

Geo- Processing Reihe

Geoprocessing Series



Stefan HUBER

GEO3D:
INTERAKTIVE 3-D-VISUALISIERUNG

Geographisches Institut
Universität Zürich, 2000

Vol. 36

Diese Arbeit wurde 1999 von der
Mathematisch-naturwissenschaftlichen Fakultät
der Universität Zürich als Diplomarbeit angenommen.

Copyright 2000:
Geographisches Institut
Universität Zürich-Irchel

All rights reserved



Geo3D: Interaktive 3-D-Visualisierung

Diplomarbeit
am Geographischen Institut
der Universität Zürich

von Stefan Huber

unter der Leitung von
Prof. Dr. K. Brassel
und dipl. phys. M. Heller

Zürich 1999

Dank

Die Gewandtheit im Formulieren der Danksagungen charakterisiert den Wissenschaftler von Rang. Es kann vorkommen, dass ein Wissenschaftler an Ende seiner Arbeit entdeckt, dass er niemandem Dank schuldet. Macht nichts, dann muss er Dankeschulden erfinden. Eine Forschung ohne Dankeschulden ist suspekt ...

UMBERTO ECO «WIE MAN EIN VORWORT SCHREIBT»

Da ich in dieser Arbeit genug zu «erfinden» hatte, brauche ich es hier nicht zu tun und danke herzlich ...

Allen voran *dipl. phys. Martin Heller*: Er war der unermüdliche Ideenlieferant, Diskussionspartner, Datenbeschaffer und gab mir den Kick, endlich programmieren zu lernen;

Dr. Bernhard Schneider für die kritische Durchsicht dieses Textes;

Prof. Kurt Brassel für das Umfeld und die Infrastruktur am Geographischen Institut;

Kurt Zimmermann und *Martin Sluka* für ihre Diskussionslust und die Modelle, die ich veröffentlichen durfte;

Bernard Ournié für die Übersetzung von **Geo3D** samt Manual ins Französische;

All den «*People@Somewhere.Earth*», die sich übers Internet an der Diskussion beteiligt haben.

Zusammenfassung

Die Visualisierung von Daten ist seit jeher eine der zentralen Aufgaben des Fachbereichs Geographie. Oft sind diese Daten dreidimensional (3-D). Kleinere 3-D-Datensätze können inzwischen auf neueren Computern der unteren Preisklasse (Lowend-Computer) – und damit von einer breiten Anwenderschaft – interaktiv visualisiert werden. Interaktive 3-D-Visualisierung kann einerseits Interpretation und visuelle Analyse, andererseits Kommunikation und Präsentation von Resultaten unterstützen.

Ziel dieses Projekts ist die Konzeption und Entwicklung einer Applikation (**Geo3D**) zur interaktiven Visualisierung von 3-D-Daten auf Lowend-Computern. Als Testdaten wurden Höhlen- und Geländemodelle eingesetzt. Die Applikation **Geo3D** ist übers Internet frei erhältlich und leistet damit einen Beitrag zur Verbreitung von 3-D-Anwendungen im geographischen Umfeld.

In einem Grundlagenkapitel werden die Voraussetzungen und Grenzen verschiedener Methoden der Visualisierung von 3-D-Daten gezeigt. Anschließend werden ausgewählte Grundlagen der 3-D-Computergraphik und die in **Geo3D** verwendete 3-D-Funktionsbibliothek *QuickDraw 3D* vorgestellt.

Die Navigation im virtuellen Raum ist das zentrale Thema des **Geo3D**-Konzepts. Dabei soll die Synthese von menschlicher Wahrnehmung und geometrischen Grundlagen zu einem sinnvollen Konzept der Navigation führen. Darstellungsqualität, Mittel zur visuellen Analyse der Daten und Kommunikation zwischen Applikationen sind weitere Themen.

Die vielfältigen Einsatzmöglichkeiten wie auch die Grenzen von **Geo3D** in der Visualisierung von 3-D-Daten werden im nächsten Kapitel besprochen. Es wird gezeigt, dass die enge Verzahnung von Applikationen zur Datenmodellierung und -visualisierung sich als zukunftssträftig erweist. Nach den Schlussfolgerungen werden Vorschläge für zukünftige Arbeiten gemacht. **Geo3D** samt Manual ist auf der beiliegenden CD-ROM zu finden.

Summary

The visualization of data has always been one of the central tasks in the field of geography. Often these data are three-dimensional (3D). Nowadays, smaller data records can be visualized interactively even on personal computers. Thus interactive 3D-visualization become accessible to a large number of users. On the one hand, interactive 3D-visualization facilitates the interpretation and the visual analysis of data. On the other hand it makes communication and the presentation of research results more accessible.

The aim of this project is to conceive and to develop an application (**Geo3D**) for the interactive visualization of 3D-data on personal computers. Terrain models and models of caves were used as test data. **Geo3D** is freely accessible over internet and thus supports the spread of 3D-visualization in the field of geography.

In a chapter on basics, the prerequisites and the limits of various methods of visualization are being discussed. Then we will present selected fundamentals of 3D-computer graphics and the 3D-graphics library *QuickDraw 3D* used in **Geo3D**.

Navigation in virtual space is the central issue in the concept of **Geo3D**. Based on a synthesis of human perception and geometrical basics, we will develop a meaningful tool of navigation. The quality of representation, means for the visual analysis of data, and the communication between applications are further topics.

The many possibilities of using **Geo3D** but also the limits of its use are being discussed in the next chapter. We will show that the communication among modeling and visualization applications has proved promising for future projects. Suggestions for further work can be found after the conclusions. **Geo3D** and its manual are accessible on the enclosed CD-ROM.

Inhalt

	Dank	i
	Zusammenfassung	iii
	Summary	iv
	Inhalt	v
	Abbildungen und Tabellen	vii
Kapitel 1	Ausgangslage und Fragestellung	I
	1.1 Ausgangslage	1
	1.2 Zielsetzung	4
	1.3 Sprachliche Regelung der verwendeten Begriffe	5
	1.4 Verwendete Daten	6
	1.5 Aufbau der Arbeit	6
Kapitel 2	Grundlagen und Voraussetzungen	7
	2.1 Methoden der Visualisierung von 3-D-Daten	7
	2.1.1 Physische Modelle	7
	2.1.2 Visualisierung auf Papier	8
	2.1.3 Visualisierung auf dem Bildschirm	8
	2.2 Methoden der Computergraphik	10
	2.2.1 Rendering, Imaging und Geometrisches Modell	10
	2.2.2 Rendering-Methoden	11
	2.2.3 Kamera	12
	2.2.4 Beleuchtung und Interpolation	14
	2.2.5 Animation	17
	2.3 QuickDraw 3D (QD3D)	18
	2.3.1 QuickDraw 3D-Architektur	18
	2.3.2 3D Metafile Format (3DMF)	21

Kapitel 3	«Geo3D»-Konzept	25
3.1	Einleitung	25
3.2	Bemerkungen zum User-Interface.....	26
3.3	Navigation	27
3.3.1	Alltäglicher vs. synthetischer Raum.....	27
3.3.2	Zur Geometrie der Navigation.....	29
3.3.3	Navigations-Konzept.....	32
3.3.4	Aufzeichnung und Wiedergabe.....	35
3.3.5	Navigation und User-Interface.....	36
3.4	Rendering-Qualität	37
3.5	Analyse-Features	40
3.5.1	Koordinatenwerte und Distanz.....	41
3.5.2	Geraden und Ebenen.....	41
3.5.3	Mehrfache Ansichten eines Modells.....	42
3.5.4	Selektive Darstellung von Komponenten.....	43
3.5.5	Weitere Features	44
3.6	Schnittstellen zu Applikationen und Dateien.....	45
3.6.1	Import- und Exportformate.....	45
3.6.2	Hyperlinks	46
3.6.3	Externe Bedienung von «Geo3D».....	47
Kapitel 4	«Geo3D» im Visualisierungsprozess	49
4.1	Einsatz und Grenzen von «Geo3D».....	49
4.2	Zur Visualisierung von Geländemodellen.....	51
4.3	«Toporobot» und «Geo3D».....	53
Kapitel 5	Schlussfolgerungen und Ausblick	55
5.1	Fazit.....	55
5.2	Ausblick	57
	Quellen	59
	Literatur	59
	Internet	62
	Anhang	65
A	Inhalt der CD-ROM	65
B	Systemanforderungen von Geo3D.....	65
C	Verwendete Software.....	66
D	Proprietäre Geo3D-Dateiformate	66

Abbildungen und Tabellen

Umschlagillustration

Matterhorn: Rendering in «Geo3D» mit Plug-in-Renderer «FineArt3D»,
Nachbearbeitung in «Photoshop» (Datenquelle M. Heller)

1 Ausgangslage und Fragestellung

Abb. 1-1	«WorldView» – ein VRLM-Plug-in für Web-Browser (aus Zedi 1998)	3
----------	--	---

2 Grundlagen und Voraussetzungen

Abb. 2-1	Schematisches Modell des Rendering (Foley/van Dam/Feiner/Hughes/ Phillipps 1994, S. 225)	11
Abb. 2-2	Perspektivische Kamera mit Kamerakoordinatensystem und Begrenzung des Bildraums.	13
Abb. 2-3	Beleuchtungsmodelle (Datenquelle M. Heller)	15
Abb. 2-4	Phong-Beleuchtung mit variierenden Spiegelungsexponenten und -koeffizienten (Apple Technical Publications 1997a, S. 920)	16
Abb. 2-5	QuickDraw 3D Architektur (nach Fernicola/Thomson 1995, S. 11)	19
Abb. 2-6	QD3D-Klassenhierarchie (nach QuickDraw-3D-API)	20
Abb. 2-7	3DMF-Datei mit Trimesh und Textur	23

3 «Geo3D»-Konzept

Abb. 3-1	User-Interface unter Mac OS 8.6 am Beispiel Geo3D	26
Abb. 3-2	Rolleffekt (Datenquelle M. Heller)	31
Tab. 3-1	Veränderung der Kameraparameter nach Bewegungsvarianten	33
Abb. 3-3	Navigation mit der Maus (Datenquelle M. Sluka)	37
Abb. 3-4	Einstellung von Beleuchtungsparametern in Geo3D.	38
Abb. 3-5	Semitransparentes Geländemodell mit darunterliegendem Höhlensystem (Datenquelle M. Sluka)	39
Abb. 3-6	Semitransparente horizontale Ebene (Datenquelle M. Heller)	41
Abb. 3-7	Gekoppelte Seitenansichten in Geo3D (Datenquelle M. Heller)	43
Abb. 3-8	Orientierungshilfen (Datenquelle M. Sluka)	45

4 «Geo3D» im Visualisierungsprozess

Abb. 4-1	Einfluss der Lichtrichtung auf die Wahrnehmung der Geländeform (Datenquelle M. Heller)	52
Abb. 4-2	Steuerung der Geo3D-Kamera auf dem 2-D-Plan von LimeLight (Datenquelle K. Zimmermann)	54

Anhang

Abb. A-1	Beispiel für das Dateiformat «Gruppen»	67
Abb. A-2	Beispiel für das Dateiformat «Animation»	67

I Ausgangslage und Fragestellung

I.1 Ausgangslage

Die Visualisierung von Daten ist seit jeher eine der zentralen Aufgaben des Fachbereichs Geographie. Dazu der Begriff «Wissenschaftliche Visualisierung» (*Visualization in Scientific Computing, Scientific Visualization*) in der prominenten Definition von McCORMICK/DEFANTI/BROWN (1987, S. 3): «Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. (...)» Darauf bezugnehmend ergänzt MACEachREN (1992, S. 101): «Visualization is foremost an act of cognition, a human ability to develop mental representations that allow geographers to identify patterns and to create or impose order. The mental representations formed and the patterns people see are closely linked to expectations they bring to a given situation.» Visualisierung soll also nicht nur das Symbolische ins Ikonische¹ überführen (kurz: das Modell darstellen), sondern durch Apperzeption² erweitert werden.

Im Fachbereichs Geographie wird oft mit dreidimensionalen Daten gearbeitet. 3-D-Daten repräsentieren meist reale oder simulierte Ausschnitte der Erdoberfläche, der Atmosphäre oder der Lithosphäre. Die Zielsetzungen der Visualisierung von 3-D-Daten sind einerseits Interpretation und visuelle Analyse, andererseits Kommunikation und Präsentation von Resultaten. Ob nun Daten aus geologischen Erhebungen visuell analysiert werden, die Qualität eines Geländemodells visuell interpretiert wird oder die Schichtung von Luftmassen präsentiert werden, das Ziel ist immer eine Darstellung, die der Dreidimensionalität der Daten gerecht wird. Physische Modelle sind, da sie

1. Der Begriff «geometric» von McCormick/DeFanti/Brown ist in diesem Zusammenhang sehr einschränkend und missverständlich. Er wurde deshalb durch «ikonisch» ersetzt.

2. Begrifflich urteilendes Erfassen im Unterschied zur sinnlichen Wahrnehmung (Perzeption).

dreidimensional sind, ideale Darstellungen von 3-D-Daten. Sie sind jedoch mit (zu) hohem Aufwand verbunden. Andere Formen der Visualisierung leiden unter der Reduktion von drei auf zwei Dimensionen. Dies betrifft jegliche Darstellung auf Papier und Bildschirm. Die Reduktion auf zwei Dimensionen führt meist zu Überlagerungen. Statische Darstellungen können nur einen Standort, animierte Sequenzen nur eine vorgegebene Zahl von Ansichten berücksichtigen. Interaktive Visualisierung von 3-D-Daten bietet durch die frei Wahl des Standorts und der Blickrichtung einen Ausweg, der den Qualitäten physischer Modelle sehr nahe kommt. Kap. 2.1, S. 7 gibt einen Überblick über die wichtigsten Methoden der Visualisierung von 3-D-Daten.

Da die Neuberechnung einer 3-D-Darstellung rechnerisch aufwändig ist, war die interaktive Visualisierung komplexerer Modelle bis vor kurzer Zeit nur auf leistungsfähigen Workstations möglich (z.B. SUTER 1997). Die Situation im Hardwarebereich hat sich entscheidend geändert. Neuere Computer der untern Preisklasse (im Folgenden Lowend-Computer genannt) bieten innerhalb gewisser Grenzen genügend Leistung für die interaktive 3-D-Visualisierung. Die interaktive Manipulation von 3-D-Ansichten ist letztlich bei gegebener erwarteter Reaktionszeit eine Funktion von Hardwareleistung, Modellgrösse und Algorithmen (Rechenaufwand). Zu Beginn meiner Beschäftigung mit 3-D-Visualisierung im Jahre 1997 waren bereits entsprechende 3-D-Funktionsbibliotheken zur Programmierung von 3-D-Anwendungen für verschiedene Betriebssysteme erhältlich³. Dadurch konnte und kann sich ein weit grösserer Personenkreis mit diesem Thema befassen. Spezielle 3-D-Hardware, die 3-D-Anwendungen erheblich beschleunigt, war vor zwei Jahren noch wenig verbreitet. Diese Tatsache behinderte die Etablierung von 3-D-Anwendungen. Inzwischen jedoch gehört 3-D-Hardware zur standardmässigen Ausrüstung neuer Lowend-Computer.

Wir haben nun Visualisierungsziele, welche die interaktive Visualisierung von 3-D-Daten zumindest attraktiv erscheinen lassen («seeing the unseen» nicht nur im übertragenen, sondern auch im wörtlichen Sinn). Weiter erlauben heutige Lowend-Computer einer breiten Anwenderschaft, interaktive Visualisierung auch tatsächlich anzuwenden. Um aber interaktive 3-D-Visualisierung zu realisieren, brauchen wir nebst den Modellen für diesen Zweck geeignete Applikationen (diese Applikationen werden im Folgenden «Viewer» genannt). Folgende Anforderungen sollte ein Viewer zumindest erfüllen:

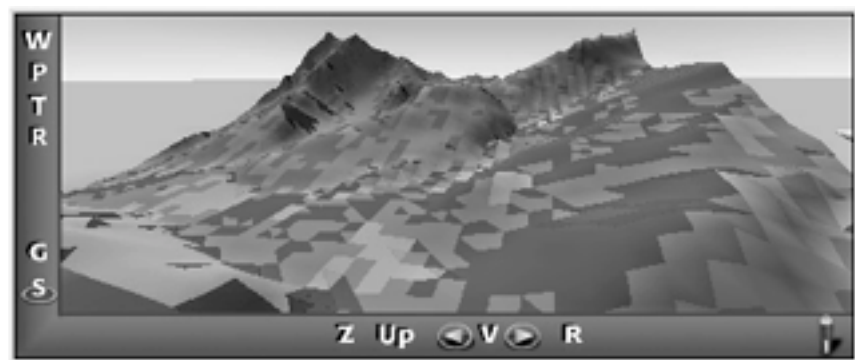
- Fließende Bewegung durch den virtuellen Raum muss gewährleistet sein.
- Beliebige Ansichten des Modells sollen erreicht werden können.
- Vielfältige, «intuitive» Navigationsmöglichkeiten müssen angeboten werden.

3. Vergleiche zwischen den damals wichtigsten 3-D-Funktionsbibliotheken OpenGL (Silicon Graphics), QuickDraw 3D (Apple) und Direct3D (Microsoft) finden sich in *Thompson 1996* und *Flohr 1996*.

- Die Leistungsstärke heutiger Hardware soll voll ausgenutzt werden: Die Darstellungsqualität soll das zur Zeit mögliche Niveau erreichen und im Wesentlichen interaktiv manipulierbar sein.
- Ein dokumentiertes Austauschformat für 3-D-Daten muss unterstützt werden.
- Die Applikation soll möglichst schnell sein; im Computer vorhandene spezielle 3-D-Hardware muss von der Applikation unterstützt werden.
- Es soll auf die 3-D-Koordinaten zugegriffen werden können und weitere Informationen sollen an Objekte angefügt werden können.

