatz von VR für das Training des dreidimensionalen Vorstellungsvermögens. So könnte man auf einer Projektionsfläche (z.B. dem Tisch bei der Holobench) das Kartenbild darstellen und auf einer anderen (z.B. der Wand der Holobench) das entsprechende 3D-Modell. Der Student könnte sich zuerst anhand der Daten im Kartenblatt ein mentales Modell des Untergrundes entwickeln und dieses dann am visualisierten 3D-Modell verifizieren. Durch Veränderungen im Modell kann er sehen, wie diese sich auf das Kartenbild auswirken. Das Ergebnis ist eine Unterstützung des Lernprozesses, wie man von den Daten, die man an der Oberfläche findet, auf die Strukturen im Untergrund schließt. Weitere Beispiele für die Anwendung von VR in der Lehre wären die Darstellung des Strahlenganges bei der Seismik in Abhängigkeit von der Auslage der Geophone, die gleichzeitige Darstellung

eines Untergrundmodells mit der sich daraus ergebenden Seismik oder die Darstellung der Grundwasserströmungen in einem geologischen Modell. Optimal wäre hier jeweils die Kopplung von Simulation und Modellierung, so daß der Student jeweils im Raum das Modell als Eingabeparameter ändern und dabei beobachten kann, wie sich dies auf die abhängigen Parameter auswirkt.

Was die Hardware betrifft, würde es für die meisten Anwendungen in der Lehre schon reichen, eine große Projektionsfläche mit hoher Auflösung zu verwenden, da Immersion und interaktives Arbeiten hier nicht notwendig sind. Daher könnte unter Umständen auf die Verwendung von teuren Hochleistungs-Graphikrechnern verzichtet und es könnte statt dessen handelsübliche PCs mit einer sehr guten Graphikkarte eingesetzt werden. Als Projektionssystem würde sich die Verwendung von passivem Stereo anbieten, da dann günstige Projektoren und Brillen (Polarisationsbrillen) verwendet werden können. Zu guter letzt darf man den Vorteil nicht unterschätzen, daß der Aufenthalt in einer solchen Lernumgebung als Erlebnis empfunden wird und so die dargestellten Zusammenhänge besser aufgenommen werden. Bei einem solchen Einsatz in der Lehre muß allerdings Rücksicht darauf genommen werden, daß viele Menschen farbenblind sind und manche auch nicht räumlich sehen können. Bei allem Potential, das die Verwendung von VR in der Lehre hat, muß ferner beachtet werden, daß vorher die entsprechenden Lehrmaterialien in Form von 3D- oder 4D-Modellen erstellt werden müssen, was einen erheblichen Aufwand bedeutet. Ein weiterer Einwand, der gegen den Einsatz spricht, ist die Tatsache, daß die Studenten bei ihrer Ausbildung gerade lernen sollen, aus Karten Schnitte zu konstruieren und die 3D-Geometrie des Untergrundes herauszulesen. Abseits der Lehre an Universitäten ließen sich VR-Systeme auch gut einsetzen, um Laien, die nicht aus dem geowissenschaftlichen Umfeld kommen, komplizierte Fragestellungen und Lösungsansätze zu vermitteln und auf diese Weise die interdisziplinäre Zusammenarbeit zu ermöglichen.

## 7.2 Resumee

Auch wenn es in dieser Arbeit nicht gelungen ist, eine VR-Applikation zu entwickeln, die den Nutzen von VR für den geowissenschaftlichen Bereich optimal aufzeigt, das heißt mit deren Hilfe Probleme gelöst werden können, die sonst nicht lösbar wären, so wird doch deutlich, daß es einige Bereiche gibt, in denen sich die Methode der virtuellen Realität trotzdem sinnvoll einsetzen läßt. Um flüssiges Arbeiten zu ermöglichen,

sind allerdings Fortentwicklungen in der Hard- und Software notwendig, die speziell für die jeweiligen geowissenschaftlichen Fragestellungen spezifisch sind. Die in dieser Arbeit verwendeten Displaysysteme eignen sich nur begrenzt für die Datenanalyse und stellen von der Hardware-Seite keine ausreichenden Interaktionsmöglichkeiten bereit, sie auch für die Interpretation und geometrische Modellierung zu verwenden. Der Schwerpunkt dieser Systeme ist eindeutig die Präsentation und Visualisierung von Daten. Hier ist ihre Anwendung besonders sinnvoll, wenn es um die Darstellung von zeitlich räumlichen Fragestellungen geht, da man mit der direkten Interaktion im Raum viel schneller auf das Geschehen reagieren kann und man aufgrund der stereoskopischen Visualisierung rascher räumliche Zusammenhänge begreift. Ein weiteres gut funktionierendes Einsatzgebiet ist die Validierung von geometrischen Modellen gegen die zugrunde liegenden Daten.

Ein Problem bei der Durchführung dieser Arbeit war, daß die notwendige Projektionstechnik und Rechnerhardware nur am Fraunhofer Institut für Medienkommunikation zur Verfügung stand. Aus organisatorischen Gründen war es daher nicht möglich, VR in den tägliche Arbeitsablauf von Geowissenschaftlern einzubinden, und die Zusammenarbeit mit ihnen hatte hauptsächlich den Charakter von Präsentationen der Technik anhand der hier vorgestellten Modelle. Somit läßt sich nur abschätzen, welche Vorteile sich durch die Verwendung von VR "im Kleinen" ergeben; auch wenn sich die meisten Fragestellungen ohne VR lösen lassen, so kann die Integration dieser Methode den Arbeitsprozeß oft effektiver gestalten und vereinfachen. Um sich also wirklich über die Vorteile der Integration von VR in den Arbeitsprozeß in anderen geowissenschaftlichen Bereichen als der Erdölindustrie klar zu werden, ist es notwendig, ein solches Labor in einem geowissenschaftlichen Umfeld zu betreiben, zum Beispiel an einem Universitätsinstitut oder an einer Großforschungseinrichtung. In einem geowissenschaftlichen Umfeld würde sich dann auch die Möglichkeit bieten, VR auf verschiedene Teilprobleme anzuwenden, die sich während einer Untersuchung ergeben. Hierfür wäre aber zuerst die Frage zu klären, ob in diesem Umfeld eine ausreichende Anzahl geometrisch komplizierter Probleme vorliegt, die sich für diese Methode eignen und ob das notwendige Know-How und Personal vorhanden sind, ein solches Visualisierungssystem zu betreiben.





# Abbildungsverzeichnis

2.1	Übersicht über die verschiedenen auf dem Markt erhältlichen Projektionssysteme, klassifiziert nach Konfiguration der Projektionen . . . . .	8
2.2	In dieser Arbeit verwendete Interaktionsgeräte . . . . .	9
2.3	Modell der CyberStage . . . . .	11
2.4	Die zweiseitige Workbench (Holobench der Firma TAN Projektionstechnologie GmbH & Co. KG in Düsseldorf) . . . . .	13
2.5	Die Wedge, ein Low-Cost Displaysystem . . . . .	16
2.6	Die “Haptic Workbench“ der Firma Reachin Technologies AB . . . . .	17
3.1	Diagramm für eine mögliche Prozeßkonfiguration bei der Visualisierung auf zwei Displays . . . . .	21
3.2	Darstellung des Szenengraphen bei AVANGO mit dem zugehörigen GUI . . . . .	25
3.3	Graphische Darstellung des Umfeldes bezüglich der Entwicklung der Software-Prototypen . . . . .	26
4.1	Darstellung von sechs Bohrlöchern zur Verdeutlichung, warum der Einsatz von VR in der Geologie und Geophysik sinnvoll sein kann . . . . .	31
4.2	Vergleich der Vorgehensweise zur Festlegung eines nach den Koordinatenachsen ausgerichteten Unterbereiches in 2D und in 3D . . . . .	32
4.3	Darstellung zum Konflikt zwischen der Scharfstellung der Augen und ihrer Konvergenz . . . . .	34
4.4	Die verschiedenen Zonen auf der zweiseitigen Workbench bezüglich der Interaktions- und Visualisierungsmöglichkeiten . . . . .	35
6.1	Visualisierung von 3D-Seismik mit drei senkrecht zueinander stehenden Schnitten . . . . .	49
6.2	Darstellung von 3D-Seismik mit Hilfe von Volumenvisualisierung . . . . .	49

6.3	Beispiel für die Darstellung von Bohrlochdaten . . . . .	51
6.4	Skizze von drei möglichen Meßanordnungen, um mit dem Georadar dreidimensionale Strukturen im Untergrund zu erfassen . . . . .	56
6.5	Konturplot eines in einer 3D-Georadarvermessung bestimmten Reflektors . . . . .	57
6.6	Darstellung einiger Radargramme aus einer 3D-Vermessung als sogenanntes Fence-Diagramm . . . . .	59
6.7	Anwendung einer Farbtabelle auf die Radargramme, um die Reflektoren hervorzuheben . . . . .	60
6.8	Darstellung zur Funktionsweise des Autotrackers bei der Interpretation von Radargrammen . . . . .	60
6.9	Darstellung aller Radargramme einer 3D-Vermessung . . . . .	61
6.10	Darstellung zur Abhängigkeit des Interpreters von der Perspektive bei der Bearbeitung von Radargrammen einer 3D-Vermessung . . . . .	63
6.11	Weitere Meßanordnungen, die mit Hilfe von Stereo-Visualisierung möglich sind, um mit dem Georadar 3D-Strukturen im Untergrund zu erfassen . . . . .	64
6.12	Histogramme für den elektrischen Widerstand im Untergrund des hydrogeologischen Versuchsfeldes Krauthausen . . . . .	69
6.13	Übersichtskarte mit den Bohrlokalationen für das hydrogeologische Versuchsfeld Krauthausen . . . . .	71
6.14	Darstellung des geologischen Profils und der natürlichen Strahlungsaktivität beim hydrogeologischen Versuchsfeld Krauthausen . . . . .	74
6.15	Darstellung von volumetrischen elektrischen Widerstandsmessungen mittels dreier Schnittebenen . . . . .	75
6.16	Darstellung von volumetrischen elektrischen Widerstandsmessungen als Voxelvisualisierung . . . . .	77
6.17	Visualisierung des Krauthausen Datensatzes an der Holobench . . . . .	79
6.18	Visualisierung des Krauthausen Datensatzes an der Holobench . . . . .	80
6.19	Herdlokation von Beben, welche innerhalb eines Zeitraumes von 1,5 Tagen beim geothermischen Reservoir Soultz-sous-Forêts gemessen wurden . . . . .	85
6.20	Das Modell des Schauinsland-Bergwerkes . . . . .	89
6.21	Markierter Bereich des Bergwerkes im Schauinsland . . . . .	91

7.1	Möglicher Arbeitsablauf für eine geophysikalische Untersuchung des Untergrundes . . . . .	95
7.2	SpaceBall und SpaceMouse der Firma 3DConnexion . . . . .	100



# Literaturverzeichnis

Advanced Visual Systems (2001): Homepage der Firma Advanced Visual Systems Inc., Waltham, Massachusetts, Im WWW: "<http://www.avs.com>"

Agrawala, M., Beers, A. C., McDowall, I., Fröhlich, B., Bolas, M., Hanrahan, P. (1997): The Two-User Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space, Proceedings of the 24th Conference on Computer Graphics & Interactive Techniques, Los Angeles, Association for Computing Machinery (ACM), New York, S. 327 - 332  
Im WWW (2001): "<http://graphics.stanford.edu/papers/twoviewer>"

ArcView (1996): Mit ArcView GIS Arbeiten, Environmental Systems Research Institute Inc., Redlands, California, Firmenwebsite: "<http://www.esri.com>"

Bad Urach (2001): Stadtwerke Bad Urach - Geothermische Energie,  
Im WWW: "[http://www.geothermie.de/bad\\_urach.htm](http://www.geothermie.de/bad_urach.htm)"

Baria, R., Baumgärtner, J., Gerard, A., Jung, R., Garnish, J. (1999): European HDR Research Programme at Soultz-sous-Forêts (France) 1987 - 1996, Geothermics, 28, Elsevier, S. 655 - 669

Barrass, S., Zehner, B. (2000): Responsive Sonification of Well-Logs, Proceedings of the Sixth International Conference on Auditory Display (ICAD 2000), April 2000, Atlanta, Georgia