Internet-Recherchen ergaben, dass das einzige Visualisierungssystem, das den obigen Anforderungen in etwa genügt, die «Virtual Reality Modeling Language» (VRML⁴) mit den entsprechenden Viewern ist. VRML als Beschreibungssprache für 3-D-Daten ist mächtig und hat den Vorteil, dass sie als offener Standard definiert ist; die VRML-Viewer sind fast ausschliesslich in Form von Plug-ins (Programmerweiterungen) für bestehende Web-Browser (wie *Netscape Navigator*, *Internet Explorer*) erhältlich (Abb. I-1). Da das User-

Abb. I-1
«WorldView» – ein
VRML-Plug-in für Web-
Browser (aus Zedi 1998)



Interface eines Plug-ins ins Fenster des Browsers eingebettet wird, sind die Möglichkeiten der Interaktion zwangsläufig eingeschränkt. Die Laufzeitstabilität der Viewer ist oft unbefriedigend. Die Viewer bieten verschiedene, aber klar limitierte Navigations-Möglichkeiten im Raum und ein beschränktes Set von zusätzlichen Einstellungen, welche die Qualität der Darstellung betreffen. Zusätzliche Informationen können an Objekte angefügt werden. Der Zugriff auf die 3-D-Koordinaten ist hingegen nicht möglich. Da VRML eine reine Beschreibungssprache ist, basiert die Implementation der Viewer auf verschiedenen 3-D-Funktionsbibliotheken: z.B. *CosmoPlayer* auf *OpenGL*, *WorldView* auf *QuickDraw 3D*. Die 3-D-Hardwareunterstützung basiert auf

4. VRML wurde in erster Linie als Standardbeschreibungssprache für 3-D-Modelle für das World Wide Web entwickelt. Als Einführung in die VRML-Version 2.0 (VRML97) ist Ames/Nadeau/Moreland 1997 zu empfehlen. Weitere Informationen über VRML und deren Nachfolgerin Extensible 3D (X3D) sind u.a. unter <http://www.vrml.ch>, <http://www.vrml.org> und <http://www.web3d.org> zu finden. VRML 1.0 ist gänzlich verschieden von VRML97 und hat inzwischen praktisch keine Bedeutung mehr.

den verwendeten Bibliotheken und ist meist gewährleistet. Die verschiedenen VRML-Viewer unterstützen aber nur einen gewissen Teil der VRML-Spezifikationen. ZEDI 1998 hat den Einsatz von VRML in der Geographie in seiner Diplomarbeit eingehend untersucht und beschreibt auch ausführlich die Interaktivität von VRML-Szenen.

3DMF (3D Metafile Format) ist eine weitere 3-D-Beschreibungssprache. Sie ist integraler Bestandteil von Apples QuickDraw-3D-Architektur (QD3D). 3DMF ist weniger umfangreich als VRML (vor allem in Bezug auf die Einbindung von Animation und weiteren Medien), dafür kompakter und wird von QD3D-Applikationen, weil Bestandteil von QD3D, garantiert vollständig unterstützt. Kein QD3D-Viewer erreichte aber den Funktionsumfang der VRML-Viewer.

Es zeigte sich auch, dass der Einsatz von 3-D auf Lowend-Computern sehr zögernd erfolgt. Zwar bieten viele kommerzielle 3-D-Graphik-Programme Exportfunktionen für VRML und 3DMF, meist aber in eher fragwürdiger Qualität. Typische Mängel sind: unvollständige oder fehlerhafte Implementation der Spezifikationen und nicht optimierter («langsamer») Code. Wichtiger für die wissenschaftliche Visualisierung ist die grosse Aktivität im Fachbereich Chemie. Hier wurden die neuen Möglichkeiten sehr rasch in Beschlag genommen. Ähnliches gilt in eingeschränktem Mass für den Fachbereich Architektur. Im geographischen Umfeld hingegen ist der Einsatz von Lowend-Computern noch kaum erfolgt. Dies mag mit den eher grossen Datenmengen zusammenhängen. Eine für diese Arbeit zentrale Ausnahme ist Martin Hellers *LimeLight*, ein Höhlenkartographieprogramm, das optimierte 3-D-Daten exportieren kann.

1.2 Zielsetzung

Die zur Zeit vorliegenden Viewer für 3-D-Daten erfüllen die geforderte minimale Funktionalität nur teilweise. Wie soll nun ein erweitertes Visualisierungssystem für Lowend-Computer aussehen? Die Beantwortung dieser Frage führt zu den Zielsetzungen dieser Arbeit.

Die Möglichkeiten interaktiver 3-D-Visualisierung auf Lowend-Computern sollen erweitert werden, um damit einen Beitrag zur Verbreitung von 3-D-Anwendungen im geographischen Umfeld zu leisten. Es wird ein Konzept eines Viewers vorgestellt, das von den Erfahrungen mit verfügbaren 3-D-Daten (Höhlensysteme, Digitale Geländemodelle) ausgeht und die bestehenden Lösungen wie auch die beschränkte Leistung der Hardware berücksichtigt. Der Schwerpunkt liegt dabei im Bereich der Navigation im virtuellen Raum. Da erfolgreiche Visualisierung nicht zuletzt von einer guten Darstellung abhängt, soll die Editierbarkeit aller darstellungsrelevanten Parameter garantiert werden. Weiter sind zusätzliche Mittel für die visuelle Analyse der Daten zu prüfen. Wesentliche Teile dieses Konzepts werden implementiert (**Geo3D**) und veröffentlicht. Durch die Freigabe des Programms über das

Internet erhoffe ich mir Anregungen und Kommentare eines grösseren Personenkreises sowie Hinweise auf Fehler des Programms. Die Veröffentlichung ist als Beitrag zur Verbreitung von 3-D-Anwendungen zu verstehen. Nicht Bestandteil dieser Arbeit ist hingegen das Herstellen (*Modeling*) von 3-D-Modellen.

Zusammenhängend mit der Zielsetzung ergeben sich folgende Fragen: Welche Anforderungen sollte ein Viewer für geographische Daten erfüllen? Wie können diese im Hinblick auf ein möglichst intuitiv zu bedienendes *User-Interface* strukturiert und umgesetzt werden? Für welche Zwecke kann ein Programm wie **Geo3D** eingesetzt werden? Und wo liegen die derzeitigen Grenzen? Die Beantwortung dieser Fragen wird subjektiv ausfallen und ist als Entwurf zu betrachten, der auch Grundlagen für zukünftige Arbeiten liefern soll.

Als Plattform für die Implementation von **Geo3D** dienen Power-Macintosh-Computer unter dem Betriebssystem Mac OS. Die 3-D-Programmierung basiert auf QuickDraw 3D (siehe Kap. 2.3, S. 18). Die Wahl erfolgt aus pragmatischen Gründen: Soft- und Hardware sind verfügbar und in hohem Mass geeignet für die Ziele dieser Arbeit. Das Programm **Geo3D** wird integraler Bestandteil dieser Arbeit sein. Es soll stabil implementiert sein, auf allen Power-Macintosh-Computern⁵ und unter den Betriebssystemversionen von Mac OS der letzten Jahre laufen. Es soll frei von bekannten Fehlern sein. Die Fragestellung dieser Arbeit wird um folgende technischen Aspekt erweitert: Wie effizient und stabil ist die Programmierung mit QuickDraw 3D?

I.3 Sprachliche Regelung der verwendeten Begriffe

Die Fachsprache der Computerliteratur ist Englisch. Die meisten Publikationen werden in dieser Sprache veröffentlicht. Deutschsprachige Publikationen und Übersetzungen ins Deutsche sind vergleichsweise selten. Meist setzen sich die englischsprachigen Begriffe im Deutschen durch; *Computer*, *Software*, *Hardware*, *File* sind populäre Beispiele. Wenn überhaupt, werden Begriffe nachträglich eingedeutscht und neben den englischen verwendet. Von den erwähnten vier Begriffen wird einzig das Wort «File» im Fremdwörterduden (DUDEN 1997) übersetzt: als Datei. In dieser Arbeit werden, wenn die Gefahr von Missverständnissen besteht, die originalen Begriffe bevorzugt. Alternativ wird bei der erstmaligen Verwendung eines eingedeutschten Begriffes der englische in Klammern hinzugefügt.

5. Computer von Apple (<http://www.apple.com>), die einen PowerPC-Prozessor haben. Alle in den letzten fünf Jahren produzierten «Macs» gehören in diese Kategorie.

1.4 Verwendete Daten

Die in dieser Arbeit abgebildeten Modelle stammen aus folgenden Quellen:

- Geländemodell «Matterhorn». Datenquelle: M. Heller. Die ursprünglichen Daten wurden mittels ATM-Filtering (Adaptive triangular mesh filtering) reduziert unter Verwendung der Applikation *Bermuda* (HELLER 1990).
- Höhlenmodell «Erdmann». Datenquelle K. Zimmermann. Für die Datenmodellierung wurde die Applikation *LimeLight* verwendet (siehe <http://www.geo.unizh.ch/~heller/toporobot/> und HELLER 1994).
- Höhlen- und Geländemodell «Cachtice». Datenquelle M. Sluka. Das Höhlenmodell wurde ebenfalls mit *Limelight* modelliert. Das Geländemodell (ein GRID) wurde mit der Applikation *GeoCon* (<http://www.data-comm.ch/luzern/>) konvertiert.

Die Darstellungen sind **Geo3D** entnommen und wurden mit einer Ausnahme (Umschlagillustration) nur geringfügig mit *Photoshop* für den Druck nachbearbeitet.

1.5 Aufbau der Arbeit

Kapitel 2 (S. 7) zeigt Vor- und Nachteile verschiedener Methoden der Visualisierung von 3-D-Daten auf und beschäftigt sich anschliessend mit ausgewählten Grundlagen der 3-D-Computergraphik, im speziellen mit der 3-D-Funktionsbibliothek QuickDraw 3D. Kapitel 3 (S. 25) – der Kern dieser Arbeit – beschreibt das Konzept einer Applikation für interaktive 3-D-Visualisierung von geographischen (und andern) Daten. **Geo3D** ist die Realisierung eines wesentlichen Teils der hier beschriebenen Anforderungen. Kapitel 4 (S. 49) zeigt die Möglichkeiten und Grenzen von **Geo3D** im Visualisierungsprozess unter besonderer Berücksichtigung der Kommunikation zwischen Applikationen. Kapitel 5 (S. 55) fasst die Resultate zusammen und wirft einen Blick in die Zukunft. Daran anschliessend werden die verwendeten Quellen dokumentiert. Nach den Literaturangaben werden alle im Text erwähnten Internet-Adressen unter dem betreffenden Stichwort aufgelistet. **Geo3D** samt Manual und weitere Dokumente sind auf der beiliegenden CD-ROM zu finden.

2 Grundlagen und Voraussetzungen

2.1 Methoden der Visualisierung von 3-D-Daten

Dreidimensionale Daten (3-D-Daten) sind im Fachbereich Geographie von grosser Bedeutung. Beispiele sind Geländemodelle, Atmosphärenmodelle, Modelle von geologischen Strukturen, Höhlenmodelle etc. Es gibt verschiedene Ansätze, 3-D-Modelle zu visualisieren. Herkömmliche Ansätze sind physische Modelle und Darstellungen auf Papier, neueren Datums ist die Computervisualisierung auf dem Bildschirm. Die nachfolgenden Erläuterungen helfen, interaktive 3-D-Visualisierung (das Ziel dieser Arbeit) vor dem Hintergrund anderer, vertrauterer Formen der Visualisierung von 3-D-Modellen zu sehen – ohne Vollständigkeit zu beanspruchen.

2.1.1 Physische Modelle

Bekannte physische Modelle aus der Geographie sind das Relief und der Globus. Aus der Anatomie bekannt ist das zerlegbare Modell vom Menschen. Die Vorteile eines physischen Modells sind offensichtlich: Der Betrachter kann das Modell von allen Seiten einsehen, kann es überblicken oder Details anschauen. Der Einsatz von transparenten Materialien und Zerlegbarkeit ermöglichen die Darstellung von sehr komplexen dreidimensionalen Strukturen. Das physische Modell ist an und für sich statisch, doch der Betrachter kann sich frei bewegen und so den betrachteten Ausschnitt bestimmen. Wie wir später sehen werden, entspricht dies einer interaktiven Ausprägung des Begriffs «Bewegungsdynamik» aus der Computergraphik (Kap. 2.2.5). Eine weitere Eigenschaft physischer Modelle ist, dass sie echte 3-D-Darstellungen sind – die drei Dimensionen bleiben erhalten. Physische Modelle sind aber sehr aufwändig in der Herstellung und benötigen unter Umständen viel Platz. Sie eignen sich deshalb vor allem für die Präsentation von Ergebnissen.

2.1.2 Visualisierung auf Papier

Es gibt unzählige Beispiele von Visualisierungen von 3-D-Modellen auf Papier: Topographische Karten, Panoramen, Blockdiagramme, auch die in dieser Arbeit mehrfach abgebildeten Computermodelle von Gelände und Höhlen. Papier ist ein statisches Medium. Der/die Betrachter/in statischer Darstellungen hat keinen Einfluss auf die dargestellte Ansicht. Mehrere Ansichten bedingen mehrere Abbildungen. Wenn eine oder wenige Abbildungen genügen, um einen Sachverhalt zu vermitteln, ist Papier ein ideales Medium – platzsparend und ohne technische Hilfsmittel lesbar.

Papier hat nur zwei Dimensionen. 3-D-Modelle werden auf zwei Dimensionen reduziert. Projektionen können orthogonal oder perspektivisch sein. Die Projektion von 3-D-Daten auf eine Ebene führt im Allgemeinen zu Überdeckungen. Ein Spezialfall sind hier sogenannte 2.5-D-Daten. Typischer Vertreter dieses Typs sind Geländemodelle. 2.5-D zeichnet sich dadurch aus, dass für einen gegebenen 2-D-Punkt genau ein Wert in der dritten Dimension existiert. Eine orthogonale Projektion in Richtung der dritten Dimension führt somit nicht zu Überdeckungen. Die topographische Karte ist das wohl bekannteste Beispiel einer solchen Abbildung. Papier ist ein geeignetes Medium, wenn durch geschickte Wahl des Standorts, der Blickrichtung und der Projektion der Sachverhalt dargestellt werden kann.

Um den Verlust der dritten Dimension zu entschärfen, werden verschiedene Methoden der Tiefenillusion eingesetzt: Schräglichtschattierung, Dunst etc. Schattierung betont die Plastizität (z.B. auf topographische Karten), Dunst die Tiefenwirkung (oft in Panoramen angewandt).

2.1.3 Visualisierung auf dem Bildschirm

Statische Darstellungen auf dem Bildschirm haben im Wesentlichen die gleichen Qualitäten wie Darstellungen auf dem Papier. Bildschirme haben aber eine vergleichsweise beschränkte räumliche Auflösung; ein guter Druck ist diesbezüglich dem Bildschirm weit überlegen. Interaktive Wahl des Ausschnitts und Vergrößerung sind dagegen Vorteile, die Papier nicht bieten kann. Doch Vergrößerungen und Verkleinerungen (Zoomen) sind mit Schwierigkeiten bezüglich Generalisierung und Interpolation verbunden.

Animierte Darstellungen sind Darstellungen, welche aus mehreren, chronologisch aneinandergereihten Einzelbildern (Sequenzen) bestehen. Auf diese Weise wird der Eindruck einer Bewegung (Dynamik) vermittelt. Wichtig für diese Arbeit ist die Bewegungsdynamik (Kap. 2.2.5). Der Vorteil animierter Darstellungen ist, dass eine Vielzahl von Ansichten in nachvollziehbarer Abfolge betrachtet werden kann. Die Sequenz kann jedoch vom/von der Betrachter/in nicht beeinflusst werden. Üblich ist bei animierten Darstellungen die Möglichkeit, Einzelbilder anzusehen. Auch dies ist eine beschränkte Form der Interaktion.

Interaktive Darstellungen von 3-D-Modellen erlauben dem/der Betrachter/in, Standort und Blickrichtung (synthetische Kamera) in Echtzeit zu verändern (interaktive Bewegungsdynamik). Es entsteht der Eindruck fließender Bewegung durch das 3-D-Modell. Die Darstellung ist zwar zweidimensional, kommt aber den Qualitäten physischer Modelle sehr nahe, ohne deren Nachteile zu teilen. Die Modelle müssen aber einige Voraussetzungen für eine erfolgreiche interaktive 3-D-Visualisierung erfüllen:

- Komponenten und Strukturen müssen potentiell sichtbar sein.

Während 2.5-D-Modelle diese Forderung problemlos erfüllen, können in echten 3-D-Modellen Komponenten von andern vollständig verhüllt werden. Dies kann an einem Beispiel aus der Geologie verdeutlicht werden. Mehrere eng übereinanderliegende Schichten sind zu visualisieren. Die Darstellung der Schichten als geschlossene Körper verunmöglicht es, etwas zu erkennen. Die Reduktion auf Schichtgrenzen bietet einen Ausweg. Sind die Zwischenräume zu klein, müssen weitere Mittel eingesetzt werden. Der Einsatz von Transparenz und/oder selektive Darstellung der einzelnen Schichtgrenzen (Kap. 3.5.4) bietet hier einen Ausweg. Potentielle Sichtbarkeit genügt in vielen Fällen nicht. Zusammengehörende Komponenten müssen oft identifizierbar sein. Dies kann beispielsweise über themenbezogene Einfärbung der Komponenten erreicht werden.

- Die Leistung der Hardware muss als limitierender Faktor berücksichtigt werden.

Die maximale Grösse der Modelle, die noch interaktiv manipuliert werden können, wird durch die vorhandene Leistung der Hardware bestimmt. Welche Bildraten (Kap. 2.2.5) noch interaktives Arbeiten ermöglichen, ist in einem gewissen Rahmen von den subjektiven Bedürfnissen abhängig. Im Bereich der Datenmodellierung sind, wenn die Modelle zu gross sind, die Zerlegung in Teilmodelle, Optimierung der Datenstruktur (Kap. 2.3.2) und Generalisierungsmethoden mögliche Lösungen (für Geländemodelle siehe Kap. 4.2).

- Kognitive Schwierigkeiten müssen berücksichtigt werden.

Nicht zu unterschätzen sind die kognitiven Schwierigkeiten bei der interaktiven Visualisierung räumlicher Daten. Im Gegensatz zur 2-D-Kartographie gibt es keine verbindlichen Regeln, wie mit den unzähligen Darstellungsmöglichkeiten umgegangen werden soll. Solange die gleiche Person die Modelle generiert und visualisiert, kann das Problem individuell gelöst werden. Sobald verschiedene Personen beteiligt sind, muss dieses Problem diskutiert werden.

Das Visualisierungssystem muss ebenfalls gewisse Voraussetzungen erfüllen. Diese werden im **Geo3D**-Konzept (Kap. 3) ausführlich diskutiert. So soll hier nur die wichtigste erwähnt werden:

- Das interaktive Navigationssystem muss fließende Bewegung durch den virtuellen Raum gewährleisten und beliebige Ansichten des Modells ermöglichen.

Die Navigation muss einfach und intuitiv erfolgen, muss präzise und stabil sein. Tritt Orientierungsverlust ein, muss die Rückkehr zu bekannten Ansichten angeboten werden.

2.2 Methoden der Computergraphik

2.2.1 Rendering, Imaging und Geometrisches Modell

“Computer graphics concerns the pictorial synthesis of real or imaginary objects from their computer-based models, whereas the related field of image processing ... treats the converse process: the analysis of scenes, or the reconstruction of models of 2D or 3D objects from their pictures (FOLEY/VAN DAM/FEINER/HUGHES 1996, S 2).” Die Computergraphik stellt also Methoden zur Verfügung, ein computerbasiertes «Geometrisches Modell» in ein Bild zu verwandeln. In unserem Fall wird aus einem im Arbeitsspeicher vorliegenden dreidimensionalen Modell, bestehend aus offenen oder geschlossenen 3-D-Oberflächen samt Attributen, eine zweidimensionale Darstellung auf dem Bildschirm berechnet. Dieser Prozess wird oft (auch in dieser Arbeit) als *Rendering* bezeichnet. *Imaging* ist im Gegensatz dazu der Prozess, aus einer Darstellung das dahinter liegende Modell zu rekonstruieren oder zu analysieren. Die Qualität der im konkreten Fall angewendeten Methoden des Rendering können über den Umkehrprozess des Imaging beurteilt werden. Oder anders ausgedrückt: Wie hilfreich sind die jeweiligen Renderingmethoden für die Analyse eines geometrischen Modells? Der gewünschten Qualität des Renderings und Zahl der im geometrischen Modell enthaltenen Primitiven steht die Forderung der Interaktivität nach kurzer Reaktionszeit⁶ gegenüber. Interaktives Rendering auf vergleichsweise langsamen Computern – Ziel dieser Arbeit – ist zwangsläufig mit Zeitnot konfrontiert. Anders gesagt: Je grösser das Modell und je aufwändiger das Renderingverfahren, desto länger wird die Reaktionszeit sein.

Der Begriff «Geometrisches Modell» (*geometric model*) wird nach FOLEY/VAN DAM/FEINER/HUGHES 1996, S 288 folgendermassen definiert: “Geometric or graphical models describe components with inherent geometrical properties ... Among the ingredients a geometric model may represent are the following:

6. Die Reaktionszeit ist die Zeit, die zwischen der Eingabe eines Befehls durch ein Eingabegerät (Maus, Tastatur) und dem, meist visuellen, Feedback des Computers verstreicht. Das Feedback bei interaktiver 3-D-Visualisierung ist in der Regel das neu gerenderte Bild.

- Spatial layout and shape of components (i.e. the geometry of the entity), and other attributes affecting the appearance of components, such as color
- Connectivity of components (i.e. the structure or topology of the entity); (...).
- Application-specific data values and properties associated with components ...”

Der Begriff «Geometrisches Modell» wird in dieser Definition sehr weit gefasst und in diesem Sinn auch für diese Arbeit verwendet. Der Begriff «Komponente» (*component*) wird ebenfalls im obigen Sinn verwendet.

2.2.2 Rendering-Methoden

Das dem Rendering zugrunde liegende schematische Modell ist in Abb. 2-1 dargestellt. Das Ziel des schematischen Modells, die Abbildung eines digitalen dreidimensionalen Modells auf ein zweidimensionales Ausgabemedium, kann durch verschiedene Strategien erreicht werden (Diese können von den Strategien des schematischen Modells abweichen). Im Folgenden werden die elementaren Schritte mit den zugehörigen Koordinatentransformationen beschrieben. Anschliessend werden weitere Teilschritte vorgestellt und in eigenen Teilkapiteln verschiedene Beleuchtungsmodelle und ein Kameramodell besprochen.

Die Geometrie der Komponenten wird oft in einem lokalen Koordinatensystem beschrieben. Die Positionierung aber erfolgt in einem sogenannten Weltkoordinatensystem. Entsprechend müssen allfällige lokale Koordinatensysteme transformiert werden. Die Komponenten im Weltkoordinatensystem werden durch Kappen (*Clipping*) des Bildraums (Kap. 2.2.3) beschnitten. Der Bildraum wird durch die Kameraparameter bestimmt und erlaubt eine Reduktion der weiterzuverarbeitenden Datenmenge auf die potentiell sichtbaren Komponenten. Die Komponenten im Bildraum müssen anschliessend auf die Projektionsebene (*View Plane*) projiziert werden. Nun wird der Inhalt des endlichen View-Plane-Fensters (der uns interessierende Ausschnitt der View Plane) auf Bildschirmkoordinaten transformiert. Die Darstellung muss

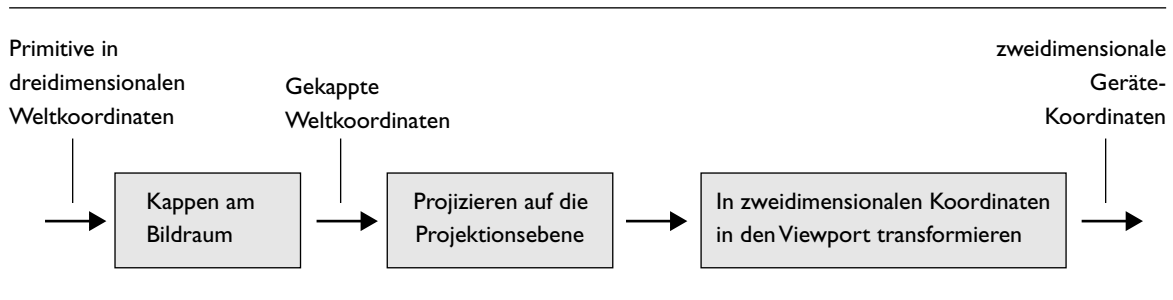


Abb. 2-1 Schematisches Modell des Rendering (Foley/van Dam/Feiner/Hughes/Phillipps 1994, S. 225)

schliesslich der Pixel-Auflösung des Ausgabemediums entsprechend gerastert werden.

Vor dem Projizieren auf die Projektionsebene werden in der Regel Zwischenschritte eingeschoben, die den Rechenaufwand und die Darstellung optimieren. Die der Kamera zugewandten Flächen werden ermittelt, die abgewandten Flächen werden eliminiert (*Backface Culling*). Tiefensortierverfahren ermitteln und entfernen geometrische Komponenten, die durch näher bei der Kamera liegende, opake Komponenten verdeckt werden. Der Rechenaufwand wird durch Backface Culling und Tiefensortierverfahren beträchtlich vermindert. Sind einmal die darzustellenden Komponenten ermittelt, wird – falls erwünscht – die für räumliches Wahrnehmen wichtige Beleuchtung (bez. Schattierung) der Oberflächen erzeugt. Umgebungsparameter wie Nebel oder Dunst können hier ebenfalls einfließen. Schattierungsberechnungen sind sehr aufwändig. Für interaktive Anwendungen werden daher sehr vereinfachte Modelle verwendet.

Die folgenden zwei Kapitel dienen der vertieften Betrachtung von Kamera-, Beleuchtungs- und Interpolationsmodellen. Dabei werden hauptsächlich die von der 3-D-Funktionsbibliothek QuickDraw-3D (die von **Geo3D** schliesslich verwendet wird) unterstützten Modelle und Lösungen besprochen.

2.2.3 Kamera

Als zweckmässige Metapher wird bei der Erzeugung dreidimensionaler Szenen der Begriff «synthetische Kamera» verwendet. Wie eine Fotokamera kann die synthetische Kamera an eine beliebige Stelle verschoben und in jede Richtung gedreht werden. Den Verschluss drücken bedeutet übertragen eine begrenzte zweidimensionale Darstellung eines dreidimensionalen Modells zu erzeugen. Für animierte Sequenzen ist die Vorstellung einer Filmkamera hilfreich.

Um nun die *View Plane* (Bildebene) für das Rendering zu bestimmen, muss die synthetische Kamera genauer definiert werden. Die Kameraparameter sind abhängig von der gewünschten Projektion. Da das Bild schliesslich auf den (meist annäherungsweise) ebenen Bildschirm projiziert wird, sind nicht-planare Projektionen nicht von weiterem Interesse. Planare geometrische Projektionen können in Parallel-Projektionen und perspektivische Projektionen unterteilt werden. Im ersten Fall ist die Distanz zwischen Projektionszentrum und Projektionsebene unendlich, im zweiten Fall ist sie endlich.

Die Grundlagen der folgenden Ausführungen basieren im Wesentlichen auf dem Modell von FOLEY/VAN DAM/FEINER/HUGHES (1996, S. 237f) unter Berücksichtigung der von QuickDraw 3D verfolgten Strategie. Dieses Modell einer perspektivischen Kamera reduziert durch gewisse Konventionen die Anzahl der Parameter, die notwendig sind, um eine Kamera zu definieren. Die in den nachstehenden Ausführungen verwendeten Begriffe und deren räumliche Anordnung sind in Abb. 2-2 illustriert.

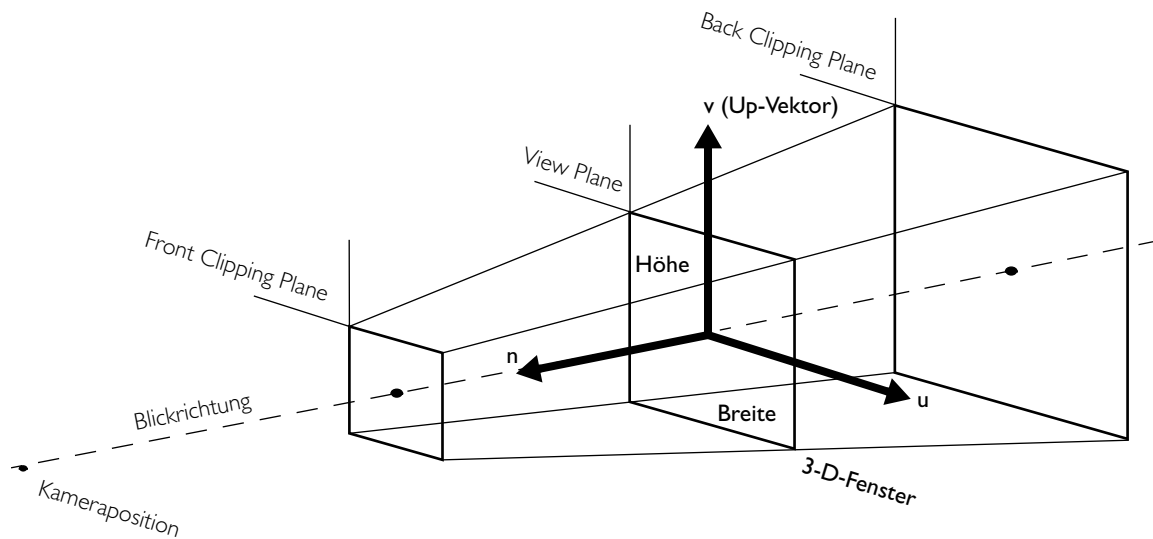


Abb. 2-2 Perspektivische Kamera mit Kamerakoordinatensystem und Begrenzung des Bildraums (Erklärung siehe Text)

Eine perspektivische Kamera besteht grundsätzlich aus der Kameraposition (in diesem Modell zugleich das Projektionszentrum) und der View Plane (Projektionsebene). In der View Plane wird ein rechteckiges Fenster (*Window*) durch die Parameter Breite und Höhe definiert. Das Fenster – im folgenden als 3-D-Fenster bezeichnet – hat die gleiche Funktion wie ein 2-D-Graphik-Fenster: Nur die Teile des geometrischen Modells, die auf das 3-D-Fenster projiziert werden, werden dargestellt. Die Projektion auf die restliche View Plane wird ignoriert. Unter der Annahme, dass die Kameraposition zugleich das Projektionszentrum ist, können wir eine halboffene Pyramide aufspannen, den Bildraum (*View Volume*). Der Bildraum wird durch die Begrenzungsebenen *Front Clipping Plane* (auch *Hither Plane* genannt) und *Back Clipping Plane* (auch *Yon Plane* genannt) zu einer stumpfen Pyramide. Nur die Teile des geometrischen Modells, die in diesem Pyramidenstumpf liegen, sind für die Darstellung relevant; die ausserhalb des Bildraums liegenden Komponenten werden durch Clipping eliminiert.

Bis anhin haben wir den Bildraum von der Kameraposition isoliert betrachtet. Wenn wir nun festlegen, dass die *View-Plane*-Normale, die durch den Mittelpunkt des 3-D-Fensters führt, auf die Kameraposition zeigt, müssen wir, um die Beziehung zwischen View-Plane und Kameraposition zu definieren, noch drei Parameter festlegen: Erstens die räumliche Lage der View-Plane-Normale, die zugleich, von der Kameraposition her betrachtet, die umgekehrte Blickrichtung ist. Die Blickrichtung wird in der Praxis durch einen Punkt auf der View-Plane-Normale (meist als *Center of Attention* oder *Point of Interest* bezeichnet) festgelegt. Zweitens die Distanz zwischen Kameraposition und View Plane (und analog zu den *Clipping Planes*) und drittens den sogenannten Up-Vektor (*up Vector*), der festlegt, wie das 3-D-Fenster in

der View Plane orientiert wird. Der Up-Vektor zeigt definitionsgemäss nach oben. Da die Distanz zwischen Kameraposition und 3-D-Fenster in diesem Modell den Öffnungswinkel implizit festlegt, kann der Distanzparameter durch den Öffnungswinkel ersetzt werden. Wird die View-Plane-Normale durch einen anderen Punkt als den Mittelpunkt des 3-D-Fensters der View Plane (*Off-Axis Viewing*) gelegt, muss dieser explizit festgelegt werden.

Es ist für die Praxis nicht hilfreich, die Kameraposition ins Unendliche zu verlegen, um eine orthographische Kamera zu modellieren. Hingegen können wir die Kameraposition zur Kameraebene mit entsprechendem 3-D-Fenster erweitern und einen quaderförmigen Bildraum aufspannen. Der Vorteil von orthographischen Kameras besteht darin, dass Parallelen des geometrischen Modells auch auf der View Plane parallel abgebildet werden. Perspektivische Kameras entsprechen dagegen unserer Wahrnehmung des Raums und beruhen im Wesentlichen auf den Abbildungseigenschaften einer konventionellen Kamera.

In der Regel wird für Kameras ein lokales rechtshändiges⁷ kartesisches Koordinatensystem verwendet (Abb. 2-2). Die View-Plane-Normale in negativer Blickrichtung ergibt die n-Achse (oder z-Achse) und zeigt gegen die Kameraposition. Der Up-Vektor entspricht der positiven v-Achse (oder y-Achse). Die u-Achse (oder x-Achse) ergibt sich nun durch Bildung eines Rechts-Systems. Von der Kameraposition aus gesehen zeigt die positive u-Achse gegen rechts. Für positive Rotationen in einem Rechtssystem gilt die folgende Konvention: Von der positiven u-Achse gegen den Ursprung gesehen überführt eine Rotation um 90° gegen den Uhrzeigersinn die positive v-Achse in die positive n-Achse. Entsprechend führt die Rotation um die v-Achse die n-Achse in die u-Achse und die Rotation um die n-Achse die u-Achse in die v-Achse.

2.2.4 Beleuchtung und Interpolation

Mit Ausnahme von Drahtgitterdarstellungen und andern speziellen Renderingverfahren beruht der dreidimensionale Eindruck zweidimensionaler Bilder vor allem auf einer geschickten Beleuchtung und Schattierung des geometrischen Modells (Abb. 2-3). Eine monochrome Komponente ist ohne gerichtete Beleuchtung nur über die Silhouette wahrnehmbar. Beleuchtungsmodelle versuchen, die komplexen Interaktionen zwischen Licht und Oberflächen nachzubilden. Auch hier ist zwischen dem Zeitaufwand der Berechnung und der Qualität des Resultats abzuwägen.

7. Die Orientierung der Koordinatenachsen eines rechtshändigen Koordinatensystems (Rechtssystem) kann durch eine kleine Fingerübung bestimmt werden: Strecke den Daumen und Zeigefinger der rechten Hand flach aus und den Mittelfinger nach vorn. Wenn der Daumen in Richtung u und der Zeigefinger in Richtung v zeigt, so zeigt der Mittelfinger in Richtung n.



Abb. 2-3 Beleuchtungsmodelle: links ohne, rechts mit Lambert-Beleuchtungsmodell
(Datenquelle M. Heller)

Die Grundlagen der Beleuchtungsmodelle sind Lichtquellen und Materialeigenschaften der Oberflächen. Die hier erwähnten Beleuchtungsmodelle erlauben mehrere Lichtquellen von verschiedener Intensität und Farbe. Ambientes Licht ist eine Idealisierung von unendlich gestreutem Licht ohne Quelle. Paralleles Licht, punktförmiges Licht und Spotlight strahlen gerichtet. Punktförmiges Licht erlaubt eine Variation der Lichtintensität in Abhängigkeit der Distanz. Spotlight entspricht punktförmigem Licht mit eingeschränktem Strahlungswinkel. Materialkonstanten sind nebst diffuser Farbe diverse Koeffizienten, die zusammen mit den jeweiligen Beleuchtungsmodellen erklärt werden.