Bethel, W., Jacobsen, J., Austin, A., Lederer, M., Little, T. (1996): Implementing Virtual Reality Interfaces for the Geosciences, LBL Report Number 38618, Lawrence Berkeley National Laboratory, Berkeley, California,  
Im WWW (2001): "<http://www-vis.lbl.gov/acti/paper>"

Blach, R., Landauer, J., Rösch, A., Simon, A. (1998): A Highly Flexible Virtual Reality System, Future Generation Computer Systems, Special Issue on Virtual

- Environments, Elsevier, Amsterdam,  
Im WWW (2001): "<http://vr.iao.fhg.de/papers/ve98pap.pdf>"
- Bolas, M., McDowall, I., Williams, G., Corr, D., Berta, J. (2000): Pixels are Good, In: Merritt, O., Benton, S. A., Woods, A. J., Bolas, M. T. (Editors), Stereoscopic Displays and Virtual Reality Systems VII, Proceedings of SPIE Vol. 3975, S. 389 - 395
- Bormann, S. (1994): Virtuelle Realität, Genese und Evaluation, Addison-Wesley (Deutschland) GmbH, Bonn, 288 S.
- Cohen, J. K. & Stockwell, Jr. J. W. (1999), CWP/SU: Seismic Unix Release 33: A Free Package for Seismic Research and Processing, Center for Wave Phenomena, Colorado School of Mines. Im WWW (2001): "<http://www.cwp.mines.edu>"
- Cruz-Neira, C., Sandin, D. J., DeFanti, T. A. (1993): Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE, Proceedings of the ACM SIGGRAPH '93 Conference, August 1993, Anaheim, California, Computer Graphics Proceedings, Annual Conference Series, 1993, Association for Computing Machinery (ACM), New York, S. 135 - 142
- Dai, F. (1997): Virtual Reality for Industrial Applications, Springer-Verlag, Berlin, Heidelberg, 173 S.
- Döring, U. (1995): Transport der reaktiven Stoffe Eosin, Uranin und Lithium in einem heterogenen Grundwasserleiter, Dissertation, Universität Kiel
- Eckel, G. (1999): Applications of the CyberStage Spatial Sound Server, In: von Recklinghausen, D. R. (Editor), Proceedings of the Audio Engineering Society 16th International Conference on Spatial Sound Reproduction, April 1999, Rovaniemi, Finland, Audio Engineering Society, Inc., New York, S. 478 - 484
- Englert, A. (1998): Räumliche Variabilität der hydraulischen Durchlässigkeit eines Quartären Aquifers (Testfeld Krauthausen), Diplomarbeit im Fach Geologie, Rheinisch-Westfälische Technische Hochschule Aachen, 118 S.
- Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F. (1990): Computer Graphics - Principles and Practice, Second Edition, Addison-Wesley Publishing Company, Inc., 1174 S.
- Fossen, H., Hesthammer, J. (1998): Structural Geology of the Gullfaks Field, Northern North Sea, In: Coward, M. P., Daltaban, T. S., Johnson, H., Structural

- Geology in Reservoir Characterization. Geological Society Special Publications 127, London, S. 231 - 261
- Freiburg (2001): VRengine: Silver Mine Visualization, Homepage des Geologischen Instituts der Universität Freiburg, Im WWW:  
“<http://www.geologie.uni-freiburg.de/root/vrml/schauinsland/silvermine.html>“
- Fröhlich, B., Barrass, S., Zehner, B., Plate, J., Göbel, M. (1999): Exploring Geoscience Data in Virtual Environments, In: David, E., Gross, M., Hamann, B. (Editors), Proceedings of the 10th IEEE Conference on Visualization (Vis99), October 1999, San Francisco, California, Association for Computing Machinery (ACM), New York, S. 169 - 173
- Fröhlich, B., Plate, J. (2000): The Cubic Mouse - A New Device for Three-Dimensional Input, In: Turner, T., Proceedings of the CHI2000 conference on Human factors in computing systems, March/April 2001, The Hague, Netherlands, ACM-Press, New York, S. 526 - 531
- Ganter, S. (1995): 3D-Modellierung der Grube Schauinsland, Südschwarzwald, Diplomarbeit im Fach Geologie, Albert-Ludwigs-Universität Freiburg, 80 S.
- Gardner, H. J., Boswell, R. W. (1998): Effective Virtual Environments - Experiences with a Low Cost Immersive System, In: Lam, K. Y., Choo, B. C., Kumar, K. (Editors), Proceedings of the HPC Asia Conference, Singapore, September 1998, Vol. 1, S. 658 - 663
- Geophysical Survey Systems Inc. (2001): Homepage der Firma Geophysical Survey Systems Inc., North Salem, New Hampshire, Im WWW:  
“<http://www.geophysical.com>“
- Geowall Konsortium (2001): The Geowall Consortium,  
Im WWW: “<http://www.evl.uic.edu/cavern/geowall/indexmain.html>“
- Gocad (2001): Homepage des Gocad Projektes an der Universität Nancy, Frankreich,  
Im WWW: “<http://www.ensg.u-nancy.fr/GOCAD/>“
- Goebbels, G. (2001): Team work in Distributed Collaborative Virtual Environments, Dissertation im Fach Informatik, Universität Pretoria, Südafrika, 189 S.
- Green, A., Lanz, E., Maurer, H., Börner, D. (1999): A template for geophysical investigations of small landfills. The Leading Edge, 18/2, Society of Exploration Geophysicists (SEG), Tulsa, S. 248 - 254

- Häfner, U., Simon, A., Doulis, M. (2000a): Unencumbered Interaction in Display Environments with Extended Working Volume, In: Stereoscopic Displays and Virtual Reality Systems VII, Merritt, O., Benton, S. A., Woods, A. J., Bolas, M. T. (Editors), Proceedings of SPIE Vol. 3957, S. 473 - 480
- Häfner, U., Bues, M., Magg, R. (2000b): Wireless Interaction in Cost-Effective Display Environments, In: Cruz-Neira, C. (Editor), Fourth International Immersive Projection Technology Workshop (IPT2000), June 2000, Ames, Iowa
- Harding, C., Loftin, B. (2000): Visualization and Modelling of Geoscientific Data on the Interactive Workbench, The Leading Edge, 19/5, Society of Exploration Geophysicists (SEG), Tulsa, S. 506 - 511
- Harding, C. (2001): Multi-Sensory Investigation of Geoscientific Data: Adding Touch and Sound to 3D Visualization and Modeling, Dissertation, Department of Geosciences, University of Houston, Texas, 158 S.,  
Im WWW (2002): "<http://users2.ev1.net/~harding77019> → Ph.D. thesis"
- Harvey, B., Wright, M. (1994): Simply Scheme - Introducing Computer Science, MIT Press, Cambridge, Massachusetts, 583 S.
- Hofmann, T. (2001): Virtual Reality Work Environment, Diplomarbeit im Fach Industrial Design, Universität GH Essen, Fachbereich 4, 157 S.
- Hybertsen, J. (1998): Statoil Testing Virtual Reality on Real Seismic Data, Offshore, May 1998, Pennwell Publishing, S. 92 - 95
- Jones, R. H., Beauce, A., Jupe, A., Fabriol, H., Dyer, B. C. (1995): Imaging Induced Microseismicity during the 1993 Injection Tests at Soultz-Sous-Forêts, France, In: Barbier, E. (Editor), Proceedings of World Geothermal Congress, May 1995, Vol. 4, Florence, Italy, International Geothermal Association, Auckland, New Zealand, S. 2665 - 2669
- Kidd, G. D. (1999): Fundamentals of 3-D Seismic Volume Visualization, The Leading Edge 18/6, Society of Exploration Geophysicists (SEG), Tulsa, S. 702 - 709
- Klein, H., Ramshorn, Ch. (1991): 3D-Computergraphik zum Visualisieren von geologischen Strukturen und simulierten geologischen Prozessen, Dissertation im Fach Geologie, Geowissenschaftliche Fakultät, Universität Freiburg, 183 S.
- Krüger, W., Bohn, C. A., Fröhlich, B., Schüth, H., Strauss, W., Wesche, G., (1995): The Responsive Workbench: A Virtual Work Environment, IEEE Computer, Vol. 28, July 1995, IEEE Computer Society, Los Alamitos, California, S. 42-48



- Lindenbeck, Ch., Ulmer, H. (1995): Entwicklung und Anwendung von Computerprogrammen zur Visualisierung geologischer Strukturen und Prozesse, Dissertation im Fach Geologie, Freiburger Geowissenschaftliche Beiträge, Freiburg im Brsg., Band 9, 280 S.
- Lohmann, G. (1998): Volumetric Image Analysis: An Overview, John Wiley & Sons, Inc, New York, Weinheim u.a., 250 S.
- McLaughlin, J. P., Orenstein, B. J. (1997): Haptic Rendering of 3D Seismic Data, In: Salisbury, J. K., Srinivasan, M. A. (Editors), Proceedings of the Second Phantom Users Group Workshop, October 1997, Dedham/Cambridge, Massachusetts, Im WWW:  
“<http://www.cs.sandia.gov/SEL/conference/pug00/proceedings.htm>“
- Midttun, M., Helland, R., Finnstrom, E. (2000): Virtual Reality - Adding Value to Exploration and Production, The Leading Edge, 19/5, Society of Exploration Geophysicists (SEG), Tulsa, S. 538 - 544
- Münster (2001): Homepage des Institutes für Geoinformatik an der Universität Münster, Im WWW: “[www.ifgi.uni-muenster.de](http://www.ifgi.uni-muenster.de)“
- Nesbitt, K. V., Orenstein, B. J., Gallimore, R. J., McLaughlin, J. P. (1997): The Haptic Workbench Applied to Petroleum 3D Seismic Interpretation, In: Salisbury, J. K., Srinivasan, M. A. (Editors), Proceedings of the Second Phantom Users Group Workshop, October 1997, Dedham/Cambridge Massachusetts, Im WWW:  
“<http://www.cs.sandia.gov/SEL/conference/pug00/proceedings.htm>“
- Opriessnig, G. (1999): Visualization in Tunnelling - a Virtual Reality Environment for Tunnelling Engineers, Felsbau 17/1, Verlag Glückauf, Essen, S. 10 - 15
- Phillips, W. S. (2000): Precise Microearthquake Locations and Fluid Flow in the Geothermal Reservoir at Soultz-sous-Forêts, France, Bulletin of the Seismological Society of America, 90/1, El Cerrito, California, S. 212 - 228
- Pongratz, E. (1999): Geologische Strukturmodellierung des Tagebaus Bergheim, Diplomarbeit im Fach Geologie, Friedrich-Wilhelms-Universität Bonn, 85 S.
- Reachin (2001): Homepage der Firma Reachin Technologies AB, Stockholm, Schweden, Im WWW: “<http://www.reachin.se>“

- Real Time Graphics (2000): Our Annual Survey of Head-Mounted Displays, Real Time Graphics, 9/2, Computer Graphics Systems Development Corp., Mountain View, California, S. 1, S. 6 - 12
- Real Time Graphics (2001a): Trimension Systems Building Visualization Centers for Shell, Real Time Graphics, 9/10, Computer Graphics Systems Development Corp., Mountain View, California, S. 1 - 2
- Real Time Graphics (2001b): Mechdyne Installs VR Systems in Taiwan, Nigeria and Venezuela, Real Time Graphics, 10/1, Computer Graphics Systems Development Corp., Mountain View, California, S. 11
- Reik, G., Zemke, C., Kesper, B., Möller, D. P. F. (1999): Dynamische Virtual Reality Modelle in Geotechnischen Anwendungen, In: Schmidt, B., Uhlenkücken, C. (Hrsg.), Visualisierung raumbezogener Daten: Methoden und Anwendungen, IFGIprints 6, Verlag Natur & Wissenschaft, Solingen, S. 35 - 43  
Weiter Informationen im WWW (2001): "www.bagis.de"
- van Reimersdahl, T., Kuhlen, T., Gerndt, A., Henrichs, J., Bischof, C. (2000): ViSTA: a Multimodal, Platform Independent VR-Toolkit based on WTK, VTK, and MPI, In: Cruz-Neira, C. (Editor), Fourth International Immersive Projection Technology Workshop (IPT2000), June 2000, Ames, Iowa
- Rohlf, J., Helman, J. (1994): IRIS performer: A High Performance Multiprocessing Toolkit for Real-Time 3-D Graphics., Proceedings of SIGGRAPH 94 Conference, July 1994, Orlando, Florida, Computer Graphics Proceedings, Annual Conference Series, 1994, Association for Computing Machinery (ACM), New York, S. 381 - 395
- Sambuelli, L., Socco, L. V., Brecciaroli, L. (1999): Acquisition and Processing of Electric, Magnetic and GPR Data on a Roman Site, Journal of Applied Geophysics, 41, Elsevier, Amsterdam, S. 189 - 204
- Sandmeier Scientific Software (2001): Firmenhomepage von Sandmeier Scientific Software, Karlsruhe, Germany,  
Im WWW: "http://www.ka.shuttle.de/software"
- Schneider, M. (1995): Charakterisierung der Variabilität der Permeabilität in einem quartären Grundwasserleiter, Diplomarbeit im Fach Geologie, Rheinisch-Westfälische Technische Hochschule Aachen, 74 S.