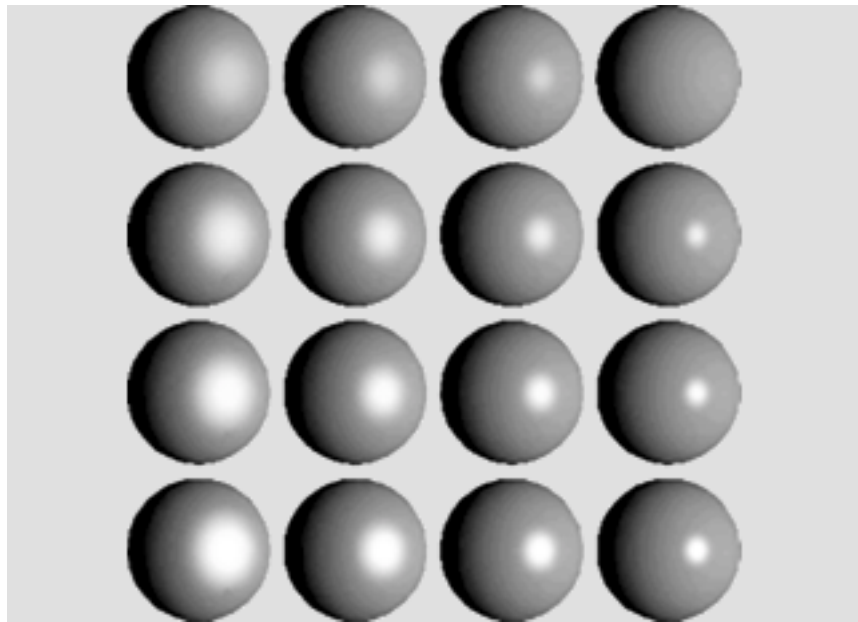
Das Lambert⁸-Beleuchtungsmodell (Abb. 2-3) geht von der Annahme aus, dass Oberflächen nur diffus reflektieren. Für einen bestimmten Punkt auf der Oberfläche gilt folgende Gleichung (APPLE TECHNICAL PUBLICATIONS 1997a, S. 917):

$$I_{\text{Lambert}} = I_a k_a O_d + \sum_i (N \cdot L_i) I_i k_d O_d$$

Der erste Teil der Gleichung bestimmt den ambienten Anteil des Lichts; I_a ist die Intensität des ambienten Lichts, k_a der Ambient-Rückstrahlkoeffizient (eine Materialkonstante) und O_d die diffuse Farbe der Oberfläche. Der zweite Teil der Gleichung wertet die gerichten Lichtquellen aus. N ist der Oberflächennormalenvektor und L_i der normalisierte Vektor, der auf die i -te Lichtquelle weist. Das Skalarprodukt $N \cdot L_i$ ist 0, wenn die Vektoren senkrecht aufeinander stehen; die i -te Lichtquelle liefert in diesem Fall keinen Beitrag zu Beleuchtung des Punkts. I_i ist die Intensität der i -ten Lichtquelle und k_d der Diffus-Koeffizient.

8. Nach J. H. Lambert (1728-1777), Physiker und Philosoph

Abb. 2-4
Phong-Beleuchtung mit
variierenden
Spiegelungsexponenten
und -koeffizienten. Der
Exponent nimmt gegen
rechts zu, der Koeffizient
nimmt gegen unten zu.
(Apple Technical
Publications 1997a, S. 920)



Das Phong-Beleuchtungsmodell (BUI-TUONG 1975, Abb. 2-4) basiert auf diffuser und Spiegelungsreflexion (*Specular Reflexion*). Für einen bestimmten Punkt auf der Oberfläche gilt die Gleichung (APPLE TECHNICAL PUBLICATIONS 1997a, S. 919):

$$I_{\text{Phong}} = I_a k_a O_d + \sum_i [((N \cdot L_i) I_i k_d O_d) + ((R \cdot V)^n k_s)]$$

Die Gleichung entspricht bis auf den letzten Summanden der Lambert-Gleichung. Die Erweiterung der Gleichung errechnet die Spiegelungsreflexion. R ist die Richtung der Reflexion und V die Blickrichtung. Die Materialkonstante Spiegelungsreflexionsexponent (n) bestimmt, wie schnell die Spiegelungsreflexion bei zunehmendem Auseinanderklaffen von Reflexions- und Blickrichtung abnimmt. Hohe Werte von n liefern kleine Spiegelungsreflexionen. Ist n gleich 0, wirkt die Spiegelungsreflexion überall und hebt sich damit auf. k_s ist der Spiegelungskoeffizient – wiederum eine Materialkonstante – und bestimmt die Helligkeit der Reflexion. Die Wirkung von Spiegelungsexponent und -koeffizient wird in Abb. 2-4 gezeigt. Der Spiegelungsexponent nimmt von links nach rechts zu; die Spiegelung wird kleiner. Der Spiegelungskoeffizient und damit die Helligkeit der Spiegelung nimmt von oben nach unten zu.

Aus den Gleichungen der beiden Beleuchtungmodelle ist ersichtlich, dass die Normale eines jeden Punktes der Oberfläche (im Rahmen der Auflösung) bekannt sein muss. Um den Aufwand der Berechnungen zu reduzieren, gibt es verschiedene Interpolationsmethoden. Die einfachste besteht darin, einem Polygon genau eine Normale zuzuordnen. Dieses Verfahren wird oft *Flat-Shading* genannt. Lineare Interpolation der Oberflächennormalen zwischen den berechneten Knotennormalen ermöglicht fließende Übergänge. Dieses Verfahren wird auch *Gouraud-Shading* (GOURAUD 1971) genannt.

2.2.5 Animation

Der Begriff «Animation» beinhaltet nach FOLEY/VAN DAM/FEINER/HUGHES (1996, S. 1057) alle Änderungen eines geometrischen Modells, die sichtbare Auswirkungen haben. Animation schliesst *Bewegungsdynamik*, also die zeitabhängige Veränderung der Position von Komponenten, *Aktualisierungsdynamik* (Wechsel der Erscheinung einzelner Komponenten, wie Form, Farbe, Transparenz, Struktur, Textur etc.) wie auch Änderungen der Kamera-position, der Kameraorientierung, der Beleuchtung und selbst der Renderingtechniken ein. Da in dieser Arbeit das geometrische Modell an Ort bleibt und nur die Kamera ihre Position ändert, wird der Begriff «Bewegungsdynamik» abweichend für jegliche Änderung von Kameraparametern gebraucht. Ob traditionelle Animation wie Film und Video oder computergestützte Animation – das Ergebnis ist eine sequentielle Anordnung von einzelnen Bildern (*Frames*).

Die *Key-Frame*-Technik erlaubt es, die wichtigsten Zeitpunkte des Ablaufs in entsprechenden statischen Bildern, den *Key-Frames*, wiederzugeben. Die dazwischenliegenden Bilder werden interpoliert. Entscheidend für das Ergebnis sind die Interpolationsmethoden: Die einfach zu berechnende lineare Interpolation erzeugt beim Übergang zur nächsten interpolierten Sequenz oft unerwünscht abrupte Änderungen. Hier können *Splines* oder andere parametrisierte kubische Kurven Abhilfe schaffen.

Nach der eingangs erfolgten Begriffsklärung kann Animation mit dynamischer Wiedergabe gleichgesetzt werden, die im Prinzip eine sequentielle Wiedergabe statischer Bilder ist. Der Eindruck von fließender Bewegung entsteht, sobald die Verschmelzungsfrequenz überschritten wird. Die Verschmelzungsfrequenz ist die Bildrate, bei der kein Flimmern mehr auftritt. Sie ist von der Beleuchtungsstärke abhängig. Als Richtwert können 50-60 Bilder pro Sekunde angenommen werden; zur Optimierung ist die Bildwiederholrate des Monitors zu berücksichtigen (DÜNOW 1999, S. 134).

Interaktiv-dynamische Wiedergabe erlaubt nun, die Dynamik über Eingabegeräte wie Maus und Tastatur zu steuern. Wünschbar sind natürlich Bildraten in der Grösse der Verschmelzungsfrequenz. Es ist schwierig, die untere Grenze der Bildrate, die noch Interaktivität erlaubt, festzulegen. Bei der von **Geo3D** unterstützten Bewegungsdynamik zeigte sich, dass mit zwei Bildern pro Sekunde die Bewegung noch knapp als gerichtet wahrgenommen werden kann. 15 Bilder pro Sekunde ergeben den Eindruck von fließender, wenn auch nicht ruckfreier Bewegung.

2.3 QuickDraw 3D (QD3D)

Die Entwicklung einer 3-D-Applikation wie **Geo3D** ist als Einpersonen-Unternehmen nur dann zu bewältigen, wenn auf eine hochentwickelte 3-D-Architektur zugegriffen werden kann. QuickDraw 3D ist eine solche, von *Apple Computer* 1995 entwickelte umfassende 3-D-Architektur. Im Folgenden werden kurz die wichtigsten Gründe genannt, wieso QD3D für dieses Unternehmen geeignet war.

QD3D besitzt ein eigenes, voll integriertes 3-D-Austauschformat (*3D Metafile Format*, kurz 3DMF), das die interne objektorientierte Struktur von QD3D in abgespeicherten Dokumenten widerspiegelt. QD3D kann die 3-D-Daten (wahlweise) selber verwalten; **Geo3D** muss in diesem Fall die Struktur des Modells nicht kennen. Zudem bietet QD3D eigene Renderer an. Diese drei Features erlauben es, mit sehr wenig Aufwand beliebige 3DMF-Dateien zu lesen und den Inhalt darzustellen. Da **Geo3D** für verschiedenste Daten eingesetzt werden soll, ist dies sehr wichtig. Die Unterstützung von Betriebssystem, Hardware (einschliesslich allfällig vorhandener 3-D-Hardware) wird an QD3D delegiert und bietet so keine Probleme. QD3D ist frei erhältlich und, da inzwischen integraler Bestandteil des Industriestandards für digitales Video, *QuickTime* (<http://www.apple.com/quicktime/>), auch auf vielen Computern installiert. QD3D erlaubt mit geringem Aufwand das Identifizieren von 3-D-Objekten und -Koordinaten über Eingabegeräte.

Eine detaillierte Beschreibung von QD3D ist in APPLE TECHNICAL PUBLICATIONS 1997a (als ältere gedruckte Ausgabe für QD3D Version 1.0: APPLE COMPUTER 1995) zu finden, die Ergänzungen zu Version 1.6 in DIERKES/LUCE/GREENSTONE 1999. Ergänzende Informationen bieten FERNICOLA/THOMPSON 1995/1996, FALCO/McBRIDE 1996, SCHNEIDER 1996/1996a, THOMPSON/FERNICOLA/DAVIDSON 1996, KUECHLE 1997, THOMPSON 1997, THOMPSON/KUECHLE 1998, GREENSTONE 1998. QD3D und 3DMF werden in den folgenden zwei Unterkapiteln kurz vorgestellt.

2.3.1 QuickDraw 3D-Architektur

Abb. 2-5 bietet einen Überblick über die QD3D-Architektur. Dazu bemerken FERNICOLA/THOMPSON 1995, S. 10: "The QuickDraw 3D architecture isolates in a layer within the system software those things that all developers have to do, leaving them to concentrate on the code that will allow their application to stand out. This architecture can be thought of as a sandwich filling that sits between your application and the hardware it's running on, isolating you from having to deal with operating system and hardware issues directly." Die einzelnen Bereiche der Architektur werden im Folgenden kurz erläutert.

QD3D ist objektorientiert. Es existiert eine Basisklasse, von der alle weiteren Subklassen abgeleitet werden (Abb. 2-6). QD3D ist in hohem Mass plattformunabhängig. Die Fenstersystem-spezifischen Informationen werden im *Draw-Context*-Objekt isoliert. QD3D liegt seit April 1999 in der Version 1.6

Applikation						
Geometrie	Datei I/O	Picking	Licht	Kamera	Attribute	Shading
Rendering						
Hardware/Betriebssystem (ev. mit 3-D-Hardware-Beschleunigung))						

Abb. 2-5 QuickDraw 3D Architektur (vereinfacht nach Fernicola/Thomson 1995, S 11)

vor und ist für Mac OS und diverse Windows-Varianten (95, 98, NT ab Version 3.5.1) erhältlich.

QD3D unterstützt verschiedenste Geometrien: von Punkten über diverse zweidimensionale Geometrien zu Quadern, Ellipsoiden, Zylindern, Tori und vier Typen von Polygonnetzen. Diese werden, da für geographische Anwendungen von besonderer Bedeutung, im folgenden Unterkapitel näher beschrieben. Auch NURBs (*non uniform rational B-splines*) werden unterstützt.

Das Lesen und Schreiben von 3DMF-Dateien ist, da 3DMF die Objektstruktur von QD3D widerspiegelt, sehr flexibel. Vom selektiven Lesen und Schreiben eines Wertes bis zum Übernehmen der ganzen Datei ohne Kenntnis des Inhalts ist alles möglich.

Der Begriff *Picking* umschreibt das Identifizieren von Objekten und Koordinaten, die von einem bestimmten Objekt, dem Pick-Objekt, geschnitten werden. Ein Pick-Objekt ist entweder ein Strahl im 3-D-Raum oder ein Bereich im 2-D-Fenster. Im zweiten Fall werden die Objekte, die auf den Pick-Bereich projiziert werden, ermittelt. Picking ist ein wichtiges Mittel für die Interaktion.

Licht-Objekte ermöglichen die Beleuchtung einer Szene. Licht ist in QD3D additiv; auf eine bestimmte Oberfläche fällt die Summe des Lichts aller vorhandenen Lichtquellen. Die Wirkung des Lichts auf die einzelnen Punkte der Oberflächen ist abhängig von oberflächenwirksamen Attributen, Schattierungs- und Interpolationsmethoden. Angeboten werden ambientes Licht (*ambient light*), paralleles Licht (*directional light*), punktförmiges Licht (*point light*) und Spot-Licht (*spot light*).

Verschiedene Attribute und Stile können global oder lokal festgelegt werden. Sie definieren Materialeigenschaften (z. B. Farbe, Interpolationsmethode, Nebel) wie auch Zusatzinformationen (z. B. Internet-Adressen). Es können beliebige *Custom Attributes* definiert werden.

Shading regelt die visuelle Erscheinung der Oberflächen in Abhängigkeit von Beleuchtung, Position, Orientierung, Interpolationsmethode und Materialeigenschaften. Dabei wird zwischen Beleuchtungs-Shaders und Oberflächen-Shaders (Texturen) unterschieden. In QD3D sind Texturen 2-D-Rasterbilder, die auf eine 3-D-Oberfläche projiziert werden.

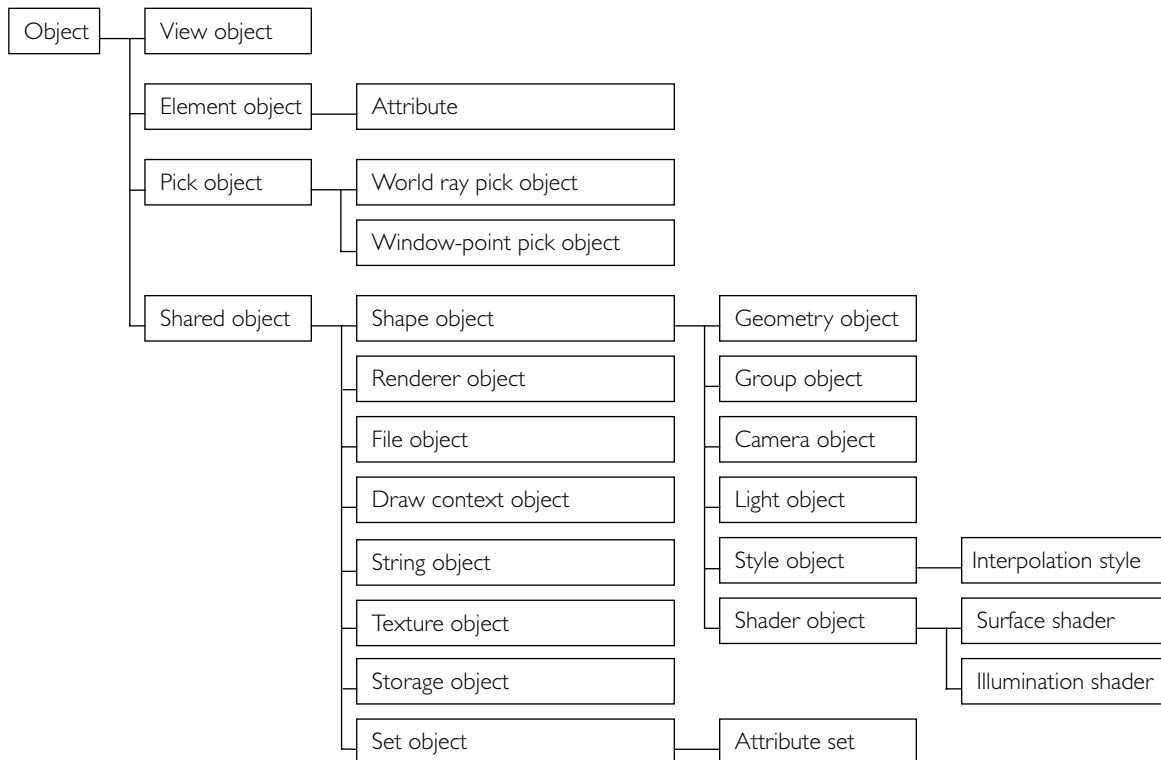


Abb. 2-6 QD3D-Klassenhierarchie: Ab 3. Kolonne nur ausgewählte Klassen (nach Quick-Draw-3D-API, <http://developer.apple.com/sdk/>).

QD3D unterstützt *Immediate-Mode* und *Retained-Mode*-Rendering. Beim Immediate-Mode verwaltet die Applikation die Daten und übergibt beim Rendering die notwendigen Parameter. Im Retained-Mode verwaltet QD3D die Daten. Der Retained-Mode erlaubt es, z. B. ein geometrisches Modell aus einer Datei zu rendern, ohne dass die Applikation über die Struktur des Modells Bescheid wissen muss. Beide Modi können nebeneinander gebraucht werden. QD3D stellt zwei Renderer-Typen zur Verfügung: Der *Wireframe Renderer* stellt Drahtgittermodelle dar, der *Interactive Renderer* – je nach Einstellung – schattierte und interpolierte Oberflächen. Ausserdem bietet QD3D eine Schnittstelle für Renderer von Drittanbietern⁹. QD3D unterstützt Hardware- und Software-Rendering. Allfällig vorhandene 3-D-Beschleunigerkarten¹⁰ werden erkannt und unterstützt. Für das Arbeiten mit Transparenzen ist 3-D-Hardware erforderlich.

9. Zum Beispiel *FineArt 3D* von *Inklination* (<http://www.inklination.com>). *PictureMaker*, ein Raytracing-Renderer von *LightWork Design* (<http://www.lightwork.com>) ist z. Z. nicht erhältlich.

10. Seit Herbst 1998 werden alle PowerMacs wie auch die meisten IBM-kompatiblen Computer mit leistungsfähigen 3-D-Graphikkarten ausgerüstet. Damit ist die letzte Hürde zur weiten Verbreitung von 3-D-Anwendungen gefallen. Zum aktuellen Stand der 3-D-Hardware siehe *Dünnow 1999* und *1999a*.