- Schroeder, W., Martin, K., Lorenzen, B. (1996): *The Visualization Toolkit, An Object Oriented Approach to 3D Graphics*, Prentice Hall PTR, New Jersey, 826 S.
- Schulze, K. C. (1996): *Einsatz elektrischer Meßverfahren zur Erkundung eines hydrologischen Testfeldes bei Jülich*, Diplomarbeit im Fach Physik, Friedrich-Wilhelms-Universität Bonn, 100 S.
- Scientific Software Group (2001), *Homepage der Firma Scientific Software Group*, Washington, DC, Im WWW: "www.scisoftware.com"
- Seichter, M. (1998): *Rekonstruktion der räumlichen Verteilungen komplexer elektrischer Leitfähigkeiten*, Dissertation, Technische Universität Braunschweig, Verlag Papierflieger, Clausthal-Zellerfeld, 140 S.
- Selley, R.C. (1998): *Elements of Petroleum Geology*, Academic Press, San Diego, California, 470 S.
- Sensable Technologies (2001): *Homepage der Firma Sensable Technologie*, Woburn, Massachusetts, Im WWW: "www.sensable.com"
- Seuffert, M. B. (1997): *Statistische Analyse und Interpolation von Gamma-Log-Messungen im Rahmen einer hydrogeologischen Fragestellung*, Diplomarbeit im Fach Physik, Friedrich-Wilhelms-Universität Bonn, 101 S.
- Sheffield, T. M., Meyer, D., Lees, J., Kahle, G., Payne, B. (1999): *Geovolume Visualization Interpretation: Color in 3D Volumes*, The Leading Edge, 18/6, Society of Exploration Geophysicists (SEG), Tulsa, S. 668 - 674
- Southard, D.A. (1992): *Transformations for Stereoscopic Visual Simulation*, Computers & Graphics, 16/4, Pergamon, Elsevier Science Ltd., Oxford, S. 401 - 410
- Springer, J., Tramberend, H., Fröhlich, B. (2000): *On Scripting in Distributed Virtual Environments*, In: Cruz-Neira, C. (Editor), *Proceedings of the Fourth International Immersive Projection Technology Workshop (IPT2000)*, June 2000, Ames, Iowa
- Stark, T. J., Dorn, G. A., Cole, M. J. (2000a): *ARCO and Immersive Environments, Part 1: The First Two Generations*, The Leading Edge, 19/5, Society of Exploration Geophysicists (SEG), Tulsa, S. 526 - 532

- Stark, T. J., Dorn, G. A., Cole, M. J. (2000b): ARCO and Immersive Environments, Part 2: Oil Industry Experience with Immersive Environments, The Leading Edge, 19/8, Society of Exploration Geophysicists (SEG), Tulsa, S. 884 - 890
- Stevenson, D. R., Bogsanyi, F. J. D. (2000): Applying Hapto-Visual Virtual Environment Systems to Industrial Applikations, In: Merrit, J. O., Benton, S. A., Woods, A. J., Bolas, M. T. (Editors), Stereoscopic Displays and Virtual Reality Systems VII, Proceedings of SPIE Vol. 3957 (2000), S. 446 - 453
- Tramberend, H. (1999): Avocado: A Distributed Virtual Reality Framework, Virtual Reality 99 Conference, March 1999, Houston, Texas, IEEE Computer Society, Los Alamitos, California, S. 14 - 21
- Tramberend, H., Hasenbrink, F., Fröhlich, B. (1999): Tools, Mediators, and Interaction Operators: A Concept for Interaction in Virtual Environments, In: Bullinger, H. J., Riedel, O. (Editors), Proceedings of the International Immersive Technology Workshop, Stuttgart, May 1999, Springer-Verlag, S. 75 - 77
- Watsen, K., Darken, R. P., Capps, M. V. (1999): A Handheld Computer as an Interaction Device to a Virtual Environment, In: Bullinger, H. J., Riedel, O. (Editors), Proceedings of the International Immersive Projection Technology Workshop, May 1999, Stuttgart, Springer Verlag, S. 51 - 57
- Wernecke, J. (1994): The Inventor Mentor - Programming Object Oriented 3D Graphics with Open Inventor Release 2, Addison Wesley Publishing Company, 514 S.
- Wesche, G., Droske, M. (2000): Conceptual Free-Form Styling on the Responsive Workbench, In: Kwang Yun Wohn (Editor), Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 2000), October 2000, Seoul, Korea, Association for Computing Machinery (ACM), New York, S. 83 - 91
- Whitehouse, D. (1999): Visualization at the Australian National University, In: Vis-Files, Hibbard, B. (Columnist), Computer Graphics, 33/4, Association for Computing Machinery (ACM), New York, S. 9 - 11
- Williams, B. K. (1994): Visualization of Reservoir Simulation Data Using an Immersive Virtual Reality System, Proceedings of the European Petroleum Computer Conference in Aberdeen, March 1994, Society of Petroleum Engineers, S. 77-83

- Wright, D. (2000): A Survey of Projection-Based Immersive Displays, In: Stereoscopic Displays and Virtual Reality Systems VII, Merritt, O., Benton, S. A., Woods, A. J., Bolas, M. T. (Editors), Proceedings of SPIE, Vol. 3957, S. 482 - 492
- Zehner, B., Barrass, S., Plate, J., Göbel, M., Fröhlich, B. (1999): Interaktive Visualisierung und Validierung von Geophysikalischen Daten in Virtuellen Umgebungen, In: Schmidt, B., Uhlenkücken, C. (Hrsg.), Visualisierung raumbezogener Daten: Methoden und Anwendungen, IFGIprints 6, Verlag Natur & Wissenschaft, Solingen, S. 59 - 72
- Zehner, B., Kümpel, H.-J. (2001): Immersive Visualisierungstechniken bei der Geophysikalischen Erkundung des Untergrundes, Kartographische Nachrichten, 51/4 (2001), Kirschbaum Verlag, Bonn, S. 196 - 200
- 3DConnexion, 2001: Merger der Firmen LogiCad3D und Labtec 3D, <http://www.3DConnexion.com>



## Liste der Abkürzungen

CRT	Cathode Ray Tube
CMP	Common Midpoint
CPS	Counts Per Second
DLP	Digital Light Processing
FOV	Field Of View
GMD	Gesellschaft für Mathematik und Datenverarbeitung
GPR	Ground Penetrating Radar
HMD	Head Mounted Display
$K_f$	Hydraulische Permeabilität
SGI	Silicon Graphics Inc.
VE	Virtuelle Umgebung (Virtual Environment)
VR	Virtuelle Realität (Virtual Reality)
VRML	Virtual Reality Modeling Language
6DOF	6 Degrees Of Freedom





# Anhang Implementierung

Um einen Einblick in die Implementierung der Prototypen zu vermitteln, soll hier als Beispiel der Source-Code dokumentiert werden, welcher der Klasse zur Visualisierung der geoelektrischen Daten vom hydrogeologischen Versuchsfeld Krauthausen zugrunde liegt (siehe Abschnitt 6.3). Hierbei wird davon ausgegangen, daß der Leser sich mit den Konzepten der objektorientierten Programmierung und der Graphikbibliothek IRIS Performer oder einer anderen, das Konzept des Szenengraphen verwendenden, Graphikbibliothek auskennt. Da AVANGO sich in ständiger Fortentwicklung befand, wurde versucht die relativ aufwendige Implementierung der Visualisierung weitgehend unabhängig von dieser Rahmenapplikation zu halten. Daher beruht die Programmierung auf einer Zwei-Schichten Struktur.

Die Klasse *fdDataVol* implementiert die Datenverwaltung und Visualisierung der Daten. Über den Aufruf ihrer Elementfunktionen (member functions) läßt sich die Visualisierung steuern. Eine Instanz dieser Klasse ist eingebettet in eine Instanz der Klasse *fpFDDDataVol*, welche von der AVANGO-Klasse (*fpDCS*) abgeleitet ist und dazu dient, die Visualisierung und Funktionalität der *fdDataVol*-Klasse in das AVANGO System einzubinden. Werden Felder der *fpFDDDataVol*-Klasse geändert (meist vom Scheme-Interface aus), so werden die entsprechenden Elementfunktionen der Klasse *fdDataVol* aufgerufen und auf diese Weise die Visualisierung wie gewünscht geändert. Um diesen Mechanismus zu dokumentieren, sind nachfolgend eine Reihe von Source-Dateien angehängt:

- Die Header-Datei der Klasse *fdDataVol*, um zu zeigen welche Funktionen für diese Klasse zur Darstellung der Daten implementiert wurden.
- Die Header-Datei der Klasse *fpFDDDataVol*, um zu zeigen welche Felder und Funktionen für diese Klasse zur Steuerung der Visualisierung implementiert wurden.

- Die Implementierung der Klasse *fpFDDataVol* mit Feldern und Funktionen, um zu dokumentieren, wie das Setzen der Felder in eine Veränderung der Visualisierung umgesetzt wird. Die beiden zentralen Funktionen sind *fieldHasChanged(fpField &field)* und *evaluate()*. Wird vom Scheme-Interface aus ein Feld geändert, so wird jedesmal die Funktion *fieldHasChanged(fpField &field)* aufgerufen und die Adresse des entsprechenden Feldes übergeben. Auf diese Weise kann die Klasse registrieren welche Felder geändert wurden. Jedesmal, wenn das Bild neu berechnet wird, wird die Funktion *evaluate()* aufgerufen, welche dann, in Abhängigkeit davon welche Felder neu gesetzt wurden, die entsprechenden Funktionen der Klasse *fdDataVol* aufruft, um die Visualisierung anzupassen.
- Ein Beispiel für die Steuerung der Visualisierung mit Hilfe der Klasse *fpFDDataVol* vom Scheme-Interface aus soll zeigen, wie diese Klasse instantiiert und wie Felder gesetzt werden.
- Implementierung der Klasse *fdDataVol* mit Hilfe der IRIS Performer Bibliothek.

Insbesondere bei der Klasse *fdDataVol* sind sich einige Funktionen sehr ähnlich, da sie dieselbe Funktionalität für unterschiedliche Raumrichtungen implementieren. Hier wurde jeweils nur die Implementierung für eine Raumrichtung angegeben. Wie in Abschnitt 3.1 beschrieben, werden die Application- Cull- und Draw-Schritte bei der Visualisierung auf mehrere Prozesse aufgeteilt. Hierbei muß verhindert werden, daß mehrere Prozesse gleichzeitig auf dieselben Daten zugreifen, da sonst zum Beispiel der Application-Prozeß die Koordinaten der Schnittebenen verändern kann, während der Draw-Prozeß sie gerade zeichnet. Aus diesem Grunde werden die dynamischen Daten von der IRIS Performer Klasse *pfFlux* verwaltet. Diese hat mehrere Kopien der Daten, die durch eine Bildnummer gekennzeichnet sind. Auf diese Weise können die verschiedenen Prozesse auf eine Kopie der Daten zugreifen, welche (zeitlich) dem Bild entspricht, an dem sie gerade arbeiten.

```

// Headerfile fuer die Klasse zur Visualisierung von Skalaren Daten auf einem dreidimensionalen zellen-
// basierten Finite-Differenzen Gitter.
//
// 1.) Anzeige der Werte ueber Farbkodierung auf drei senkrecht zueinander stehenden Schnitten, welche
// zwischen den Koordinatenachsen aufgespannt werden
//
// 2.) Anzeige von verschiedenen Wertebereichen als Voxelvisualisierung
//
// Die Einbindung in die AVANGO Rahmenapplikation geschieht ueber den Knoten "fpFDDDataVol"
//
// Da verschiedene Prozesse (App, Cull, Draw) auf die geometrischen Daten zugreifen, muessen diese
// gepuffert werden (pfFluxedBuffer).
//
// -----Bjoern Zehner, Bonn, 2000-----

```

10

```

#ifndef FINITE_DIFFERENCES_DATA_VOL_H
#define FINITE_DIFFERENCES_DATA_VOL_H

```

```

#include <Performer/pf.h>
#include <Performer/pf/pfGeode.h>
#include <Performer/pf/pfDCS.h>
#include <Performer/pr/pfGeoSet.h>
#include <Performer/pr/pfGeoState.h>
#include <Performer/pr/pfMaterial.h>
#include <Performer/pr/pfTexture.h>

```

20

```

#include <iostream.h>
#include <fstream.h>
#include <string.h>

```

30

```

#define VAL_INVALID_VALUES -999.25

```

```

static int makeFluxedGSet(pfFluxMemory *fmem);

```

```

class fdDataVol
{
private:
    // Beschreibung des Datenvolumens
    int _numXBorders; //
    int _numYBorders; // Anzahl der Zellgrenzen in den drei Raumrichtungen
    int _numZBorders; //

    float *_xBorderPos; //
    float *_yBorderPos; // Positionen der Zellgrenzen in den drei Raumrichtungen
    float *_zBorderPos; //

    int _numXCells; //
    int _numYCells; // Anzahl der Zellen (jeweils _num_Borders - 1)
    int _numZCells; //

    float ***_values; // Die darzustellenden Daten

    float _valMin; // Minimum, Maximum und Wertebereich aller Daten,
    float _valMax; // wird z.B. verwendet um die Farbtabelle an den
    float _valRange; // Wertebereich anzupassen

```

40

```

    // Geometrie der drei Schnittebenen
    pfDCS *_dcs;
    pfGeode *_xyGeode;
    pfGeode *_xzGeode;
    pfGeode *_yzGeode;

```

60

```

    uint _syncGroup;

    pfTexture *_texture;
    pfTexEnv *_texenv;

    pfFlux *_xyFluxedGSet; //
    pfFlux *_xyFluxedCoords; // Geometrie der Ebene zwischen x und y Achse
    pfFlux *_xyFluxedTexCoords; //

    pfFlux *_xzFluxedGSet; //
    pfFlux *_xzFluxedCoords; // Geometrie der Ebene zwischen x und z Achse
    pfFlux *_xzFluxedTexCoords; //

```

70

```

pfFlux *_yzFluxedGSet;           //
pfFlux *_yzFluxedCoords;        // Geometrie der Ebene zwischen y und z Achse
pfFlux *_yzFluxedTexCoords;     //
                                                                    80

pfFlux *_fluxedVoxGSet;         //
pfFlux *_voxFluxedCoords;       //
pfFlux *_voxFluxedTexCoords;    // Beschreibung der Geometrie fuer die Visualisierung
pfFlux *_voxFluxedNorms;       // als Voxel
pfFlux *_voxFluxedBBox;        //
pfGeoide *_voxGeoide;           //
int _voxMaxNumberOfVertices;    //
int _voxNumberOfVertices;       //
                                                                    90

bool _ifVoxelVizIsShown;        //
bool _ifSectionsAreShown;       //

void initVoxelViz();
enum {CONSTX_LOW_BORDER, CONSTX_HIGH_BORDER, CONSTY_LOW_BORDER, CONSTY_HIGH_BORDER,
      CONSTZ_LOW_BORDER, CONSTZ_HIGH_BORDER};

void voxAddCellBorder(int xIndex, int yIndex, int zIndex, int border, pfVec3 *coords,
                      pfVec2 *texCoords, pfVec3 *norms, float cTablePos);
                                                                    100

// Funktionen die von ausserhalb der Klasse aufgerufen werden koennen
public:
fdDataVol(int numXBorders, int numYBorders, int numZBorders, float *xBorderPos, float *yBorderPos,
           float *zBorderPos);

fdDataVol(const char *filename);
~fdDataVol();
                                                                    110

void evaluateValuesMinMax();
void writeGridInformation();
void writeInformation();

pfDCS *getDCS();

void initTexture();

void calculationsFinished();
                                                                    120

void initXYSection(float initialZ); // Initialisieren der drei Schnittebenen und setzen
void initXZSection(float initialY); // an eine Anfangsposition
void initYZSection(float initialX); //

void showSectionViz(); // An und ausschalten der Visualisierung mittels
void removeSectionViz(); // Schnittebenen

void setXYSection(float zPos); // Setzen der Schnittebenen entlang der zu ihnen
void setXZSection(float yPos); // senkrechten Achse
void setYZSection(float xPos); //
                                                                    130

void setXYSectionRel(float zRelPos); // Setzen der Schnittebenen relativ zur Ausdehnung des
void setXZSectionRel(float yRelPos); // Datensatzes. Ein Uebergabewert von 0.5 setzt sie
void setYZSectionRel(float xRelPos); // z.B. in die Mitte des Datensatzes

void setMinValue(float min); // Setzen der Extremwerte die auf die Enden der Farb-
void setMaxValue(float max); // tabelle abgebildet werden

float getXPositionFromRelativeValue(float relVal);
float getYPositionFromRelativeValue(float relVal);
float getZPositionFromRelativeValue(float relVal);
                                                                    140

int getIndexForXPos(float xPos); // Bestimmung welchen Index die Zelle hat in der die
int getIndexForYPos(float yPos); // angegebene Position liegt (Entlang x y z)
int getIndexForZPos(float zPos); //

// Umsetzung der Float-RGBA-Werte in 32 bit Farbkodierung
uint getTexel(float r, float g, float b, float alpha)

void loadColormapFromFile(const char *filename);
                                                                    150

bool isXYGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relY);
bool isXZGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relZ);
bool isYZGSet(pfVec3 hitPos, pfGeoSet *gset, float &relY, float &relZ);

```

```
// Berechnung der Voxelvisualisierung
void voxSetBorders(float lowerBorder, float upperBorder);
void voxSetTransparency(float alpha);

void showVoxelViz(); // An und ausschalten der Voxelvisualisierung
void removeVoxelViz(); //
};

#endif
```



```

// Headerfile fuer die Klasse fpFDDataVol. Dieses ist von der AVANGO Klasse fpDCS abgeleitet und stellt
// die Verbindung der fdDataVol Klasse in die Rahmenapplikation AVANGO dar. Die Aenderung der Felder
// wird ausgewertet und die entsprechenden Funktionen der Klasse fdDataVol aufgerufen, so das man die
// Visualisierung der Daten ueber den Feldmechanismus von AVANGO steuern kann
//
//-----Bjoern Zehner, 2001-----

#ifndef FP_FD_DATA_VOL_H
#define FP_FD_DATA_VOL_H
// 10

#include <Performer/pf.h>
#include <libfp/performer/fpDCS.h>

#include "fdDataVol.h"

class fpFDDataVol : public fpDCS
{
    FP_FC_DECLARE();
// 20

private:
    bool    _ifNewDataFilename;    // Variablen halten fest, ob das jeweilige Feld
    bool    _ifNewColormapFilename; // gerade geaendert worden ist, oder nicht
    bool    _ifNewRelXSlicePos;    //
    bool    _ifNewRelYSlicePos;    //
    bool    _ifNewRelZSlicePos;    //
    bool    _ifNewShowSections;    //
    bool    _ifNewValuesMin;       //
    bool    _ifNewValuesMax;       //
    bool    _ifNewVoxBorderMin;    //
    bool    _ifNewVoxBorderMax;    //
    bool    _ifNewVoxTransparency; //
    bool    _ifNewShowVoxels;     //
// 30

public:
    //
    // Die Felder die vom Scheme Interface aus sichtbar sind
    //
    fpSFString DataFilename;       // Name der Datei, die die Daten enthaelt
// 40

    fpSFString ColormapFilename;   // Name der Datei, die die Farbkodierung enthaelt

    fpSFFloat RelXSlicePos;        // Relative Position der Schnitte senkrecht zu
    fpSFFloat RelYSlicePos;        // den jeweiligen Achsen. RelYSlicePos=0.5 bedeutet,
    fpSFFloat RelZSlicePos;        // dass der Schnitt entlang der x-y Ebene in der
    // Mitte des Models, bezueglich seiner Ausdehnung
    // entlang der y Achse, steht

    fpSFBool ShowSections;         // An und Ausschalten der Visualisierung mittels
    // Schnittebenen (true, false)
// 50

    fpSFFloat ValuesMin;           // Setzen der Datenwerte, welche auf den Rand der
    fpSFFloat ValuesMax;           // Farbtabelle abgebildet werden. Allen Werten da-
    // runter bzw. darueber werden jeweils die Farben
    // vom Rand der Farbtabelle zugewiesen.

    fpSFFloat VoxBorderMin;        // Festlegung des Wertebereichs, fuer den die
    fpSFFloat VoxBorderMax;        // Zellen bei der Voxelvisualisierung angezeigt
    // werden
// 60

    fpSFBool ShowVoxels;          // An und Ausschalten der Visualisierung als
    // Voxel (true, false)

    fdDataVol *_datavol;          // Klasse / Object mit der Visualisierung

    // -----Funktionen-----
    fpFDDataVol();
    ~fpFDDataVol();
// 70

    // Die Funktion wird fuer jedes AVANGO-Feld das geaendert wird mit der Adresse des Feldes als Uebergabe-
    // parameter aufgerufen.
    void fieldHasChanged(fpField &field);

    // Die Funktion wird jeweils aufgerufen, wenn das Bild neu berechnet wird. Hier liegt die Funktionalitaet,
    // da in Abhaengigkeit, welche Felder geaendert wurden, der Zustand des visualisierten Objektes neu

```

```
// bestimmt wird.  
void evaluate(); 80  
  
// Die Funktion wird aufgerufen, wenn eine Instanz der fpFDDataVol Klasse mit einem Werkzeug ausgewählt  
// wurde und leitet die notwendigen Reaktionen ein.  
void reactOnHit(pfVec3 position, pfGeoSet *hitGSet);  
  
/* virtual */ pfMemory * castToPerformer();  
  
static pfType *classType;  
  
static Elk_Object getInformationCB(fpTyped* obj, Elk_Object arg); 90  
};  
#endif
```



```

//-----fpFDDataVol.c++-----
#include "fpFDDataVol.h"

FP_FC_DEFINE(fpFDDataVol)

pfType* fpFDDataVol::classType = NULL;

fpFDDataVol::fpFDDataVol() :
    _ifNewDataFilename(false),
    _ifNewColormapFilename(false),
    _ifNewRelXSlicePos(false),
    _ifNewRelYSlicePos(false),
    _ifNewRelZSlicePos(false),
    _ifNewShowSections(false),
    _ifNewValuesMin(false),
    _ifNewValuesMax(false),
    _ifNewVoxBorderMin(false),
    _ifNewVoxBorderMax(false),
    _ifNewVoxTransparency(false),
    _ifNewShowVoxels(false),
    _datavol(NULL)
{
    setType(classType);
    FP_FC_START_CONSTRUCT(fpDCS);

    FP_FC_ADD_FIELD(DataFilename, "");
    FP_FC_ADD_FIELD(ColormapFilename, "");
    FP_FC_ADD_FIELD(RelXSlicePos, 0.0);
    FP_FC_ADD_FIELD(RelYSlicePos, 0.0);
    FP_FC_ADD_FIELD(RelZSlicePos, 0.0);
    FP_FC_ADD_FIELD(ShowSections, 1);
    FP_FC_ADD_FIELD(ValuesMin, -999.95);
    FP_FC_ADD_FIELD(ValuesMax, 999.95);
    FP_FC_ADD_FIELD(VoxBorderMin, 0.0);
    FP_FC_ADD_FIELD(VoxBorderMax, 0.0);
    FP_FC_ADD_FIELD(VoxTransparency, 0.5);
    FP_FC_ADD_FIELD(ShowVoxels, 1);

    FP_FC_FINISH_CONSTRUCT();
}

fpFDDataVol::~fpFDDataVol()
{
}

void fpFDDataVol::initClass()
{
    if (classType == NULL) {
        //initialize Performer rtti
        fpDCS::initClass();
        classType = new pfType(fpDCS::getClassType(), "fpFDDataVol");

        //initialize AVANGO rtti
        FP_FC_INIT(fpDCS, fpFDDataVol, TRUE, classType);

        getClassTypeId().registerMemberFunction("getInformation", getInformationCB);
    }
}

/* virtual */
pfMemory* fpFDDataVol::castToPerformer()
{
    return (pfMemory*) this;
}

//
// Die Funktion wird jedesmal aufgerufen, wenn ein AVANGO Feld geaendert wird (z.B. vom Scheme
// Interface aus).
/////////
void fpFDDataVol::fieldHasChanged(fpField &field)
{
    if (&field == &DataFilename) {
        _ifNewDataFilename = true;
    }
}