Daneben bietet QD3D perspektivische wie orthographische Kameras, gut ausgebaute Mathematik-Funktionen, Objekte zum Speichern und Gruppieren von Daten wie auch die Unterstützung von Eingabegeräten mit beliebig vielen Freiheitsgraden an.

2.3.2 3D Metafile Format (3DMF)

3DMF (*3D Metafile Format*) ist das QuickDraw-3D-eigene Standarddateiformat für 3-D-Graphik. 3DMF unterstützt verschiedenste Arten von geometrischen Objekten, Texturen, Schattierungen, Lichtquellen, Kameras, Renderers, Materialeigenschaften, hierarchische Informationen etc.: Fast alle Objekte, die mit QuickDraw 3D generiert werden können, lassen sich mit 3DMF beschreiben und speichern.

3DMF kann in Text- oder binärer Form geschrieben werden. Eine Binärdatei ist im allgemeinen kleiner und wird von einer Applikation wesentlich schneller gelesen. Eine Textdatei hingegen kann in einem Texteditor kontrolliert und editiert werden. Zudem ist sie garantiert plattformunabhängig; ASCII ist einer der wenigen wirklichen Standards in der Computerwelt. Unter dem Mac OS sollte der *File-Typ*¹¹ auf «3DMF» gesetzt sein, damit Applikationen das Dateiformat erkennen können. Der Typ «TEXT» ist zulässig, wenn das Dokument in einem Texteditor bearbeitet wird.

Das 3DMF ist hierarchisch strukturiert. Attribute werden vererbt und können überschrieben werden. Die Objekthierarchie entspricht der von QuickDraw 3D. Mit Hilfe von Labels und Referenzen können Redundanzen vermieden werden. Zudem kann 3DMF durch sogenannte *Custom Attributes* erweitert werden. Eine detaillierte Beschreibung des Formats ist in APPLE TECHNICAL PUBLICATIONS 1997a zu finden, die Ergänzungen zu Version 1.6 in DIERKES/LUCE/GREENSTONE 1999.

Für geographische Daten sind insbesondere Polygonnetze interessant. Mit Polygonnetzen können beliebige offene oder geschlossene 3-D-Oberflächen beschrieben werden. Da die einzelnen Knoten im Allgemeinen von mehreren Polygonen verwendet werden, sind Polygonnetze effizienter als eine Vielzahl entsprechender einzelner Polygone. QD3D bietet vier verschiedene Typen von Polygonnetzen an (eine detaillierte Beschreibung findet sich in SCHNEIDER 1996):

- Die *Trigrid*-Primitive ist wegen der impliziten rektangulären Topologie sehr effizient, aber topologisch unflexibel. Sie eignet sich aber gut für Digitale Geländemodelle mit Grid-Datenstruktur. Nachteilig ist, dass die Triangulation der Rechtecke nach einem vorgegebenen Muster erfolgt.

11. Unter Mac OS wird das verwendete Format durch den File-Typ angezeigt. Das Suffix des Dateinamens erfüllt in vielen anderen Betriebssystemen dieselbe Funktion (z.B. *name.3dmf*). Von Apple Computer sanktionierte Suffixe sind «3dmf», «3dm», «qd3d» und «qd3» (*QuickTime 3 Animation and 3D File Formats*: <http://www.apple.com/quicktime/authoring/ffan.html>).

- Die *Polyhedron*-Primitive ist topologisch und bezüglich der Attributierung von Flächen, Knoten und Kanten sehr flexibel. Polyhedrons sind schnell und platzsparend, d.h. für die meisten Anwendungen geeignet.
- Die *Mesh*-Primitive ist die Primitive der Wahl für interaktives Konstruieren und Editieren. Sie braucht aber viel Speicherplatz und ist sehr langsam beim Einlesen und eher langsam beim Rendering. Sie ist daher für Viewer weniger geeignet.
- Die Daten der *Trimesh*¹²-Primitive (Geometrie wie Attribute) werden im Gegensatz zu den andern besprochenen Primitiven in expliziten Arrays abgelegt und sind daher sehr stark komprimiert. Das macht sie extrem platzsparend und sehr schnell beim Rendern, Einlesen und Schreiben. Allerdings bestehen *Trimeshes* konsequent aus einer Dreieckstruktur. Da die Attribute in Arrays abgelegt sind, ist Trimesh nur dann geeignet, wenn jeder Knoten oder jede Fläche dasselbe Attribute mit variablem Werte hat (z.B. Knoten-Normalen zum Glätten der ganzen Oberfläche, uv-Koordinaten für Texturen).

Sind komplexe Einfärbungen der Polygonnetze erforderlich (z.B. thematische Darstellungen auf einen Geländemodell), dann sind Texturen die effizienteste Lösung.

Als Beispiel einer 3DMF-Datei wird in Abb. 2-7 der Text eines digitalen Geländemodells mit einem vertikalen Farbverlauf gezeigt. Es wird ein Trimesh mit *SurfaceUV*-Attribut (bestimmt für jeden Knoten einen 2-D-Parameter für die Textur), Normalen-Attribut (Normale für jeden Knoten; erlaubt es, mit entsprechender Interpolationsmethode die Oberfläche zu glätten) und Textur verwendet. Die Arrays sind gekürzt (gekennzeichnet mit «...»).

12. Siehe auch Greenstone 1998.

```

3DMetafile ( 1 6 Normal tableofcontents0> ) # Obligator Header mit Versionsangabe
Container (
  TriMesh (
    12800 0 0 0 6561 2          # Anzahl Dreiecke (12 800), Flächen-Attribute (0), Kanten (0),
                                # Kantenattribute (0), Knoten (6 561), Knotenattribute (2)
    6557 1 6558 ... 6558 0 6559      # 3 Knoten pro Dreieck (Werte gleich Knotennummer)
    -1025 1791.8 0 ... -1050 1790.1 -25  # Knotenkoordinaten
    -3000 1200 -2000 -1000 2000 0 False ) # Bounding Box (2 Begrenzungskoordinaten)
  AttributeArray (
    3 0 2 0 0          # Attributetyp Normalen (3), Reserviert (0!),
                      # Knotenattribute (2), Attributarrayposition (0), Flag (0!)
    -0.10984 0.9925387 -0.0529354 ...  # Normalenvektoren
    -0.1603126 0.9855223 0.05518701
  )
  AttributeArray (
    1 0 2 1 0          # Attributetyp SurfaceUV (1), Reserviert (0!),
                      # Knotenattribute (2), Attributarrayposition (1), Flag (0!)
    0 0.746296 ... 0 0.739492          # UV-Koordinaten (hier U = 0, da eindimensionaler Verlauf)
  )
  Container (
    AttributeSet ( )
    Container (
      TextureShader ( )          # Shader für Textur
      PixmapTexture (           # Texturtyp
        2 32 8 32 RGB32 BigEndian BigEndian  # Breite 2, Höhe 32 Pixel, Bytes pro Zeile, Pixel-
                                              # größe, Pixeltyp, Bit- und Byte-Adressierung
        0x00AD0C0C00AD0C0C00AC130C00AC130C ... # 4 Farbwerte pro Zeile
        0x008FCD09008FCD09008ED309008ED309
      )
    ) # Container
  ) # Container
) # Container

```

Abb. 2-7 3DMF-Datei mit Trimesh und Textur (gekürzt)

3 «Geo3D»-Konzept

Ever tried. Ever failed. No matter. Try again. Fail again. Fail better.

SAMUEL BECKETT

3.1 Einleitung

Das **Geo3D**-Konzept ist, wie in der Zielsetzung (Kap. 1.2) formuliert, das Kernstück dieser Arbeit. Parallel zum Entwurf des Konzepts wurden wesentliche Features¹³ implementiert und so auf ihre Tauglichkeit geprüft. Das Konzept ist somit ebenso Grundlage wie Ergebnis der Implementation von **Geo3D**. Die Ideen, die hier präsentiert werden, reflektieren meine subjektive Sicht; weder eine repräsentative Befragung noch Perzeptionsstudien standen Pate. Nun, so subjektiv sind die Ideen nicht, denn bestehende Programme¹⁴ wurden studiert, nützliches übernommen, einiges zum vornherein verworfen. Anfänglich wenige, später vielleicht zwei Dutzend Leute liessen sich in Diskussionen verwickeln; die Veröffentlichung von **Geo3D** (<http://www.data-comm.ch/luzern/>) übers Internet begann Früchte zu tragen.

Der Schwerpunkt des **Geo3D**-Konzepts liegt in der Beschäftigung mit der Navigation im virtuellen Raum. Dabei soll die Synthese von menschlicher Wahrnehmung und geometrischen Grundlagen zu einem sinnvollen Konzept der Navigation führen. Weiter werden die Themen User-Interface, Rende-

13. Im Computerbereich üblicher Ausdruck für Merkmale/Fähigkeiten, die eine Applikation anbietet.

14. Von besonderem Interesse sind die VRML-Browser (siehe <http://www.vrml.ch>) wie *WorldView* (<http://www.intervista.com>) für das Mac-OS und *Cosmo Player* (<http://www.cosmosoftware.com>) für Windows und Silicon Graphics Workstations. Daneben diverse QD3D-Applikationen wie *Fly Through* (<http://www.halcyon.com/jennings/FlyThrough.html>), *Walker* (von *LightWork Design* <http://www.lightwork.com>, nicht mehr erhältlich) und der QD3D-eigene Viewer.

ring-Qualität (darstellungsrelevante Parameter), weitere Analyse-Features und Schnittstellen zu Applikationen und Dateien zur Sprache kommen. Umfassende Schnittstellen sind Grundlage für erweiterte Visualisierungssysteme.

Das folgende Konzept kann und soll nicht abschliessend sein. Weitere Features sind denkbar und können realisiert werden. Das Konzept sollte sinnvollerweise zusammen mit dem Programm Geo3D betrachtet werden. Im Text wird jeweils darauf hingewiesen, welche Teile des Konzepts implementiert sind.

3.2 Bemerkungen zum User-Interface

Das *User-Interface*, oft auch *Human-Interface* genannt, ist die Oberfläche, mit Hilfe derer mit einer Applikation interagiert werden kann. Heute sind die User-Interfaces typischerweise mehr graphisch denn textorientiert. Die einzelnen Elemente verwenden Metaphern, die vorwiegend dem Büro (Schreibtisch, Ordner) und technischen Geräten (Knöpfe, Regler, *Radio Button*) entlehnt sind.



Abb. 3-1 User-Interface unter Mac OS 8.6 am Beispiel Geo3D (Screenshot)

Der Menübefehl «Beenden (*Quit*)» ist bei jedem Mac-Programm auf Anhieb zu finden. Die vielen Übereinstimmungen zwischen Programmen sind nicht zufällig: Sie sind den Richtlinien von *Apple* entnommen. Die Berücksichtigung dieser Richtlinien (APPLE COMPUTER 1995a, APPLE TECHNICAL PUBLICATIONS 1997) und eine gute Strukturierung der Oberfläche sind Grundlagen einer für Benutzer/innen gut nachvollziehbaren und damit leicht anzuwendenden Applikation. Geschickt gewählte Default-Einstellungen tragen ein Weiteres dazu bei.

Geo3D hält sich an diese Richtlinien, die im Konzept nicht weiter diskutiert werden. Die Ausnahme ist der zentrale und zugleich am schwierigsten zu lösende Teil des User-Interface: die Steuerung der synthetischen Kamera (Wie navigiere ich im synthetischen Raum?). Es lassen sich hierzu leider keine allgemein gebräuchlichen Lösungen finden¹⁵. Die vorgeschlagene Lösung lehnt sich zum Teil an QuickTime VR¹⁶ an. Sie wird im Kap. 3.3.5 näher erläutert.

3.3 Navigation

Die Navigation im virtuellen Raum verfolgt mehrere, voneinander abhängige Ziele. Beliebige Ansichten des geometrischen Modells müssen erreicht werden können, fließende Bewegung durch den Raum muss gewährleistet sein. Beide Ziele sollen durch vielfältige Navigationsmöglichkeiten verschiedenen Ansprüchen gerecht werden. Weiter soll die Navigation einfach und intuitiv erfolgen. Da kleine Fehler sich schnell kumulieren, soll die Navigation präzise und stabil sein. Weil freie Navigation schnell zu Orientierungslosigkeit führen kann, muss die Rückkehr zu bekannten Ansichten angeboten werden.

3.3.1 Alltäglicher vs. synthetischer Raum

Typische Szene aus dem Leben eines Diplomanden (1): Der Diplomand geht in die Küche mit üblicher Deckenleuchte. Typische Szene aus dem Leben eines Diplomanden (2): Derselbe nimmt ebendort eine Kaffeetasse aus dem Schrank. Der Diplomand würde, müsste er Rechenschaft über die erfolgten Bewegungen ablegen, etwas gestelzt sagen, er habe im ersten Fall seine Position in der Wohnung verändert und er habe im zweiten die Position der Kaffeetasse relativ zu sich verändert. Und er würde weder sagen, «Sie bewegt sich doch», noch «Die Küche kam mir entgegen». Zum Licht würde er bemerken, dass sich die Deckenleuchte auf der Tasse spiegle.

15. Die Richtlinien für QuickDraw 3D (*Apple Computer 1996*) sind auf Programme zur Modellierung von 3-D-Objekten ausgerichtet.

16. *QuickTime VR* (VR für Virtual Reality) ist eine Erweiterung von *QuickTime*, einem Industriestandard für digitales Video. *QuickTime VR* erlaubt unter anderem, geschlossene 360°-Panoramen mit zugehöriger Navigationstechnik darzustellen. Siehe dazu <http://www.apple.com/quicktime/>.

Es ist nicht zwingend, alltägliches Denken zur Grundlage der Navigation im synthetischen Raum zu machen. Da das Rendering auf einer synthetischen Kamera basiert, macht es aber durchaus Sinn, den Blick des Diplomanden der Kamera gleichzustellen. Ob der Diplomand in die Küche geht oder die Küche ihm entgegenkommt, spielt hingegen optisch keine Rolle. Szene 1 und Szene 2 unterscheiden sich technisch nur durch die Beleuchtung. In Szene 2 verändert sich die Lage der Leuchte relativ zur Tasse. Die Leuchte liesse sich aber auch an die Tasse koppeln.

Da wir ja das Ziel verfolgen, aus geographischen Daten modellierte geometrische Modelle darzustellen, drängt sich die Frage auf, mit was für einem (synthetischen) Raum wir es zu tun haben. Der hier verwendete synthetische Raum ist dreidimensional und kartesisch. Genaugenommen ist er, bedingt durch die limitierten signifikanten Stellen von Gleitkommazahlen, erstens endlich und besteht zweitens aus endlich vielen diskreten Punkten, die gegen den Ursprung des Koordinatensystems dichter gestreut sind. Die ungleiche Streuung ist eine Folge der exponentiellen Darstellung der Gleitkommazahlen.

Im einfachen Fall, in dem sich geometrische Objekte nicht relativ zueinander bewegen, müssen also, um Bewegung darzustellen, die Koordinaten des Objekts relativ gegenüber den Koordinaten der Kameraparameter verändert werden. Das wichtigste Argument¹⁷, hierzu nur die Kameraparameter zu verändern, ist folgendes: Geodaten basieren üblicherweise auf einem zum vornherein festgelegten Koordinatensystem. Ist das Koordinatensystem orthogonal (wie im kleinmassstäblichen Bereich üblich), können die Koordinatenwerte ohne grossen Aufwand dem geometrischen Modell entnommen werden. Falls später auch gewisse geometrische Objekte relative Bewegungen zueinander erfahren sollen, lassen diese Bewegungen sich auf obige Weise von der Navigation entkoppeln. Das Gleiche gilt auch für die Beleuchtung: Die räumliche Beziehung zum geometrischen Modell wird durch die Kamerabewegung nicht beeinflusst. Eine Lichtquelle könnte aber ohne weiteres animiert werden. Der Entscheid, mit Hilfe der Kamera im Raum zu navigieren, bedeutet aber nicht, dass wir nicht Bewegungen des geometrischen Modells simulieren können. Ob wir den Eindruck haben, dass wir (die Kamera) uns bewegen oder ob wir eine Bewegung des Modells wahrnehmen, ist von der kontextabhängigen Beleuchtung, dem User-Interface und – nicht zu unterschätzen – dem jeweiligen individuellen Vorstellungsvermögen abhängig.

Navigation bedeutet also, Kameraparameter zu verändern. Damit wird offensichtlich, dass die Navigation nicht nur Änderung des Kamerastandorts (die Fortbewegung an sich) bedeutet, sondern auch Änderung der Blickrichtung sowie der Lage der View-Plane und der Richtung des Up-Vektors. Technisch

17. Das naheliegende Argument, dass es einfacher ist, die Kamerakoordinaten statt den vielen Modellkoordinaten zu ändern, ist in unserem Fall kaum stichhaltig. Der Implementationsaufwand ist etwa gleich gross. Der Aufwand der Berechnung spielt im Rendering-Prozess eine untergeordnete Rolle.

formuliert bedeutet nun Navigation die Änderung eines oder mehrerer Kameraparameter.

Für ein taugliches Navigationsmodell muss die menschliche Vorstellung von Raum einfließen. Dazu VILÉM FLUSSER (1992 S. 43): «Der Lebensraum erklärt sich daraus, daß wir Würmer sind. D.h. wir sind Röhren, durch die die Welt reingeht und am anderen Ende rauskommt. Dadurch haben wir die Vorstellung von vorne und hinten. Außerdem sind wir bilateral symmetrisch, dadurch haben wir die Vorstellung von links und rechts. Und da wir aufrechte Würmer sind, haben wir die Vorstellung von oben und unten. Der Lebensraum ist eine schmale Kiste, die theoretisch endlos lang und breit ist, aber maximal drei Meter hoch und zehn Zentimeter tief unter die Erde geht. Diese Kiste ist derartig flach, daß wir kubisch gesehen nicht zu Hause sind.» Wenn wir diese Darstellung akzeptieren, können wir zwei weitere Kriterien aufstellen: Die Wahrnehmung der eigenen Position ist für die Vorstellung von Raum relevant. Ich muss wissen, wo und in welcher Lage ich mich befinde, um zu verstehen was ich sehe. Und es gibt eine bevorzugte Ebene, an der wir uns orientieren. Es ist die (horizontale) Ebene, in der wir uns bewegen.

3.3.2 Zur Geometrie der Navigation

Ausgehend von der Annahme, dass Navigation als Änderung der Kameraparameter beschrieben werden kann, muss nun eine Methode entwickelt werden, die aufzeigt, wie die relevanten Parameter Kamerastandort, Blickrichtung und Up-Vektor gesteuert werden können. Da die Veränderung der Kameraparameter auf den aktuellen Parametern beruhen, ist es zweckmässig, von den Kamerakoordinaten auszugehen. Der Anschaulichkeit willen gehen wir bei den folgenden Überlegungen immer von einer «aufrechten» Position aus. Die Blickrichtung liegt in der xz -Ebene¹⁸ des Weltkoordinatensystems (im Gegensatz zum Kamerakoordinatensystem), der Up-Vektor ist parallel zur globalen y -Achse und zeigt in Richtung der globalen y -Achse. Die Begriffe vorne, hinten, oben, unten, links und rechts werden immer kamera-bezogen verwendet. Sie entsprechen der subjektiven Wahrnehmung. Wenn die View-Plane, wie wir in Kapitel 2.2.3 voraussetzen, normal zur Blickrichtung liegt, können wir Veränderungen der Parameter durch Translation und Rotation, jeweils in jeder der drei Dimensionen und auf den Ursprung Kamerastandort bezogen, beschreiben. Die gewonnen sechs Varianten sind:

- Translation in Richtung der Normale der Ebene Blickrichtung-Up-Vektor (u -Achse). Subjektiv bedeutet das eine Verschiebung nach links oder rechts (u -Translation).
- Translation in Blickrichtung (n -Achse). Subjektiv gesehen eine Vorwärts- oder Rückwärtsbewegung (n -Translation).

18. Wenn wir in Zukunft der Einfachheit halber von Ebene sprechen, meinen wir eine beliebige Ebene einer Ebenenschar (Menge aller zueinander parallelen Ebenen). Hier ist es konkret eine Ebene, die parallel zur x - und z -Achse liegt.

- Translation in Richtung des Up-Vektors (v-Achse). Subjektiv eine Verschiebung nach oben oder unten (*v-Translation*).
- Rotationsachse um die Normale der Ebene Blickrichtung-Up-Vektor (u-Achse). Die Rotationsachse geht durch die Kameraposition. Subjektiv ein Drehen nach oben oder unten (*pitch, u-Rotation*).
- Rotation um den Up-Vektor (v-Achse) durch die Kameraposition. Subjektiv ein Drehen nach links oder rechts (*yaw, v-Rotation*).
- Rotation um die Blickrichtung (n-Achse) Subjektiv ein Drehen um die eigene Blickachse (*roll, n-Rotation*).

Es ist hilfreich, sich hier zu überlegen, wie wir das Modell bewegen können, um die gleiche visuelle Wirkung zu erzielen. Das bleibt solange trivial, wie wir die räumliche Beziehung zwischen Modell und Kamera von der Kamera her betrachten. Eine Translation der Kamera nach links, oben oder vorne muss übersetzt werden in eine Translation des Modells nach rechts, unten oder hinten. Gleiches gilt sinngemäss für die Rotation des Modells um die Kamera.

Sobald wir aber die Bewegung der Kamera vom Koordinatensystem des Modells aus betrachten, wird die Sache komplexer. Während die Translationen und die Rotation um die Blickrichtung (n-Achse) weiterhin mit negativem Vorzeichen beschrieben werden können, ist die Situation beim seitlichen Drehen und dem Drehen nach oben und unten anders. Das Rotieren der Kamera entspricht nicht dem negativen Rotieren des Modells. Die Rotationsachsen im Fall «Kamera bewegen» gehen durch den Kamerastandort, im Fall «Modell bewegen» durch irgend einen festzulegenden Punkt, den wir Rotationszentrum des Modells (kurz: Zentrum) nennen werden. Wird nun die Kamera, die auf das Zentrum gerichtet ist, seitlich rotiert, entfernt sich das Zentrum aus dem Fokus. Wird hingegen bei fixer Kamera das Modell um eine Achse durch das Zentrum rotiert, bleibt das Zentrum im Fokus der Kamera. Der letztere Fall kann ohne weiteres als Bewegung der Kamera umgedeutet werden: Die Kamera wird, wieder mit umgekehrtem Vorzeichen, auf einer Kreisbahn mit Rotationsachse durch das Zentrum rotiert. Die Blickrichtung der Kamera bleibt auf das Zentrum gerichtet. Die zwei weiteren gewonnenen Bewegungsvarianten lauten:

- Rotation um den Up-Vektor (v-Achse) durch das Rotationszentrum des Modells (*v-Zentrums-Rotation*).
- Rotationsachse um die Normale der Ebene Blickrichtung-Up-Vektor (u-Achse). Die Rotationsachse geht durch die Zentrum (*u-Zentrums-Rotation*).

Die acht Bewegungstypen haben, auf die Navigation bezogen, verschiedene Qualitäten. Die n-Translation (Translation in Blickrichtung), beliebig kombiniert mit Rotationen um den Kamerastandort, führt immer zu einer Standortveränderung in Blickrichtung. Alltäglich ausgedrückt: Ich gehe wohin ich schaue. Translationen in u- und v-Richtung (horizontal und vertikal) ent-

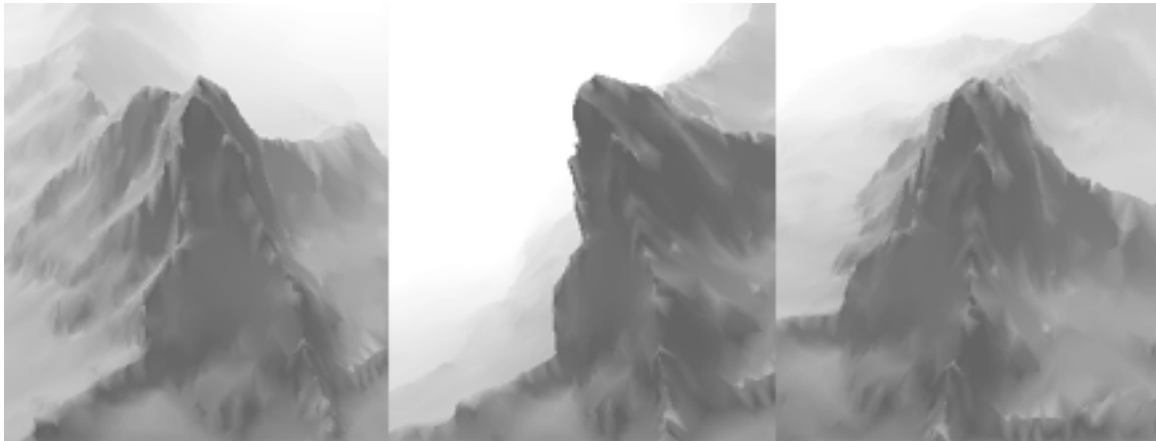


Abb. 3-2 Rolleffekt. links: Ausgangsposition mit Blick gegen unten geneigt; Mitte: Ansicht nach Zentrumsrotation – der Berg gerät aus dem Lot; rechts: Ansicht nach Zentrumsrotation mit Antirollmodus – unveränderte Kameraneigung (Datenquelle M. Heller)

koppeln Standortveränderung und Blickrichtung. Die Blickrichtung wird nicht verändert. Der Blick aus dem Fenster eines fahrenden Zuges ... Rotationen der Kamera um das Rotationszentrum des Modells schliesslich ändern den Standort und die Blickrichtung. Der Blick vom erdumkreisenden Satelliten auf die Erde ...

Beliebige Kombinationen von Translationen und v-Rotationen (seitliches Drehen oder Kreisen) verändern den Up-Vektor nicht. Unsere Wahrnehmung von oben und unten wird nicht verändert. Mit der n-Rotation (roll) wird nur der Up-Vektor verändert. Die Welt auf den Kopf stellen ... Die u-Rotation (drehen/kreisen nach unten/oben) hat unter bestimmten Umständen überraschende Auswirkungen. Wird sie alleine angewandt, ändert sich zwar die Lage des Up-Vektors, aber er hat in der von der Blickrichtung vorgegebenen View-Plane die Lage, die maximal nach oben zeigt (Die globale y-Koordinate des Up-Vektor ist maximal)¹⁹. Dies gilt auch wenn wir die u-Rotation mit Translationen kombinieren. Folgt einer u-Rotation eine v-Rotation (seitliches Drehen oder Kreisen), zeigt der Up-Vektor nicht mehr maximal nach oben. Es zeigte sich, dass dieser Roll-Effekt, beispielsweise beim Navigieren über einem Geländemodell, oft störend ist. Wir bewegen uns plötzlich nicht mehr in der dominanten Ebene – das Gelände «kippt» seitlich. Abb. 3-2 illustriert den Roll-Effekt. Es muss betont werden, dass der Roll-Effekt nicht die Folge einer objektiven Änderung der Orientierung der View-Plane ist. Er ist Ausdruck des subjektiven Verlusts des «Oben» bedingt durch

19. Von einer horizontalen Kameraposition ausgehend, stimmt diese Feststellung nur für eine Rotation im Bereich zwischen -90° und 90° . Eine Rotation um -90° oder 90° führt zu einer Blickrichtung parallel zur globalen y-Achse. Die y-Koordinate des Up-Vektors ist zwangsläufig 0. Rotation um mehr als 90° oder -90° stellen die View Plane auf den Kopf. Der Up-Vektor zeigt nun maximal nach unten.

die Veränderung der Blickrichtung. Eine wahlweise Korrektur des Rolleffekts kann als Modifikation der obigen Bewegungsvarianten gesehen werden.

Da geographische Daten im kleinmassstäblichen Bereich meist in kartesischen Koordinaten vorliegen, sind Translationen der Kamera entlang der Koordinatenachsen gut nachvollziehbar. Drei weitere Bewegungsvarianten können in den Katalog aufgenommen werden:

- Translationen der Kamera in Richtung der globalen Koordinatenachsen (x -, y -, z -Achse; x -Translation, y -Translation, z -Translation).

Ein Parameter der perspektivischen Kamera wurde bisher ausgeklammert: der Öffnungswinkel, der analog der Brennweite einer konventionellen Kamera das Zoomen ermöglicht. Das Verändern des Öffnungswinkels tangiert die andern Kameraparameter nicht.

Die bisherigen Überlegungen zur Navigation werden im nächsten Kapitel in das Navigations-Konzept einfließen.

3.3.3 Navigations-Konzept

Die Navigation im virtuellen Raum wird in dieser Arbeit als Veränderung der Kameraparameter betrachtet. Die Veränderung der Parameter wird relativ zu den aktuellen vorgenommen. Eine Ebene des virtuellen Raums wird im Sinne Flussers bevorzugt (Kap. 3.3.1). Die betrachteten Varianten der Translation und Rotation werden in sinnvollen Kombination verschiedenen Navigationsmodi zugeordnet. Als Ergänzung werden verschiedene vordefinierte Ansichten, vordefinierte Sequenzen von Ansichten, die modellbezogene Wahl von Ansichten und die Aufzeichnung und Wiedergabe der Kameraparameter vorgeschlagen. Der letzte Punkt wird im nächsten Kapitel (Kap. 3.3.4) erläutert.

Kern des Navigationskonzepts sind drei Modi plus eine Modifikation für das Navigieren einer perspektivischen Kamera. Jeder Modus hat sechs frei kombinierbare Freiheitsgrade (bez. sieben, wenn das Zoomen miteingerechnet wird). Jedem Freiheitsgrad wird eine der oben beschriebenen Bewegungsvarianten zugeordnet. Tab. 3-1 listet die verschiedenen Bewegungsvarianten auf, zeigt deren Zuordnung zu den Modi und die Auswirkungen auf die Kameraparameter.

Der Kern des Navigationskonzepts umfasst die folgenden Punkte:

- Der Modus *Walk* (Gehen) erlaubt den Gebrauch von n -, v -, u -Translation und n -, v -, u -Rotation. Er basiert also ausschliesslich auf dem Kamerakoordinatensystem. Er erlaubt beliebige Bewegung in Blickrichtung und parallel zur View-Plane. Der Modus Walk wurde in dieser Form implementiert.
- Der Modus *Center* (Zentrum) erlaubt den Gebrauch von n -, v -, u -Translation, n -Rotation, u - und v -Zentrumsrotation. Zum Kamerakoordinatensystem braucht er zusätzlich einen Punkt in Weltkoordinaten für die

Bewegung (Modi)	Kamera- position	Blickrichtung	Orientierung der View-Plane	Öffnungswinkel des Bildraums
n-Translation (W, C)	ja	nein	nein	nein
u-Translation (W, C)	ja	nein	nein	nein
v-Translation (W, C)	ja	nein	nein	nein
x-Translation (A)	ja	nein	nein	nein
y-Translation (A)	ja	nein	nein	nein
z-Translation (A)	ja	nein	nein	nein
n-Rotation (W, C, A: ohne U)	nein	nein	ja	nein
u-Rotation (W, A)	nein	ja	nein	nein
v-Rotation (W, A)	nein	ja	nein, U: (ja)	nein
u-Zentrums-Rotation (C)	ja	ja	nein	nein
v-Zentrums-Rotation (C)	ja	ja	nein, U: (ja)	nein
Zooming	nein	nein	nein	ja

Tab. 3-I Veränderung der Kameraparameter nach Bewegungsvarianten (W: Walk; C: Center; A: Axes; U: Upright)

Zentrumsrotation (das Zentrum). Das Zentrum liegt per default im Zentrum des Quaders, der das gesamte geometrische Modell umhüllt. Das Zentrum kann aber interaktiv verschoben werden. Die Spezialität dieses implementierten Modus ist die Navigation auf Kreisbahnen um das Zentrum.

- Der Modus *Axes* (Achsen) kombiniert x-, y- und z-Translation mit n-, v- und u-Rotation. Die Lageverschiebung basiert auf Weltkoordinaten, die Blickrichtung ist davon unabhängig. Dieser Modus wurde aus Zeitgründen nicht implementiert, da die x-, y- und z-Translationen bereits über die Seitenansichten (Kap. 3.5.3) verfügbar sind.
- Das Zoomen der Kamera ist in jedem Modus gewährleistet und wurde implementiert.
- Eine dominante Ebene wird festgelegt. Die Funktion *Horizontalize* (horizontal ausrichten) erlaubt die Rückkehr der Blickrichtung in die dominante Ebene und zwingt den Up-Vektor in eine zur dominanten Ebene senkrechten Lage. Ebene und Funktion wurden implementiert.
- Die Modus-Modifikation *Upright* (aufrecht bleiben) kann zugeschaltet werden. Sie bezieht sich auf die dominante Ebene. Der Up-Vektor wird in

einer maximal nach oben zeigenden Lage gehalten. Die durch den Roll-Effekt entstehende Schiefelage wird fortlaufend durch n-Rotation korrigiert. Diese Modifikation wurde implementiert.

- Die Modus-Modifikation *Collision-Detection* (Kollisionserkennung) hat keinen Einfluss auf die Kameraparameter. Sie unterbindet lediglich, dass die Kamera durch geometrische Komponenten fliegt. Diese Modifikation wurde nicht implementiert.

Als vordefinierte zentrumsbezogene Ansichten werden vorgeschlagen:

- Vordefinierte Ansichten von vorne, hinten, rechts, links, oben und unten. Diese Ansichten sind achsenparallel zu den Weltkoordinatenachsen und berücksichtigen die dominante Ebene. Der Blick richtet sich auf das Zentrum. Das Modell wird vollständig abgebildet. Diese Ansichten wurden implementiert.
- Die Kamera soll auf das Zentrum gerichtet werden können (implementiert).

Als einfache Möglichkeit, einen Bereich des geometrischen Modells in die Mitte des Blickfelds zu rücken, wird die modellbezogene Wahl einer Ansicht vorgeschlagen. Die modellbezogene Wahl erfolgt durch Selektieren eines Punkts des Modells mit der Maus. Die Kamera wird nach einem der unten erläuterten Verfahren auf den selektierten Punkt gerichtet. Folgende Varianten werden vorgeschlagen:

- Translation der ganzen Kamera. Im Modus Center wird der selektierte Punkt zugleich zum neuen Zentrum. (implementiert)
- Translation der ganzen Kamera gekoppelt mit der Funktion Horizontalize. Im Modus Center wird der selektierte Punkt zugleich zum neuen Zentrum. (implementiert)
- Analog zu den beiden Translationsvarianten werden zwei Rotationsvarianten vorgeschlagen. Dabei bleibt der Kamerastandort bestehen, die Kamera zeigt auf den selektierten Punkt (nicht implementiert).

Um in Ruhe einen Überblick über das geometrische Modell zu gewinnen, werden schliesslich als vordefinierte Sequenzen (Bewegungsanimation) vorgeschlagen:

- Automatisierte u- und v-Zentrumrotation in jeweils beiden Richtungen. Dabei wird von den aktuellen Kameraparametern ausgegangen. (implementiert)
- Kombination von automatisierten Sequenzen und interaktiver Navigation (nicht implementiert).
- Freie Definition von automatisierten Sequenzen können festgelegt werden (nicht implementiert).

Obwohl das Navigationskonzept eine allgemein verwendbare, nicht modell-bezogene Navigation vorschlägt, wird wegen der grossen Bedeutung digitaler Geländemodelle in der Geographie eine Ausnahme vorgeschlagen. Die Kamera soll in festgelegter vertikaler Distanz zum Geländemodell navigiert werden können. Die vertikale Distanz zum Modell ist wählbar. Eine Art virtueller Begehung des Geländes kann so erreicht werden. Dieses Feature wurde nicht implementiert.

3.3.4 Aufzeichnung und Wiedergabe

Aufzeichnung und Wiedergabe der Kameraparameter werden ermöglicht. Damit können geeignete Ansichten und ganze Navigationspfade (Bewegungsanimation) registriert werden. Die implementierte Grundfunktionalität umfasst:

- Sequentielles oder einzelnes Aufzeichnen der Kameraparameter.
- Sequentielles oder einzelnes Wiedergeben der Kameraparameter.
- Einfacher Zugriff auf bestimmte Ansichten (erste, letzte, vorherige, nachfolgende).
- Die Daten können über ein Animations-Format in eine Textdatei geschrieben und aus derselben eingelesen werden. Die Datei kann durch einen beliebigen Texteditor editiert werden. Andere Applikationen können Sequenzen berechnen und als Textdatei exportieren. Das Format «Animation» ist in Anhang D beschrieben.
- Unabhängig von den obigen Funktionen werden vor jeder Sequenz (ob interaktiv oder nicht; eine Sequenz besteht aus mindestens einer Änderung der Kameraparameter) die Kameraparameter automatisch aufgezeichnet. Dies erlaubt das Widerrufen der Bewegung – bei Orientierungsverlust von grosser Bedeutung.