```

```

} else if (&field == &ColormapFilename) {
    _ifNewColormapFilename = true;
} else if (&field == &RelXSlicePos) {
    _ifNewRelXSlicePos = true;
} else if (&field == &RelYSlicePos) {
    _ifNewRelYSlicePos = true;
} else if (&field == &RelZSlicePos) {
    _ifNewRelZSlicePos = true;
} else if (&field == &ShowSections) {
    _ifNewShowSections = true;
} else if (&field == &ValuesMin) {
    _ifNewValuesMin = true;
} else if (&field == &ValuesMax) {
    _ifNewValuesMax = true;
} else if (&field == &VoxBorderMin) {
    _ifNewVoxBorderMin = true;
} else if (&field == &VoxBorderMax) {
    _ifNewVoxBorderMax = true;
} else if (&field == &VoxTransparency) {
    _ifNewVoxTransparency = true;
} else if (&field == &ShowVoxels) {
    _ifNewShowVoxels = true;
}
}
}

//
// Die Funktion wird jedesmal aufgerufen, wenn das Bild neu berechnet wird (fuer jedes Frame).
// In Abhaengigkeit davon welche AVANGO Felder seit ihrem letzten Aufruf geaendert worden sind,
// werden die notwendigen Operationen ausgefuehrt.
//////////
void fpFDDataVol::evaluate()
{
    if (!_ifNewDataFilename) {
        if (_datavol != NULL) {
            removeChild(_datavol->getDCS());
            delete _datavol;
            _datavol = NULL;
        }
        _datavol = new fdDataVol(DataFilename.getValue().c_str());
        float xpos = _datavol->getXPositionFromRelativeValue(0.5);
        float ypos = _datavol->getYPositionFromRelativeValue(0.5);
        float zpos = _datavol->getZPositionFromRelativeValue(0.5);

        _datavol->initXYSection(zpos);

        _datavol->initXZSection(ypos);

        _datavol->initYZSection(xpos);

        pfDCS *newdcs = _datavol->getDCS();
        if (newdcs == NULL) {
            cout << "new dcs is NULL" << endl;
        }
        if (strcmp("", ColormapFilename.getValue().c_str()) != 0) {
            cout << "loading new colormap from file " << ColormapFilename.getValue().c_str() << endl;
            _datavol->loadColormapFromFile(ColormapFilename.getValue().c_str());
        }
        addChild(newdcs);
        _ifNewDataFilename = false;
    }
    if (!_ifNewColormapFilename && (_datavol != NULL)) {
        _datavol->loadColormapFromFile(ColormapFilename.getValue().c_str());
        _ifNewColormapFilename = false;
    }
    if (!_ifNewRelXSlicePos) {
        float xpos = _datavol->getXPositionFromRelativeValue(RelXSlicePos.getValue());
        _datavol->setYZSection(xpos);
        _ifNewRelXSlicePos = false;
        _datavol->calculationsFinished();
    }
    if (!_ifNewRelYSlicePos) {
        float ypos = _datavol->getYPositionFromRelativeValue(RelYSlicePos.getValue());
        _datavol->setXZSection(ypos);
        _ifNewRelYSlicePos = false;
        _datavol->calculationsFinished();
    }
}

```

```

if (!_ifNewRelZSlicePos) {
    float zpos = _datavol->getZPositionFromRelativeValue(RelZSlicePos.getValue());
    _datavol->setXYSection(zpos);
    _ifNewRelZSlicePos = false;
    _datavol->calculationsFinished();
}
160
if (!_ifNewShowSections) {
    if (_datavol != NULL) {
        if (ShowSections.getValue()) {
            _datavol->showSectionViz();
        } else {
            _datavol->removeSectionViz();
        }
    }
}
170
if (!_ifNewValuesMin || !_ifNewValuesMax) {
    if (!_ifNewValuesMin) {
        float min = ValuesMin.getValue();
        float diff = min + 999.95;
        if (diff < 0.0) diff *= (-1.0);
        if (diff > 0.1) {
            _datavol->setMinValue(min);
        }
        _ifNewValuesMin = false;
    }
    if (!_ifNewValuesMax) {
        float max = ValuesMax.getValue();
        float diff = max + 999.95;
        if (diff < 0.0) diff *= (-1.0);
        if (diff > 0.1) {
            _datavol->setMaxValue(ValuesMax.getValue());
        }
        _ifNewValuesMax = false;
    }
    RelXSlicePos.setValue(RelXSlicePos.getValue()); //
    RelYSlicePos.setValue(RelYSlicePos.getValue()); // Triggert das Neuzeichnen der Geometrie
    RelZSlicePos.setValue(RelZSlicePos.getValue()); //
}
180
if (!_ifNewVoxBorderMin) {
    if (_datavol != NULL) {
        _datavol->voxSetBorders(VoxBorderMin.getValue(), VoxBorderMax.getValue());
        _ifNewVoxBorderMin = false;
    } else {
        cout << "_datavol->voxSetBorders() was called, but no data were loaded so far" << endl;
    }
}
200
if (!_ifNewVoxBorderMax) {
    if (_datavol != NULL) {
        _datavol->voxSetBorders(VoxBorderMin.getValue(), VoxBorderMax.getValue());
        _ifNewVoxBorderMax = false;
    } else {
        cout << "_datavol->voxSetBorders() was called, but no data were loaded so far" << endl;
    }
}
if (!_ifNewVoxTransparency) {
    if (_datavol != NULL) {
        _datavol->voxSetTransparency(VoxTransparency.getValue());
        _ifNewVoxTransparency = false;
    }
}
210
if (!_ifNewShowVoxels) {
    if (_datavol != NULL) {
        if (ShowVoxels.getValue()) {
            _datavol->showVoxelViz();
        } else {
            _datavol->removeVoxelViz();
        }
        _ifNewShowVoxels = false;
    }
}
220
}
}

void fpFDDataVol::reactOnHit(pfVec3 position, pfGeoSet *hitGSet)
{
    float relX, relY, relZ;
    if (_datavol->isXYGSet(position, hitGSet, relX, relY)) {
}
230

```

```
    RelXSlicePos.setValue(relX);
    RelYSlicePos.setValue(relY);
} else if (_datavol->isXZGSet(position, hitGSet, relX, relZ)) {
    RelXSlicePos.setValue(relX);
    RelZSlicePos.setValue(relZ);
} else if (_datavol->isYZGSet(position, hitGSet, relY, relZ)) {
    RelYSlicePos.setValue(relY);
    RelZSlicePos.setValue(relZ);
}
}
```

240

```
Elk_Object fpFDDataVol::getInformationCB(fpTyped* obj, Elk_Object arg1)
{
    fpFDDataVol *self = (fpFDDataVol*) obj;

    if (self->_datavol != NULL) {
        self->_datavol->writeInformation();
    }

    return fp_scheme_bundle(true);
}
```

250

```

;;;;;;
;; Scheme Programm welches die Verwendung der fpFDDataVol Klasse verdeutlicht
;;
;;-----Bjoern Zehner 2001-----
;;

;; Instantiieren der Klasse fpFDDataVol. Bei setzen des Feldes DataFilename werden die Daten
;; aus der Datei geladen. Bei setzen des Feldes ColormapFilename wird die gewuenschte Farb-
;; tabelle aus einer Datei geladen
(define el-volume (make-instance-by-name "fpFDDataVol"))
(fp-set-value el-volume 'Name "el-volume")
(fp-set-value el-volume 'DateFilename "./el-data/wenner-data.dat")
(fp-set-value el-volume 'ColormapFilename "./colormaps/colormap-8.rgb")

;; Erzeugen eines Objekts vom Type fpDCS. Das Objekt el-volume wird im Szenengraphen unter
;; dieses gehaengt
(define el-volume-dcs (make-instance-by-name "fpDCS"))
(fp-set-value el-volume-dcs 'Name "el-volume-dcs")
(fp-add-lvalue el-volume-dcs 'Children el-volume)

;; Setzen der Randwerte fuer die Farbkodierung
(fp-set-value el-volume 'ValuesMin 0.0)
(fp-set-value el-volume 'ValuesMax 400.0)

;; Einbinden der Visualisierung in den Szenengraphen der aktuellen AVANGO Anwendung
(av-add el-volume-dcs)

;; Die Staebe der unter Hardware beschriebenen Cubic-Mouse werden in das AVANGO System als
;; Felder (des ad-sensors) eingebunden, welche am einen Ende des Stabes den Wert 0 und am
;; anderen Ende den Wert 1 haben. Die folgenden drei Zeilen verbinden die drei senkrecht zueinander
;; stehenden Schnitte durch den Datenraum mit den Feldern des ad-sensors, so dass diese
;; mit Hilfe der Cubic Mouse bewegt werden koennen
(fp-connect-from el-volume 'RelXSlicePos ad-sensor 'Value0)
(fp-connect-from el-volume 'RelYSlicePos ad-sensor 'Value1)
(fp-connect-from el-volume 'RelZSlicePos ad-sensor 'Value2)

;;;;;
;; Anschalten der Voxel Visualisierung
(fp-set-value el-volume 'ShowVoxels 1)

;;;;;
;; Ausschalten der drei senkrecht zueinander stehenden Schnittebenen
(fp-set-value el-volume 'ShowSections 0)

;;;;;
;; Setzen der Randbedingungen fuer die Voxel Visualisierung. In diesem Fall werden nur noch
;; die Zellen des Datenvolumens angezeigt, deren elektrischer Widerstand zwischen 200.0 und
;; 500.0 Ohm*m liegt.
(fp-set-value el-volume 'VoxBorderMin 200.0)
(fp-set-value el-volume 'VoxBorderMax 500.0)

```



```

//-----fdDataVol.c++-----
#include "fdDataVol.h"

static int makeFluxedGSet(pfFluxMemory *fmem)
{
    pfGeoState *gstate;
    pfGeoSet *gset;
    pfMaterial *material;

    if (fmem == NULL) {
        return pfFluxedGSetInit(fmem);
    }

    pfFluxedGSetInit(fmem);

    gset = (pfGeoSet *) fmem->getData();

    material = new(pfGetSharedArena()) pfMaterial;
    // material->setColor(PFMTL_EMISSION, 1.0, 1.0, 1.0);
    gstate = new(pfGetSharedArena()) pfGeoState();
    gstate->setAttr(PFSTATE_FRONTMTL, material);
    gstate->setAttr(PFSTATE_BACKMTL, material);
    gstate->setMode(PFSTATE_CULLFACE, PFTR_OFF);
    gstate->setMode(PFSTATE_TRANSPARENCY, PFTR_ON);
    gset->setGState(gstate);
    gset->setNumPrims(0);
    gset->setPrimType(PFGS_QUADS);
    return 0;
}

//
// Konstruktor
//
fdDataVol::fdDataVol(int numXBorders, int numYBorders, int numZBorders, float *xBorderPos,
                    float *yBorderPos, float *zBorderPos) :
    _numXBorders(numXBorders),
    _numYBorders(numYBorders),
    _numZBorders(numZBorders),
    _numXCells(numXBorders - 1),
    _numYCells(numYBorders - 1),
    _numZCells(numZBorders - 1),
    _values(NULL),
    _valMin(0.0),
    _valMax(0.0),
    _valRange(0.0),
    _texture(NULL),
    _texenv(NULL),
    _xyFluxedGSet(NULL),
    _xzFluxedGSet(NULL),
    _yzFluxedGSet(NULL),
    _voxFluxedCoords(NULL),
    _voxFluxedTexCoords(NULL),
    _voxFluxedNorms(NULL),
    _voxMaxNumberOfVertices(0),
    _voxNumberOfVertices(0),
    _ifVoxelVizIsShown(true),
    _ifSectionsAreShown(true)
{
    _dcs = new pfDCS;
    _xyGeode = new pfGeode;
    _xzGeode = new pfGeode;
    _yzGeode = new pfGeode;
    _dcs->addChild(_xyGeode);
    _dcs->addChild(_xzGeode);
    _dcs->addChild(_yzGeode);

    _syncGroup = pfFlux::getNamedSyncGroup("e1_volviz_syncgroup");

    _xyFluxedCoords = new pfFlux((_numXBorders-1) * (_numYBorders-1) * 4 *
                                sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
    _xyFluxedTexCoords = new pfFlux((_numXBorders-1) * (_numYBorders-1) * 4 *
                                    sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
    _xyFluxedCoords->setSyncGroup(_syncGroup);
    _xyFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
    _xyFluxedTexCoords->setSyncGroup(_syncGroup);
    _xyFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
    _xyFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
}