Es zeigt sich, dass das Zusammenfassen von einzelnen Ansichten und Sequenzen ungünstig ist. Darum wird als Erweiterung vorgeschlagen:

- Trennung von Einzelansichten und Sequenzen. Auf jede Einzelansicht soll direkt über ein Menü zugegriffen werden können (nicht implementiert).

Aufgezeichnete Daten können weiter bearbeitet werden. Als nicht implementierte Ergänzungen werden vorgeschlagen:

- Verschiedene Interpolationsmöglichkeiten zwischen *Key Frames* (Kap. 2.2.5). Die Zahl der interpolierten Frames kann absolut oder in Abhängigkeit der Distanz festgelegt werden.
- Glättung der Animationspfade.
- Kombination von Wiedergabe und interaktivem Navigieren.

3.3.5 Navigation und User-Interface

Mehrere Bewegungsvarianten können in Geo3D miteinander ausgeführt werden. Um die verschiedenen Bewegungsvarianten auszuführen, werden quantitative Eingabewerte für die gewünschten Bewegungsvarianten verlangt, die dann in Bewegung umgesetzt werden. Besonders geeignet ist die Eingabe der Bewegung als Geschwindigkeit. Ein konstanter Wert wird als konstante Geschwindigkeit interpretiert.

Traditionelle und in jeder Konfiguration verfügbare Eingabewerkzeuge sind Maus und Tastatur. Der Mauszeiger (*Cursor*) wird, der Maussteuerung durch die Hand entsprechend, auf dem Monitor bewegt. Die Bildschirmkoordinaten der Mausposition sind programmintern abrufbar, die Abweichung gegenüber der vorherigen Position berechenbar. Es lassen sich also quantitative Aussagen für zwei Koordinatenachsen machen. Die Tastatur offeriert im Gegensatz dazu die parallele Eingabe²⁰ von vielen Parametern, die Aussage ist binär: Ein Taste ist entweder gedrückt oder nicht gedrückt. Die Verwendung von komplexeren Mäusen und Eingabegeräten aus der Welt der Computerspiele wie z. B. Joysticks wurde nicht weiter geprüft, kann aber als interessante Option betrachtet werden. Das Navigationskonzept ist schwerpunktmässig auf sechs Freiheitsgrade ausgerichtet und daher für solche Eingabegeräte geeignet.

Implementiert wurde die parallele Eingabe von bis zwei Freiheitsgraden via Maus und allen Freiheitsgraden über die Tastatur. Die Eingabewerte werden als Geschwindigkeit interpretiert. Ein *Clickhold*²¹ in das Fenster legt den Nullpunkt fest. Solange die Maus gedrückt gehalten wird (die Mausposition kann beliebig verschoben werden), wird vor jedem Renderingzyklus die Differenz Mausposition-Nullpunkt für beide Koordinatenachsen separat ermittelt. Die beiden Werte fliessen entsprechend den der Achsen zugeordneten Bewegungstypen in die Navigation ein.

Ein visuelles Feedback, das die aktuelle Situation der Eingabe anzeigt, ist zwingend für eine präzise Navigation (Abb. 3-3). Der Nullpunkt wird zu einer quadratischen Nullzone erweitert, deren Begrenzung wahlweise während des ganzen Clickholds angezeigt wird. Neun Zonen werden neun Zeigertypen zugeordnet: Ein Zeiger für die Nullzone (keine Bewegung) und acht Zeiger, welche die sternförmig angeordneten Zonen repräsentieren. Ermittelt werden diese Zonen durch die Kombinationen, die sich aus der Differenz Mausposition-Nullzone für beide Achsen ergeben: für jede Achse kann der Wert entweder negativ, Null oder positiv sein. Bei zwei Achsen ergeben sich neun mögliche Wertepaare: die neun Zonen.

Der Maus können verschiedene Bewegungsvarianten zugeordnet werden. Durch Voreinstellung kann das Geschwindigkeitsverhältnis Maus-Tastatur

20. Üblich ist die parallele Eingabe nur in Kombinationen mit Befehlstasten (*Command Keys*). Es ist jedoch möglich, beliebige Tastenkombinationen zu erfassen und auszuwerten.

21. Die Maustaste wird gedrückt und unten gehalten.

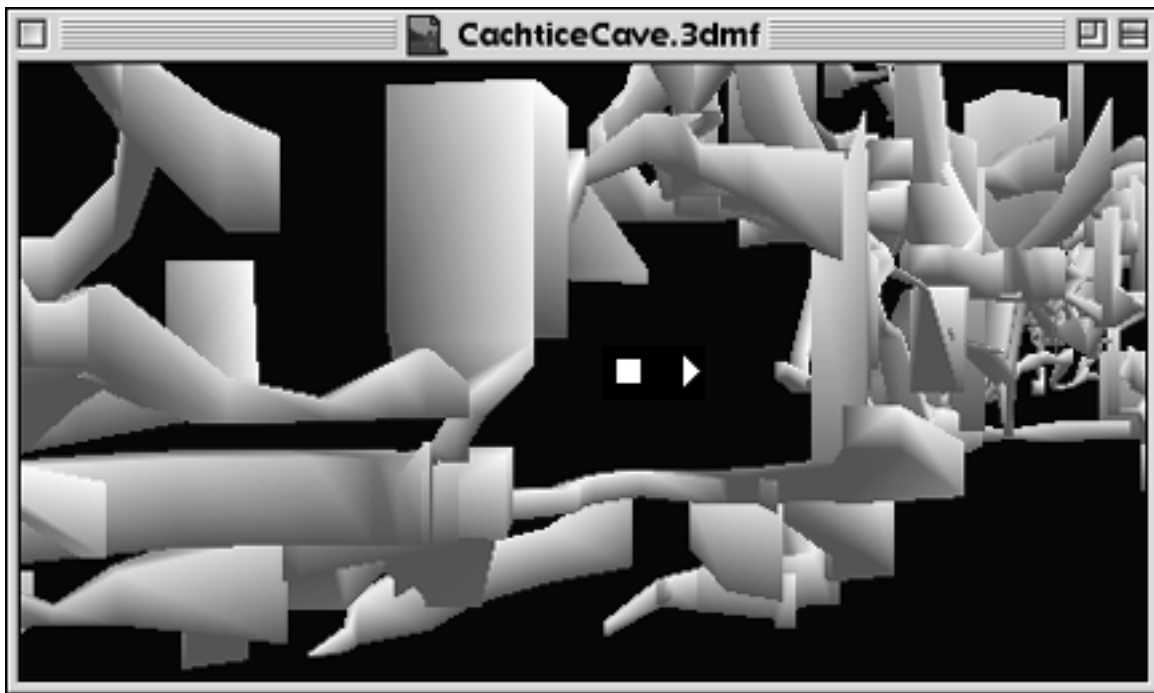


Abb. 3-3 Navigation mit der Maus: Nullzone (Quadrat in der Mitte) und Zeiger (Datenquelle M. Sluka)

(wieviele Pixel entsprechen einer Taste) festgelegt werden. Ebenso können die Geschwindigkeiten für die verschiedenen Bewegungsvarianten manipuliert sowie eine absolute Begrenzung der Bildrate festgelegt werden.

3.4 Rendering-Qualität

Unter Rendering-Qualität fasse ich alle die Darstellungsqualität beeinflussenden Parameter zusammen, die auf das gesamte Modell einwirken. Die Qualität des geometrischen Modells an und für sich wird als gegeben betrachtet und ist nicht Bestandteil dieser Arbeit. Ziel ist es, dass die Parameter, welche die Rendering-Qualität beeinflussen, verändert werden können. Dies aus zwei Gründen: Die Analyse eines Modells wird massgeblich durch die Rendering-Qualität bestimmt. Die Wahl der Rendering-Qualität richtet sich dabei nach dem Phänomen, das sichtbar gemacht werden soll (Das ist keinesfalls gleichbedeutend mit Fotorealismus). Der zweite Grund ist die Geschwindigkeit des Rendering. Muss die Bildrate erhöht werden, sind rechnerisch einfachere Methoden anzuwenden.

Der Weg ist durch QuickDraw 3D weitgehend vorgegeben: Alle relevanten QD3D-Parameter sollen auf der globalen Ebene nach Bedarf eingestellt werden können. Für 3DMF-Dokumente, die diese Parameter nicht spezifizieren, sollen sinnvolle Defaults gesetzt werden. Weiter spielt die Fenstergrösse eine Rolle; sie muss veränderbar sein. Und zuletzt soll während der Navigation ein

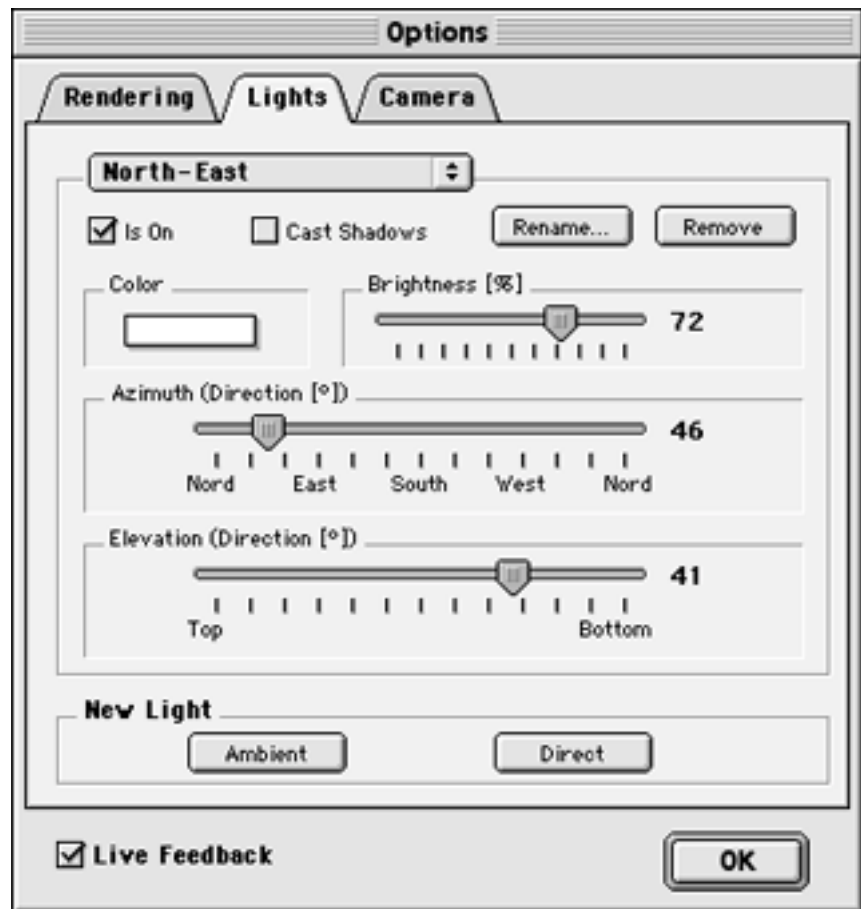


Abb. 3-4
Einstellung von
Beleuchtungsparametern
in Geo3D

automatisches Umschalten auf eine schnellere Rendering-Methode möglich sein.

Die QD3D-Parameter können in rein globale und überschreibbare eingeteilt werden. Rein globale Parameter beeinflussen immer das ganze Modell und können nicht überschrieben werden. Überschreibbare Parameter können für gewisse Komponenten überschrieben werden (objektorientierter Ansatz).

Rein globale Parameter sind:

- Verfügbare Renderer (bei Plug-in-Renderern meist zusätzliche, spezielle Parameter),
- Reines Software-Rendering vs. Hardwareunterstützung,
- Textur-Filtering (Darstellungsqualität allfälliger vorhandener Texturen),
- Hintergrundfarbe,
- Beleuchtung.

Bei den ersten vier Parametern kann aus zwei oder mehreren Varianten gewählt werden. Die Beleuchtung ist komplexer (Abb. 3-4): Es kann eine beliebige Anzahl Lichtquellen definiert werden. **Geo3D** soll also nicht nur die Parameter verändern, sondern auch bestehende Lichtquellen löschen oder

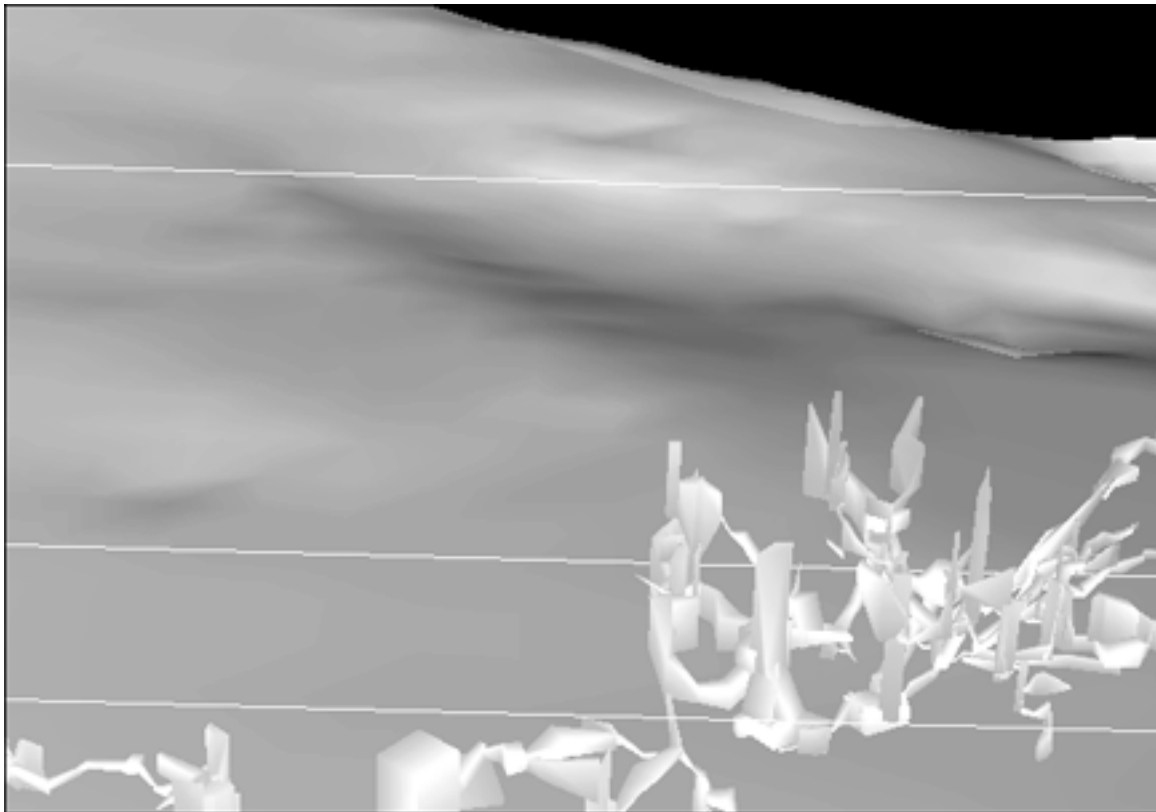


Abb. 3-5 Semitransparentes Geländemodell mit darunterliegendem Höhlensystem
(Datenquelle M. Sluka)

neue schaffen können. Die Wirkung dynamischer Lichtquellen soll durch einen mit der Kamera verbundenen «Scheinwerfer» erprobt werden. Für Geländemodelle wäre die alternative Eingabe der parallelen Beleuchtung statt über Azimut und Elevation über Tageszeit, Jahreszeit und Breitengrad sinnvoll (nicht implementiert). Dies würde zusätzliche Erkenntnisse über die potentielle Besonnung der Oberfläche erlauben.

Überschreibbare Parameter sind:

- Schattierung (Beleuchtungsmodell),
- Interpolation,
- Darstellung der Rückseiten von Flächen (*Backfacing*),
- Darstellungsstil (Punkte, Kanten, Flächen),
- Nebel (verschiedene Varianten mit eigenen Parametern),
- Materialkonstanten wie Farbe, Reflexionseigenschaften, Transparenz etc. (Abb. 3-5 zeigt den Einsatz von Transparenz),
- Tessellation (Unterteilung von gekrümmten Oberflächen/Kurven in Dreiecke/Segmente)

Materialkonstanten werden oft überschrieben (insbesondere die Farbe); die andern Parameter werden meist global verwendet.

Welche QD3D-Parameter benötigt werden, ist kontextabhängig. **Geo3D** muss in jedem Fall sicherstellen, dass die jeweils benötigten Parameter definiert sind – sinnvolle Default-Werte müssen vorhanden sein. Um Parameter effizient zu ändern, ist ein sofortiges visuelles Feedback unabdingbar. Dies gilt vor allem für komplexe Parameter wie die Beleuchtung. Alle QD3D-Parameter lassen sich über eine 3DMF-Datei importieren. In **Geo3D** können fast alle Parameter editiert werden. Nicht implementiert wurde, da für geographische Daten von geringerer Wichtigkeit, das Editieren der spezifischen Punkt- und Spotlichteinstellungen sowie der Materialkonstanten (werden meist überschrieben) und der Tessellation.

Wie erwähnt spielt nebst den QD3D-Parametern die Fenstergrösse für die Qualität eine entscheidende Rolle. Je grösser das Fenster, desto besser ist die Auflösung (und desto kleiner die Bildrate). Die Fenster sollen

- frei skalierbar sein,
- geordnet und an den Bildschirm angepasst werden können.
- Für Präsentationen soll die Menüleiste ausgeblendet werden können.
- Das gleichzeitige Verwenden mehrerer Bildschirme wird unterstützt. Für Präsentationen kann ein Fenster die Fläche aller vorhandenen Bildschirme ausfüllen.

Alle fensterbezogenen Features wurden implementiert.

Eine Möglichkeit, die Vorteile von gewünschter Rendering-Qualität und Rendering-Geschwindigkeit zu vereinen, ist das automatische Umschalten auf ein einfacheres Rendering-Verfahren während des Navigierens. Am effizientesten erweist sich hier die Wahl des schnellsten verfügbaren Renderers. Bei mehreren gekoppelten Ansichten soll nur das Hauptfenster neu gerendert werden können. Diese Funktionen wurden implementiert.