```

```

_xyFluxedGSet->setSyncGroup(_syncGroup);
_xyFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
80

pfFlux *xyBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
xyBounds->setSyncGroup(_syncGroup);
xyBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *xyGSet = (pfGeoSet*) _xyFluxedGSet->getWritableData();
xyGSet->setBoundFlux(xyBounds);
_xyFluxedGSet->writeComplete();

_xzFluxedCoords = new pfFlux((_numXBorders-1) * (_numZBorders-1) * 4 *
                             sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
_xzFluxedTexCoords = new pfFlux((_numXBorders-1) * (_numZBorders-1) * 4 *
                                 sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
90
_xzFluxedCoords->setSyncGroup(_syncGroup);
_xzFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xzFluxedTexCoords->setSyncGroup(_syncGroup);
_xzFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xzFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_xzFluxedGSet->setSyncGroup(_syncGroup);
_xzFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

pfFlux *xzBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
xzBounds->setSyncGroup(_syncGroup);
xzBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *xzGSet = (pfGeoSet*) _xzFluxedGSet->getWritableData();
xzGSet->setBoundFlux(xzBounds);
_xzFluxedGSet->writeComplete();
100

_yzFluxedCoords = new pfFlux((_numYBorders-1) * (_numZBorders-1) * 4 *
                              sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
_yzFluxedTexCoords = new pfFlux((_numYBorders-1) * (_numZBorders-1) * 4 *
                                  sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
110
_yzFluxedCoords->setSyncGroup(_syncGroup);
_yzFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_yzFluxedTexCoords->setSyncGroup(_syncGroup);
_yzFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_yzFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_yzFluxedGSet->setSyncGroup(_syncGroup);
_yzFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

pfFlux *yzBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
yzBounds->setSyncGroup(_syncGroup);
yzBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *yzGSet = (pfGeoSet*) _yzFluxedGSet->getWritableData();
yzGSet->setBoundFlux(yzBounds);
_yzFluxedGSet->writeComplete();
120

_xBorderPos = new float[_numXBorders];
_yBorderPos = new float[_numYBorders];
_zBorderPos = new float[_numZBorders];

int i;
130
for (i=0; i<_numXBorders; i++) _xBorderPos[i] = xBorderPos[i];
for (i=0; i<_numYBorders; i++) _yBorderPos[i] = yBorderPos[i];
for (i=0; i<_numZBorders; i++) _zBorderPos[i] = zBorderPos[i];
initTexture();
_fluxedVoxGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_fluxedVoxGSet->setSyncGroup(_syncGroup);
_fluxedVoxGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

pfGeoSet *voxGSet = (pfGeoSet*) _fluxedVoxGSet->getWritableData();
pfGeoState *voxGState = voxGSet->getGState();
140
voxGState->setAttr(PFSTATE_TEXTURE, _texture);
voxGState->setAttr(PFSTATE_TEXENV, _texenv);
voxGState->setMode(PFSTATE_ENTEXTURE, 1);
voxGState->setMode(PFSTATE_ENLIGHTING, PFTR_ON);
voxGState->setMode(PFSTATE_CULLFACE, PFTR_OFF);

_voxFluxedBBox = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
_voxFluxedBBox->setSyncGroup(_syncGroup);
_voxFluxedBBox->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfBox *bbox = (pfBox*) _voxFluxedBBox->getWritableData();
150
bbox->min = pfVec3(0.0, 0.0, 0.0);
bbox->max = pfVec3(0.0, 0.0, 0.0);
_voxFluxedBBox->writeComplete();
voxGSet->setBoundFlux(_voxFluxedBBox);

```



```

_voxGeode = new pfGeode;
_voxGeode->addGSet(voxGSet);
_dcs->addChild(_voxGeode);

_fluxedVoxGSet->writeComplete();
pfFlux::enableSyncGroup(_syncGroup);
}

//
// Konstruktor, der die Daten aus einer Datei einliest.
//
fdDataVol::fdDataVol(const char *filename) :
_values(NULL),
_valMin(0.0),
_valMax(0.0),
_valRange(0.0),
_texture(NULL),
_texenv(NULL),
_xyFluxedGSet(NULL),
_xzFluxedGSet(NULL),
_yzFluxedGSet(NULL),
_voxFluxedCoords(NULL),
_voxFluxedTexCoords(NULL),
_voxFluxedNorms(NULL),
_voxMaxNumberOfVertices(0),
_voxNumberOfVertices(0),
_ifVoxelVizIsShown(true),
_ifSectionsAreShown(true)
{
    int i, j;
    char word[255];
    char errorMsg[] = "fdDataVol::fdDataVol(char *filename) error while reading data";
    ifstream infile(filename);

    infile >> word;
    if (strcmp(word, "NUM_X_CELLS") == 0) {
        infile >> _numXBorders;
        _numXBorders += 1;
        cout << "Number of cell borders in x direction: " << _numXBorders << endl;
    } else {
        cout << errorMsg << endl;
        exit(1);
    }

    infile >> word;
    if (strcmp(word, "NUM_Y_CELLS") == 0) {
        infile >> _numYBorders;
        _numYBorders += 1;
        cout << "Number of cell borders in y direction: " << _numYBorders << endl;
    } else {
        cout << errorMsg << endl;
        exit(1);
    }

    infile >> word;
    if (strcmp(word, "NUM_Z_CELLS") == 0) {
        infile >> _numZBorders;
        _numZBorders += 1;
        cout << "Number of cell borders in z direction: " << _numZBorders << endl;
    } else {
        cout << errorMsg << endl;
        exit(1);
    }

    _xBorderPos = new float[_numXBorders];
    _yBorderPos = new float[_numYBorders];
    _zBorderPos = new float[_numZBorders];

    _numXCells = _numXBorders - 1;
    _numYCells = _numYBorders - 1;
    _numZCells = _numZBorders - 1;

    infile >> word;
    if (strcmp(word, "POS_X_CELL_BOUNDS") == 0) {

```

```

    for (i=0; i<_numXBorders; i++) {
        infile >> _xBorderPos[i];
    }
} else {
    cout << errmsg << endl;
    exit(1);
}

infile >> word;
if (strcmp(word, "POS_Y_CELL_BOUNDS") == 0) {
    for (i=0; i<_numYBorders; i++) {
        infile >> _yBorderPos[i];
    }
} else {
    cout << errmsg << endl;
    exit(1);
}

infile >> word;
if (strcmp(word, "POS_Z_CELL_BOUNDS") == 0) {
    for (i=0; i<_numZBorders; i++) {
        infile >> _zBorderPos[i];
    }
} else {
    cout << errmsg << endl;
    exit(1);
}

int numXVals = _numXBorders - 1;
int numYVals = _numYBorders - 1;
int numZVals = _numZBorders - 1;

_values = new float**[numXVals];
for (i=0; i<numXVals; i++) {
    _values[i] = new float*[numYVals];
    for (j=0; j<numYVals; j++) {
        _values[i][j] = new float[numZVals];
    }
}

// einlesen der Daten in xy scheiben von unten nach oben, erst entlang x, dann entlang y
for (int zIndex=0; zIndex < numZVals; zIndex++) {
    for (int yIndex=0; yIndex < numYVals; yIndex++) {
        for (int xIndex=0; xIndex < numXVals; xIndex++) {
            infile >> _values[xIndex][yIndex][zIndex];
        }
    }
}

infile.close();

evaluateValuesMinMax();

_dcs = new pfDCS;
_xyGeode = new pfGeode;
_xzGeode = new pfGeode;
_yzGeode = new pfGeode;

_syncGroup = pfFlux::getNamedSyncGroup("e1_volviz_syncgroup");

_xyFluxedCoords = new pfFlux((_numXBorders-1) * (_numYBorders-1) * 4 *
    sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
_xyFluxedTexCoords = new pfFlux((_numXBorders-1) * (_numYBorders-1) * 4 *
    sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
_xyFluxedCoords->setSyncGroup(_syncGroup);
_xyFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xyFluxedTexCoords->setSyncGroup(_syncGroup);
_xyFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xyFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_xyFluxedGSet->setSyncGroup(_syncGroup);
_xyFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

pfFlux *xyBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
xyBounds->setSyncGroup(_syncGroup);
xyBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *xyGSet = (pfGeoSet*) _xyFluxedGSet->getWritableData();
xyGSet->setBoundFlux(xyBounds);

```

```

_xyFluxedGSet->writeComplete();
310

_xzFluxedCoords = new pfFlux((_numXBorders-1) * (_numZBorders-1) * 4 *
                             sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
_xzFluxedTexCoords = new pfFlux((_numXBorders-1) * (_numZBorders-1) * 4 *
                             sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
_xzFluxedCoords->setSyncGroup(_syncGroup);
_xzFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xzFluxedTexCoords->setSyncGroup(_syncGroup);
_xzFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_xzFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_xzFluxedGSet->setSyncGroup(_syncGroup);
_xzFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
320

pfFlux *xzBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
xzBounds->setSyncGroup(_syncGroup);
xzBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *xzGSet = (pfGeoSet*) _xzFluxedGSet->getWritableData();
xzGSet->setBoundFlux(xzBounds);
_xzFluxedGSet->writeComplete();
330

_yzFluxedCoords = new pfFlux((_numYBorders-1) * (_numZBorders-1) * 4 *
                             sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
_yzFluxedTexCoords = new pfFlux((_numYBorders-1) * (_numZBorders-1) * 4 *
                             sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
_yzFluxedCoords->setSyncGroup(_syncGroup);
_yzFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_yzFluxedTexCoords->setSyncGroup(_syncGroup);
_yzFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
_yzFluxedGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_yzFluxedGSet->setSyncGroup(_syncGroup);
_yzFluxedGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
340

pfFlux *yzBounds = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
yzBounds->setSyncGroup(_syncGroup);
yzBounds->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfGeoSet *yzGSet = (pfGeoSet*) _yzFluxedGSet->getWritableData();
yzGSet->setBoundFlux(yzBounds);
_yzFluxedGSet->writeComplete();
350

cout << "initializing new texture" << endl;
initTexture();
if (!_texture == NULL) {
    cout << "error, texture is not initialized" << endl;
}

_fluxedVoxGSet = new pfFlux(makeFluxedGSet, PFFLUX_DEFAULT_NUM_BUFFERS);
_fluxedVoxGSet->setSyncGroup(_syncGroup);
_fluxedVoxGSet->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
360

pfGeoSet *voxGSet = (pfGeoSet*) _fluxedVoxGSet->getWritableData();
pfGeoState *voxGState = voxGSet->getGState();
voxGState->setAttr(PFSTATE_TEXTURE, _texture);
voxGState->setAttr(PFSTATE_TEXENV, _texenv);
voxGState->setMode(PFSTATE_ENTEXTURE, 1);
voxGState->setMode(PFSTATE_ENLIGHTING, PFTR_ON);
voxGState->setMode(PFSTATE_CULLFACE, PFTR_OFF);

_voxFluxedBBox = new pfFlux(sizeof(pfBox), PFFLUX_DEFAULT_NUM_BUFFERS);
_voxFluxedBBox->setSyncGroup(_syncGroup);
_voxFluxedBBox->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);
pfBox *bbox = (pfBox*) _voxFluxedBBox->getWritableData();
bbox->min = pfVec3(0.0, 0.0, 0.0); // Wird spaeter korrekt gesetzt
bbox->max = pfVec3(0.0, 0.0, 0.0); // ""
_voxFluxedBBox->writeComplete();
voxGSet->setBoundFlux(_voxFluxedBBox);
370

_voxGeode = new pfGeode;
_voxGeode->addGSet(voxGSet);
_dcs->addChild(_voxGeode);
380

_fluxedVoxGSet->writeComplete();

pfFlux::enableSyncGroup(_syncGroup);

```

```

}

fdDataVol::~fdDataVol()
{
    pfDelete(_dcs);
    pfDelete(_xyGeode);
    pfDelete(_xzGeode);
    pfDelete(_yzGeode);

    pfDelete(_xyFluxedCoords);
    pfDelete(_xyFluxedTexCoords);
    pfDelete(_xyGSet);

    pfDelete(_xzFluxedCoords);
    pfDelete(_xzFluxedTexCoords);
    pfDelete(_xzGSet);

    pfDelete(_yzFluxedCoords);
    pfDelete(_yzFluxedTexCoords);
    pfDelete(_yzGSet);

    pfDelete(_texture);
    pfDelete(_texenv);

    delete [] _yBorderPos;
    delete [] _zBorderPos;

    delete [] _xBorderPos;
    for (int i=0; i<_numXCells; i++) {
        for (int j=0; j<_numYCells; j++) {
            delete [] _values[i][j];
        }
        delete [] _values[i];
    }
    delete [] _values;
}

void fdDataVol::evaluateValuesMinMax()
{
    _valMin = 1000000000.0;
    _valMax = -1000000000.0;
    for (int xInd=0; xInd<_numXCells; xInd++) {
        for (int yInd=0; yInd<_numYCells; yInd++) {
            for (int zInd=0; zInd<_numZCells; zInd++) {
                if (_values[xInd][yInd][zInd] != VAL_INVALID_VALUES) {
                    if (_values[xInd][yInd][zInd] < _valMin) {
                        _valMin = _values[xInd][yInd][zInd];
                    }
                    if (_values[xInd][yInd][zInd] > _valMax) {
                        _valMax = _values[xInd][yInd][zInd];
                    }
                }
            }
        }
    }
    _valRange = _valMax - _valMin;
}

void fdDataVol::writeGridInformation()
{
    cout << "-----general information about the grid-----" << endl;
    cout << " number of cells: " << _numXCells << " " << _numYCells << " " << _numZCells << endl;
    cout << " values: " << endl;
    cout << " Min: " << _valMin << endl;
    cout << " Max: " << _valMax << endl;
    cout << " Range: " << _valRange << endl;
    cout << " x-Boundaries: " << endl;
    for (int i=0; i<_numXBorders; i++) {
        cout << " " << _xBorderPos[i] << endl;
    }
    cout << " y-Boundaries: " << endl;
    for (i=0; i<_numYBorders; i++) {
        cout << " " << _yBorderPos[i] << endl;
    }
    cout << " z-Boundaries: " << endl;
    for (i=0; i<_numZBorders; i++) {
        cout << " " << _zBorderPos[i] << endl;
}

```

```

    }
    cout << "-----" << endl;
}

void fdDataVol::writeInformation()
{
    writeGridInformation();
    cout << "values statistics" << endl;
    cout << " min: " << _valMin << endl;
    cout << " max: " << _valMax << endl;
    cout << " range: " << _valRange << endl;
    cout << endl;
}

pfDCS *fdDataVol::getDCS()
{
    return _dcs;
}

void fdDataVol::initTexture()
{
    uint *colors = (uint*) pfMemory::malloc(256 * sizeof(uint), pfGetSharedArena());
    float frac = 1.0 / 255.0;
    float r, g, b;
    for (int i=0; i<128; i++) {
        float colorval = 2.0 * frac * (float) i;
        r = 1.0 - colorval;
        g = colorval;
        b = 0.0;
        colors[i] = getTexel(r, g, b, 1.0);
    }
    for (i=128; i<256; i++) {
        float colorval = 2.0 * frac * (float) (i-128);
        r = 0.0;
        g = 1.0 - colorval;
        b = colorval;
        colors[i] = getTexel(r, g, b, 1.0);
    }

    _texture = new(pfGetSharedArena()) pfTexture();
    _texenv = new(pfGetSharedArena()) pfTexEnv();
    _texenv->setMode(PFTE_MODULATE);

    _texture->setImage((uint *) colors, 4, 1, 256, 1);
    _texture->setFormat(PFTEX_EXTERNAL_FORMAT, PFTEX_PACK_8);
    _texture->setFilter(PFTEX_MAGFILTER, PFTEX_LINEAR);
    _texture->setFilter(PFTEX_MINFILTER, PFTEX_LINEAR);
    _texture->setFormat(PFTEX_IMAGE_FORMAT, PFTEX_RGBA);
}

void fdDataVol::calculationsFinished()
{
    pfFlux::syncGroupReady(_syncGroup);
}

void fdDataVol::initXYSection(float initialZ)
{
    pfGeoSet *xyGSet = (pfGeoSet*) _xyFluxedGSet->getWritableData();
    xyGSet->setAttr(PFGS_COORD3 ,PFGS_PER_VERTEX, _xyFluxedCoords, NULL);
    xyGSet->setAttr(PFGS_TEXCOORD2 ,PFGS_PER_VERTEX, _xyFluxedTexCoords, NULL);

    pfVec3 *coords = (pfVec3 *) _xyFluxedCoords->getWritableData();
    pfVec2 *tcoords = (pfVec2 *) _xyFluxedTexCoords->getWritableData();

    int xMaxInd = _numXBorders - 1;
    int yMaxInd = _numYBorders - 1;

    int zIndex = getIndexForZPos(initialZ);

    int vecInd = 0;
    int numPolygons = 0;

    for (int i=0; i<xMaxInd; i++) {
        for (int j=0; j<yMaxInd; j++) {
            if (_values[i][j][zIndex] != VAL_INVALID_VALUES) {
                coords[vecInd + 0].set(_xBorderPos[i], _yBorderPos[j], initialZ);
            }
        }
    }
}

```

```

coords[vecInd + 1].set(_xBorderPos[i], _yBorderPos[j+1], initialZ);
coords[vecInd + 2].set(_xBorderPos[i+1], _yBorderPos[j+1], initialZ);
coords[vecInd + 3].set(_xBorderPos[i+1], _yBorderPos[j], initialZ);

float fraction = (_values[i][j][zIndex] - _valMin) / _valRange;
if (fraction < 0.0) fraction = 0.0;
if (fraction > 1.0) fraction = 1.0;

tcoords[vecInd + 0].set(0.0, fraction);
tcoords[vecInd + 1].set(0.0, fraction);
tcoords[vecInd + 2].set(0.0, fraction);
tcoords[vecInd + 3].set(0.0, fraction);
vecInd += 4;
numPolygons += 1;
}
}
}

pfMaterial *material = new(pfGetSharedArena()) pfMaterial();
material->setColor(PFMTL_EMISSION, 0.6, 0.6, 0.6);
pfGeoState *gstate = new(pfGetSharedArena()) pfGeoState();

xyGSet->setPrimType(PFGS_QUADS);
xyGSet->setNumPrims(numPolygons);

gstate->setAttr(PFSTATE_TEXTURE, _texture);
gstate->setAttr(PFSTATE_TEXENV, _texenv);
gstate->setMode(PFSTATE_ENTEXTURE, 1);
gstate->setMode(PFSTATE_CULLFACE, PFTR_OFF);
gstate->setAttr(PFSTATE_FRONTMTL, material);
gstate->setAttr(PFSTATE_BACKMTL, material);

xyGSet->setGState(gstate);

pfFlux *boundFlux = xyGSet->getBoundFlux();
pfBox* box = (pfBox *) boundFlux->getWritableData();
box->around(coords, numPolygons * 4);
pfSphere boundSphere;
_xyGeode->getBound(&boundSphere);
boundSphere.around(coords, numPolygons * 4);
boundFlux->writeComplete();

_xyGeode->addGSet(xyGSet);

_dcs->addChild(_xyGeode);

pfSphere sphere;
_xyGeode->getBound(&sphere);

_xyFluxedCoords->writeComplete();
_xyFluxedTexCoords->writeComplete();
_xyFluxedGSet->writeComplete();
}

void fdDataVol::initXZSection(float initialY)
{
//
// Analog zu fdDataVol::initXYSection(float initialZ)
// (siehe oben)
//
}

void fdDataVol::initYZSection(float initialX)
{
//
// Analog zu fdDataVol::initXYSection(float initialZ)
// (siehe oben)
//
}

void fdDataVol::showSectionViz()
{
if (!_ifSectionsAreShown) {
_ifSectionsAreShown = true;
_dcs->addChild(_xyGeode);
}
}

```

```

        _dcs->addChild(_xzGeode);
        _dcs->addChild(_yzGeode);
    }
}
620

void fdDataVol::removeSectionViz()
{
    if (!_ifSectionsAreShown) {
        _ifSectionsAreShown = false;
        _dcs->removeChild(_xyGeode);
        _dcs->removeChild(_xzGeode);
        _dcs->removeChild(_yzGeode);
    }
}
630

void fdDataVol::setXYSection(float zPos)
{
    pfGeoSet *xyGSet = (pfGeoSet*) _xyFluxedGSet->getWritableData();

    if (zPos < _zBorderPos[0]) {
        zPos = _zBorderPos[0];
    } else if (zPos > _zBorderPos[_numZBorders - 1]) {
        zPos = _zBorderPos[_numZBorders - 1];
    }
640
    pfVec3 *coords = (pfVec3 *) _xyFluxedCoords->getWritableData();
    pfVec2 *tcoords = (pfVec2 *) _xyFluxedTexCoords->getWritableData();

    int xMaxInd = _numXBorders - 1;
    int yMaxInd = _numYBorders - 1;

    int zIndex = getIndexForZPos(zPos);
    if (zIndex == (_numZBorders - 1)) zIndex -= 1; // right border is treated as belonging to the cell before

    int vecInd = 0;
    int numPolygons = 0;
650

    for (int i=0; i<xMaxInd; i++) {
        for (int j=0; j<yMaxInd; j++) {
            if (_values[i][j][zIndex] != VAL_INVALID_VALUES) {
                coords[vecInd + 0].set(_xBorderPos[i], _yBorderPos[j], zPos);
                coords[vecInd + 1].set(_xBorderPos[i], _yBorderPos[j+1], zPos);
                coords[vecInd + 2].set(_xBorderPos[i+1], _yBorderPos[j+1], zPos);
                coords[vecInd + 3].set(_xBorderPos[i+1], _yBorderPos[j], zPos);
660

                float fraction = (_values[i][j][zIndex] - _valMin) / _valRange;
                if (fraction < 0.0) fraction = 0.0;
                if (fraction > 1.0) fraction = 1.0;

                tcoords[vecInd + 0].set(0.0, fraction);
                tcoords[vecInd + 1].set(0.0, fraction);
                tcoords[vecInd + 2].set(0.0, fraction);
                tcoords[vecInd + 3].set(0.0, fraction);
                vecInd += 4;
                numPolygons += 1;
670
            }
        }
    }

    xyGSet->setNumPrims(numPolygons);

    pfFlux *boundFlux = xyGSet->getBoundFlux();
    pfBox* box = (pfBox *) boundFlux->getWritableData();
    box->around(coords, numPolygons * 4);
    boundFlux->writeComplete();
680

    _xyGeode->setBound(NULL, PFBOUND_DYNAMIC);

    _xyFluxedCoords->writeComplete();
    _xyFluxedTexCoords->writeComplete();
    _xyFluxedGSet->writeComplete();
}

void fdDataVol::setXZSection(float yPos)
{
690
    //
    // Analog zur Funktion void fdDataVol::setXYSection(float zPos)
    // (siehe oben)

```

```

//
}

void fdDataVol::setYZSection(float xPos)
{
//
// Analog zur Funktion void fdDataVol::setXYSection(float zPos)
// (siehe oben)
//
}
700

void fdDataVol::setXYSectionRel(float zRelPos)
{
float zPos = getZPositionFromRelativeValue(zRelPos);
setXYSection(zPos);
}
710

void fdDataVol::setXZSectionRel(float yRelPos)
{
float yPos = getYPositionFromRelativeValue(yRelPos);
setXZSectionRel(yPos);
}

void fdDataVol::setYZSectionRel(float xRelPos)
{
float xPos = getXPositionFromRelativeValue(xRelPos);
setYZSection(xPos);
}
720

void fdDataVol::setMinValue(float min)
{
_valMin = min;
_valRange = _valMax - _valMin;
}

void fdDataVol::setMaxValue(float max)
{
_valMax = max;
_valRange = _valMax - _valMin;
}
730

float fdDataVol::getXPositionFromRelativeValue(float relVal)
{
float returnvalue = (1.0 - relVal) * _xBorderPos[0] + relVal * _xBorderPos[_numXBorders - 1];
if (returnvalue < _xBorderPos[0]) return _xBorderPos[0];
if (returnvalue > _xBorderPos[_numXBorders - 1]) return _xBorderPos[_numXBorders - 1];
return returnvalue;
}
740

float fdDataVol::getYPositionFromRelativeValue(float relVal)
{
float returnvalue = (1.0 - relVal) * _yBorderPos[0] + relVal * _yBorderPos[_numYBorders - 1];
if (returnvalue < _yBorderPos[0]) return _yBorderPos[0];
if (returnvalue > _yBorderPos[_numYBorders - 1]) return _yBorderPos[_numYBorders - 1];
return returnvalue;
}
750

float fdDataVol::getZPositionFromRelativeValue(float relVal)
{
float returnvalue = (1.0 - relVal) * _zBorderPos[0] + relVal * _zBorderPos[_numZBorders - 1];
if (returnvalue < _zBorderPos[0]) return _zBorderPos[0];
if (returnvalue > _zBorderPos[_numZBorders - 1]) return _zBorderPos[_numZBorders - 1];
return returnvalue;
}

int fdDataVol::getIndexForXPos(float xPos)
{
if (xPos < _xBorderPos[0]) {
cout << "fdDataVol::getIndexForXPos(float xPos) xPos: " << xPos << " first border: " << _xBorderPos[0] << endl;
return -1;
} else if (xPos > _xBorderPos[_numXBorders - 1]) {
return -2;
}
}
int endXIndex = _numXBorders - 1;
for (int i=0; i<endXIndex; i++) {
if ((xPos >= _xBorderPos[i]) && (xPos < _xBorderPos[i+1])) {
return i;
}
}
}
760
770

```



```

    }
  }
  return (_numXBorders - 1);
}

int fdDataVol::getIndexForYPos(float yPos)
{
  //
  // Analog zur Funktion int fdDataVol::getIndexForXPos(float xPos)
  // (siehe oben)
  //
}
880

int fdDataVol::getIndexForZPos(float zPos)
{
  //
  // Analog zur Funktion int fdDataVol::getIndexForXPos(float xPos)
  // (siehe oben)
  //
}
890

uint fdDataVol::getTexel(float r, float g, float b, float alpha)
{
  // rgba Werte muessen !!!! kleiner als 255 sein
  uint returnWert = 0;

  if (255 < ((uint) r * 255.0)) {cout << "wrong argument for red value! " << endl; }
  returnWert = returnWert | (uint) (r * 255.0);
  returnWert = returnWert << 8;
800

  if (255 < ((uint) g * 255.0)) {cout << "wrong argument for green value! " << endl; }
  returnWert = returnWert | (uint) (g * 255.0);
  returnWert = returnWert << 8;

  if (255 < ((uint) b * 255.0)) {cout << "wrong argument for blue value! " << endl; }
  returnWert = returnWert | (uint) (b * 255.0);
  returnWert = returnWert << 8;

  if (255 < ((uint) alpha * 255.0)) {cout << "wrong argument for alpha value! " << endl; }
  returnWert = returnWert | (uint) (alpha * 255.0);
  return returnWert;
810
}

void fdDataVol::loadColormapFromFile(const char *filename)
{
  if (_texture->loadFile(filename)) {
    cout << "colormap loaded successfully from file " << filename << endl;
  }
  int format = _texture->getFormat(PFTEX_IMAGE_FORMAT);
  if (format == PFTEX_RGB) {
820
    cout << "new colormap format: is rgb" << endl;
  } else if (format == PFTEX_RGBA) {
    cout << "new colormap format: is rgba" << endl;
  } else if (format == PFTEX_LUMINANCE) {
    cout << "new colormap format: is luminance" << endl;
  } else {
    cout << "new colormap has unrecognized format" << endl;
  }
  _texture->setFormat(PFTEX_IMAGE_FORMAT, PFTEX_RGB);
  _texture->setRepeat(PFTEX_WRAP, PFTEX_CLAMP);
830
}

bool fdDataVol::isXYGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relY)
{
  pfGeoSet *xyGSet = (pfGeoSet *) _xyFluxedGSet->getCurData();
  if (gset == xyGSet) {
    if (hitPos[0] <= _xBorderPos[0]) {
      relX = 0.0;
    } else if (hitPos[0] >= _xBorderPos[_numXBorders-1]) {
      relX = 1.0;
840
    } else {
      relX = (hitPos[0] - _xBorderPos[0]) / (_xBorderPos[_numXBorders-1] - _xBorderPos[0]);
    }

    if (hitPos[1] <= _yBorderPos[0]) {
      relY = 0.0;
    } else if (hitPos[1] >= _yBorderPos[_numYBorders-1]) {

```

```

    relY = 1.0;
  } else {
    relY = (hitPos[1] - _yBorderPos[0]) / (_yBorderPos[_numYBorders-1] - _yBorderPos[0]);
  }
  return true;
}
return false;
}

bool fdDataVol::isXZGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relZ)
{
  //
  // Analog zur Funktion bool fdDataVol::isXYGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relY)
  // (siehe vor)
  //
}

bool fdDataVol::isYZGSet(pfVec3 hitPos, pfGeoSet *gset, float &relY, float &relZ)
{
  //
  // Analog zur Funktion bool fdDataVol::isXYGSet(pfVec3 hitPos, pfGeoSet *gset, float &relX, float &relY)
  // (siehe vor)
  //
}

void fdDataVol::initVoxelViz()
{
  int xyNums = _numXCells * _numYCells;
  int xzNums = _numXCells * _numZCells;
  int yzNums = _numYCells * _numZCells;

  // number of Cells, sufficient to visualize twice times the hull of the volume
  int numCells = 4 * (xyNums + xzNums + yzNums);
  int numVerts = 4 * numCells;

  if (_voxFluxedCoords != NULL) {
    pfDelete(_voxFluxedCoords);
  }
  if (_voxFluxedTexCoords != NULL) {
    pfDelete(_voxFluxedTexCoords);
  }
  if (_voxFluxedNorms != NULL) {
    pfDelete(_voxFluxedNorms);
  }

  pfGeoSet *gset = (pfGeoSet*) _fluxedVoxGSet->getWritableData();
  gset->setNumPrims(0);

  _voxNumberOfVertices = 0;

  _voxFluxedCoords = new pfFlux(numVerts * sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
  _voxFluxedCoords->setSyncGroup(_syncGroup);
  _voxFluxedCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

  _voxFluxedTexCoords = new pfFlux(numVerts * sizeof(pfVec2), PFFLUX_DEFAULT_NUM_BUFFERS);
  _voxFluxedTexCoords->setSyncGroup(_syncGroup);
  _voxFluxedTexCoords->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

  _voxFluxedNorms = new pfFlux(numCells * sizeof(pfVec3), PFFLUX_DEFAULT_NUM_BUFFERS);
  _voxFluxedNorms->setSyncGroup(_syncGroup);
  _voxFluxedNorms->setMode(PFFLUX_COPY_LAST_DATA, PF_ON);

  gset->setAttr(PFGS_COORD3, PFGS_PER_VERTEX, _voxFluxedCoords, NULL);
  gset->setAttr(PFGS_TEXCOORD2, PFGS_PER_VERTEX, _voxFluxedTexCoords, NULL);
  gset->setAttr(PFGS_NORMAL3, PFGS_PER_PRIM, _voxFluxedNorms, NULL);

  pfBox *bbox = (pfBox*) _voxFluxedBBox->getWritableData();
  bbox->min.set(_xBorderPos[0], _yBorderPos[0], _zBorderPos[0]);
  bbox->max.set(_xBorderPos[_numXBorders-1], _yBorderPos[_numYBorders-1], _zBorderPos[_numZBorders-1]);
  _voxFluxedBBox->writeComplete();

  _voxGeode->setBound(NULL, PFBOUND_DYNAMIC);
  pfSphere sphere;

  _voxMaxNumberOfVertices = numVerts;
  _fluxedVoxGSet->writeComplete();
}

```

```

void fdDataVol::voxAddCellBorder(int xIndex, int yIndex, int zIndex, int border, pfVec3 *coords,
                                pfVec2 *texCoords, pfVec3 *norms, float cTablePos)
{
    if ((_voxNumberOfVertices + 4) >= _voxMaxNumberOfVertices) {
        cout << "WARNING: fdDataVol::voxAddCellBorder(...): number of added Cell-Borders is to large" << endl;
        return;
    }

    int startInd = _voxNumberOfVertices;
    int normsStartInd = _voxNumberOfVertices / 4;

    if (border == CONSTX_LOW_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        coords[startInd + 1].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        coords[startInd + 2].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex+1]);
        coords[startInd + 3].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        norms[normsStartInd].set(-1.0, 0.0, 0.0);
    } else if (border == CONSTX_HIGH_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        coords[startInd + 1].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex + 1]);
        coords[startInd + 2].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        coords[startInd + 3].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        norms[normsStartInd].set(1.0, 0.0, 0.0);
    } else if (border == CONSTY_LOW_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        coords[startInd + 1].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex+1]);
        coords[startInd + 2].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex+1]);
        coords[startInd + 3].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        norms[normsStartInd].set(0.0, -1.0, 0.0);
    } else if (border == CONSTY_HIGH_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        coords[startInd + 1].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        coords[startInd + 2].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        coords[startInd + 3].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        norms[normsStartInd].set(0.0, 1.0, 0.0);
    } else if (border == CONSTZ_LOW_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        coords[startInd + 1].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex]);
        coords[startInd + 2].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        coords[startInd + 3].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex]);
        norms[normsStartInd].set(0.0, 0.0, -1.0);
    } else if (border == CONSTZ_HIGH_BORDER) {
        coords[startInd].set(_xBorderPos[xIndex], _yBorderPos[yIndex], _zBorderPos[zIndex+1]);
        coords[startInd + 1].set(_xBorderPos[xIndex], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        coords[startInd + 2].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex+1], _zBorderPos[zIndex+1]);
        coords[startInd + 3].set(_xBorderPos[xIndex+1], _yBorderPos[yIndex], _zBorderPos[zIndex+1]);
        norms[normsStartInd].set(0.0, 0.0, 1.0);
    } else {
        return;
    }

    _voxNumberOfVertices += 4;
    texCoords[startInd].set(0.0, cTablePos);
    texCoords[startInd + 1].set(0.0, cTablePos);
    texCoords[startInd + 2].set(0.0, cTablePos);
    texCoords[startInd + 3].set(0.0, cTablePos);
}

void fdDataVol::voxSetBorders(float lowerBorder, float upperBorder)
{
    initVoxelViz();

    pfVec3 *coords = (pfVec3*) _voxFluxedCoords->getWritableData();
    pfVec2 *texCoords = (pfVec2*) _voxFluxedTexCoords->getWritableData();
    pfVec3 *norms = (pfVec3*) _voxFluxedNorms->getWritableData();
    int i, j, k;
    for (i=0; i<_numXCells; i++) {
        for (j=0; j<_numYCells; j++) {
            for (k=0; k<_numZCells; k++) {
                if ((_values[i][j][k] >= lowerBorder) && (_values[i][j][k] <= upperBorder)) {
                    float fraction = (_values[i][j][k] - _valMin) / _valRange;
                    if (fraction < 0.0) fraction = 0.0;
                    if (fraction > 1.0) fraction = 1.0;

                    if (i == 0) {
                        voxAddCellBorder(i, j, k, CONSTX_LOW_BORDER, coords, texCoords, norms, fraction);
                    }
                }
            }
        }
    }
}

```