3.5 Analyse-Features

Im weitesten Sinn dienen die meisten Features auch für die visuelle Analyse von Modellen – sei es das durch die Kamera-Navigation mögliche freie Betrachten, seien es Beleuchtungs- und Rendering-Einstellungen. In diesem Teilkapitel werden besondere Mittel der visuellen und numerischen Analyse zusammengefasst.

3.5.1 Koordinatenwerte und Distanz

Die Weltkoordinaten eines geometrischen Modells aus dem geographischen Umfeld stimmen meist mit einem geographischen Koordinatensystem überein. Es werden zwei Features vorgeschlagen:

- Die 3-D-Koordinaten eines Punktes (x, y, z) auf der Oberfläche eines Objektes sollen durch das Positionieren der Maus auf der Abbildung des Punkts im 2-D-Fenster (x', y') zurückgegeben werden.
- Durch Setzen eines Fixpunktes auf einer 3-D-Oberfläche kann die Distanz zwischen Fixpunkt und einem beliebigen sichtbaren Punkt der Oberfläche bestimmt werden. Es wird die obige Methode verwendet. Die Distanz wird absolut und auf die einzelnen Koordinatenachsen bezogen angegeben. So können beispielsweise bei einem Modell mit Höhenwerten Höhendifferenzen ermittelt werden.

Beide Features wurden implementiert.

3.5.2 Geraden und Ebenen

Einfache geometrische Elemente wie Geraden und Ebenen, die als bewegliche Komponenten dem geometrischen Modell überlagert werden, können effiziente Mittel der visuellen Analyse sein – Lagebeziehungen können verdeutlicht werden. ZEDI (1998) zeigte dies für hauptebenenparallele Ebenen in VRML. Solche Elemente müssen ein- und ausblendbar sein und interaktiv mittels Maus und/oder Tastatur verschoben werden können. Frei einstellbare Transparenz erleichtert den Überblick (Abb. 3-6).

Implementiert wurden drei voneinander unabhängige hauptachsenparallele Ebenen mit freier Farb- und Transparenzwahl, die mittels Tastatur verscho-

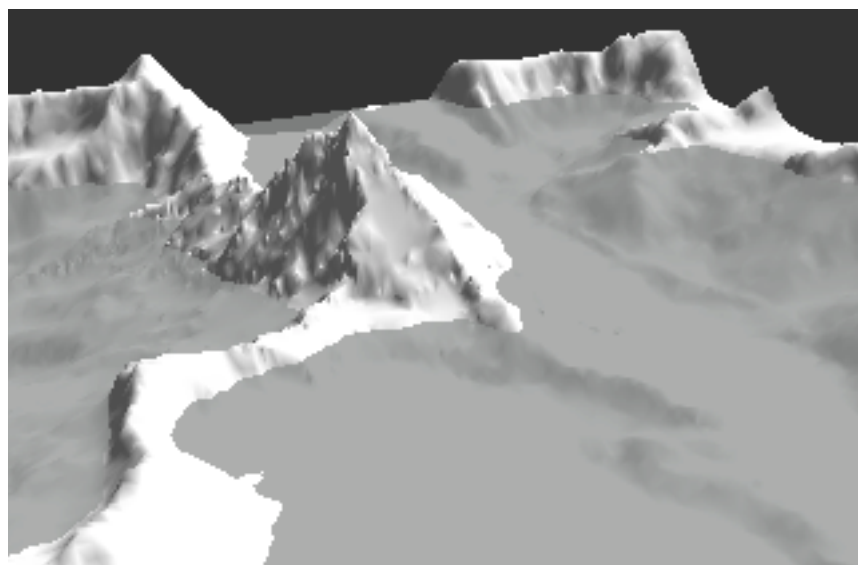


Abb. 3-6
Semitransparente
horizontale Ebene
(Datenquelle M. Heller)

ben werden und ein frei bewegliches Koordinatenachsenkreuz (Abb. 3-7), das ebenfalls über die Tastatur gesteuert wird.

Aus dem Bereich Speläologie/Geologie kam der Wunsch nach einer weiteren, völlig frei platzierbaren, Ebene. Diese Ebene kann durch die Wahl eines Punktes und der Falllinie bestimmt werden. Die Lage von geologischen Schichten könnte auf diese Weise angenähert werden.

3.5.3 Mehrfache Ansichten eines Modells

Mehrfache Ansichten des Modells erlauben Vergleiche aus verschiedenen Blickwinkeln und/oder mit verschiedenen Darstellungsparametern. Grundsätzlich gibt es hierzu zwei Möglichkeiten:

- Autonome Ansichten erlauben beliebige Einstellungen der Parameter.
- Gekoppelte Ansichten interagieren nach festgelegten Regeln.

Mehrere Modelle können nebeneinander geöffnet werden. Verschiedene autonome Ansichten eines Modells lassen sich auf gleichem Weg realisieren. Dieses Feature wurde implementiert.

Zusätzlich zur Hauptansicht sollen drei weitere, an die Hauptansicht gekoppelte Seitenansichten angeboten werden: Ansichten von oben, von vorne und von links. Die Blickrichtung dieser Darstellungen verläuft parallel zu den globalen Koordinatenachsen. Die Kameraposition befindet sich in einstellbarer Distanz zum Zentrum des Modells. Daraus folgt, dass die Kameraposition sich in einer Ebene, die normal zur jeweiligen Koordinatenachse liegt, bewegt. Die aktuelle Lage der Kameraposition in dieser Ebene wird durch die Kameraposition im Hauptfenster bestimmt. Die beiden nicht vordefinierten Koordinatenwerte der Kameraposition der Seitenansicht werden von der Kameraposition des Hauptfensters übernommen. Abb. 3-7 zeigt zwei dieser Seitenansichten. Mit Ausnahme der Kameraparameter sind die verschiedenen Ansichten identisch. Die Seitenansichten wurden in der beschriebenen Art implementiert. Die Verwendung gekoppelter Fenster verursacht selbstverständlich einen vervielfachten Rendering-Aufwand.

Hauptachsenparallele Darstellungen bieten sich unter anderem für weltkoordinatenbezogene Navigation an. Analog zur Navigation mit der Maus im Hauptfenster (Kap. 3.3.5) soll in den Seitenansichten Translationen in der jeweiligen Hauptebene, in der sich die Kameraposition bewegt, stattfinden können. Jeweils zwei der Translationsvarianten x-, y- und z-Translation (Tab. 3-1) werden der Maus zugeordnet. Diese Navigationsvariante wurde implementiert.

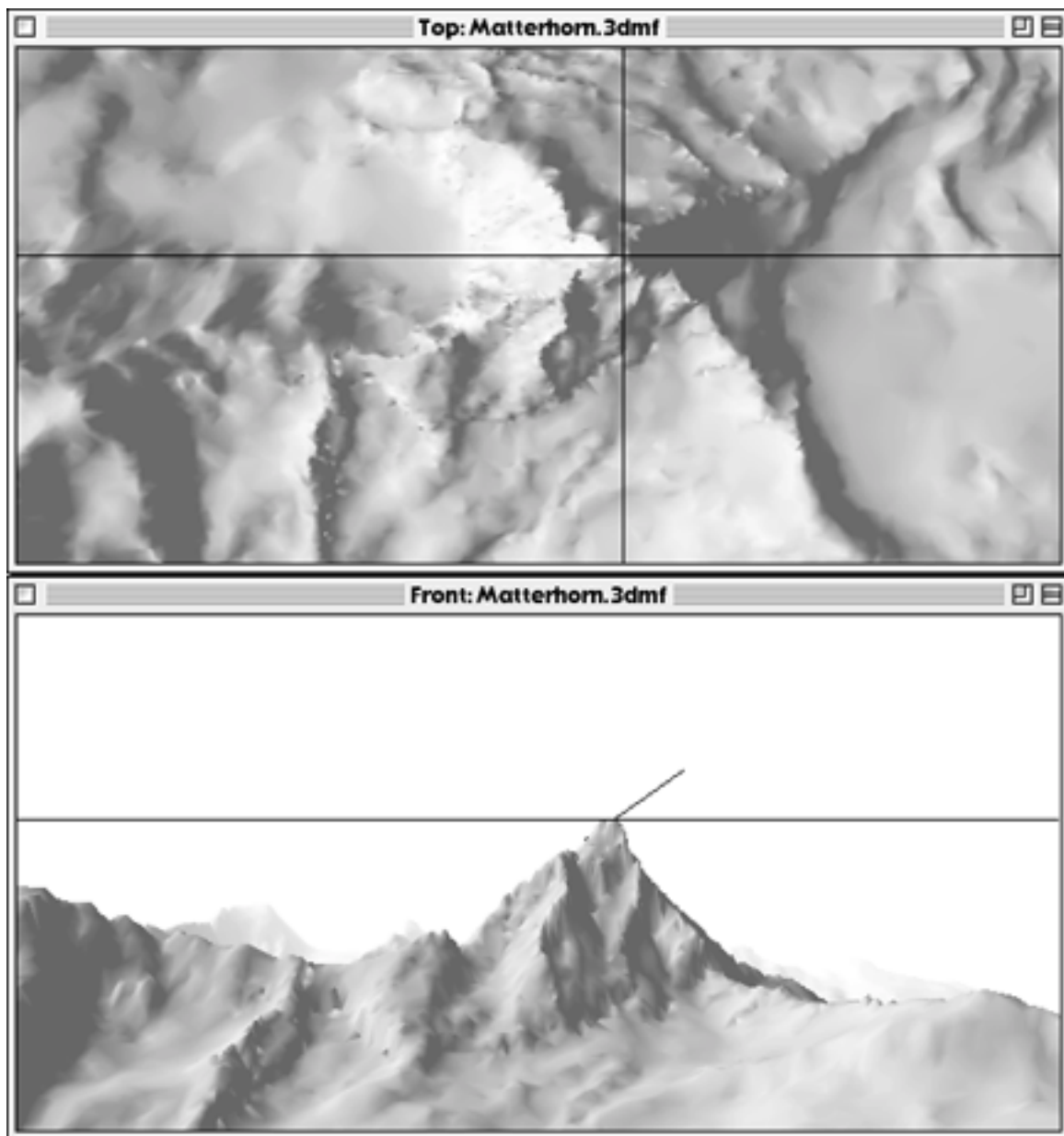


Abb. 3-7 Gekoppelte Seitenansichten in Geo3D: Das eingeblendete Koordinatenkreuz verdeutlicht die Lage (Datenquelle M. Heller)

3.5.4 Selektive Darstellung von Komponenten

Das selektive Ein- und Ausblenden einzelner Komponenten oder Gruppen von Komponenten (im folgenden Gruppen genannt) ergibt vielfältige Möglichkeiten, ein Modell zu analysieren:

- Geometrisch komplexe Modelle können in Teilmodelle zerlegt und dadurch visuell leichter erfasst werden.
- Komponenten können beispielsweise nach Regionen oder inhaltlicher Zugehörigkeit gruppiert werden.

- Verschiedene Varianten von Komponenten können in eine Datei verpackt und einzeln oder gemeinsam gesichert werden.

Zudem ist das Rendern von Teilmodellen schneller, was vor allem bei grossen Modellen ein Vorteil ist. Teilmodelle können zudem auf einfache Art extrahiert und in einer eigenen Datei gesichert werden.

Die Wahl sinnvoller Gruppen ist eine Aufgabe der Datenmodellierung. **Geo3D** stellt das Interface für die Selektion zu Verfügung. Die geometrischen Komponenten können beliebig gruppiert werden, wobei eine Komponente zu mehreren Gruppen gehören darf. **Geo3D** stellt ein Dateiformat zu Verfügung. Dieses beinhaltet die Gruppierungen wie auch deren Darstellung im Interface, konkret in einem Menü. Das Format «Gruppen» ist in Anhang D beschrieben. Zwei Selektionsmodi werden vorgeschlagen: Entweder kann jeweils nur eine Gruppe selektiert werden, oder beliebig viele Gruppen können eingeblendet werden. Die selektive Darstellung von Komponenten wurde implementiert.

3.5.5 Weitere Features

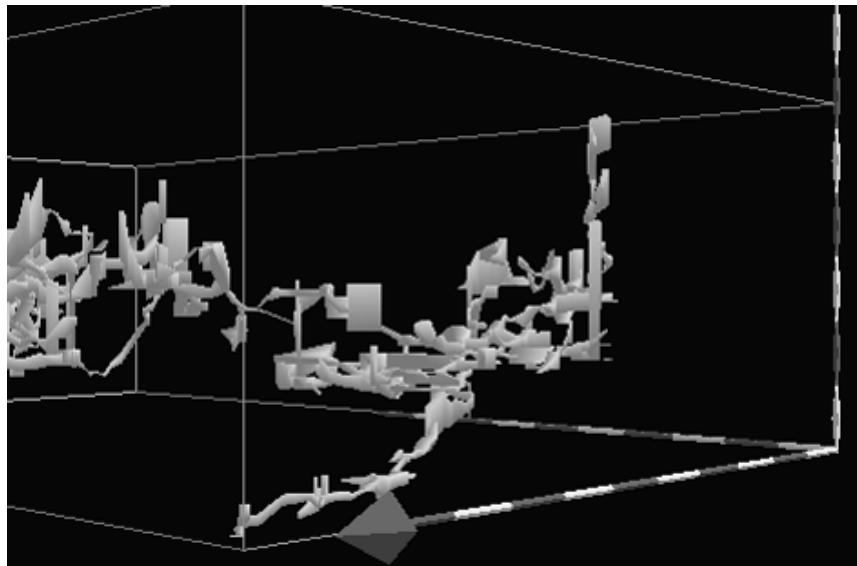
Überhöhung: Durch Dehnen und Stauchen des geometrischen Modells parallel zu den globalen Koordinatenachsen kann eine Dimension hervorgehoben oder auch korrigiert werden. Die Überhöhung von Geländemodellen ist beispielsweise ein gebräuchliches Mittel, geringe Unebenheiten plastischer darzustellen. (nicht implementiert)

Profile: Durch die Verschiebung der Front- und Backclipping-Plane der Kamera können Profile mit «Tiefenwirkung» erzielt werden. Statt dem ganzen Sichtbereich des geometrischen Modells wird ein frei bestimmbarer Tiefenbereich dargestellt. (nicht implementiert)

Orientierungshilfen: Künstliche Horizonte, Koordinatennetze, umhüllende Quader in Linienform, (Nord-)Pfeile, Massstäbe etc. können bei gewissen Modellen die Einordnung im Raum vereinfachen. Es muss aber berücksichtigt werden, dass bei interaktiver Visualisierung mit freier Wahl der Ansicht solche Objekte schwierig zu platzieren sind. Einerseits kann ein Objekt nicht oder kaum sichtbar sein, andererseits kann es die Sicht beeinträchtigen. **Geo3D** erlaubt mit Ausnahme der in Kap. 3.5.2 erwähnten Geraden und Ebenen keine interaktive Manipulation solcher Objekte – hingegen können natürlich beliebige Orientierungshilfen über eine 3DMF-Datei eingelesen werden. Abb. 3-8 zeigt ein Modell mit räumlich fixierten Orientierungshilfen.

Kamera-Lage: Die numerische und graphische Darstellung der Kamera-lage ist eine weitere Orientierungshilfe. Sie ist im Gegensatz zu den bis anhin erwähnten Hilfen modellunabhängig und kann einem eigenen Fenster oder einem Fensterbereich zugewiesen werden. Von der dominanten Ebene ausgehend kann die Blickrichtung mit Hilfe zweier Parameter wiedergegeben wer-

Abb. 3-8
Orientierungshilfen:
Nordpfeil, Massstab und
umhüllender Quader
(Datenquelle M. Sluka)



den: Azimut und Elevation. Ergänzend wird die Abweichung der globalen y-Komponente des Up-Vektor von der positiven y-Achse bestimmt, d.h. wie weit weicht sie von der senkrechten Richtung ab. Zu berücksichtigen ist, dass bei Parametern in gewissen Fällen nicht definierte Werte auftreten können. Der Azimut ist nicht definiert, wenn die Elevation 90° oder -90° beträgt. Die numerische Wiedergabe von Azimut und Elevation wurde implementiert.

3.6 Schnittstellen zu Applikationen und Dateien

Schnittstellen sind die unabdingbare Grundlage der Kommunikation zwischen Applikationen. Der Datentransfer erfolgt üblicherweise via definierte Dateiformate. Applikationen werden meist über Eingabewerkzeuge wie Maus und Tastatur bedient. Viele Applikationen können aber auch über andere Applikationen bedient werden (externe Bedienung). E-Mail-Applikationen können beispielsweise direkt aus Web-Browsern durch Klicken auf eine E-Mail-Adresse aufgerufen werden. Umfassende Visualisierungssysteme lassen sich über diesen Weg realisieren.

3.6.1 Import- und Exportformate

Geo3D unterstützt als 3-D-Dateiformat das QuickDraw-3D-eigene 3DMF-Format. Nebst dem Lesen einzelner 3DMF-Dokumente können auch weitere Dokumente in bereits geöffnete eingefügt werden. Die globalen QD3D-Parameter des einzufügenden Dokuments werden ignoriert. **Geo3D** kann offene 3DMF-Dokumente sichern und Kopien herstellen. Wahlweise kann das Dokument als editierbare Textdatei oder im kompakteren Binär-Format geschrieben werden. Die Unterstützung von 3DMF wurde vollständig implementiert. Weitere 3-D-Formate sind nicht vorgesehen – 3-D-Fremdformate

zu übersetzen ist ziemlich aufwändig, vor allem das Übersetzen von Fremdformaten ins Applikations-Format.

Um Ansichten festzuhalten, ist ein 2-D-Exportformat wünschbar. **Geo3D** exportiert Bilder in beliebiger Grösse und Auflösung im PICT-Format. PICT ist das traditionelle 2-D-Graphik-Format auf dem Mac. Nebst Einzelbildern können auch Bildsequenzen exportiert werden. Dabei wird ein ganzer Navigationspfad in Einzelbilder umgerechnet. Da das Mac OS mit *QuickTime* ausgezeichnete Werkzeuge zum Konvertieren von Bildern in andere Formate (*PictureViewer*) und von Bildsequenzen in digitale Videos (*QuickTime Player*) bereithält, wird auf weitere 2-D-Exportformate verzichtet.

Der Vollständigkeit halber sei hier noch einmal auf die proprietären Formate «Animation» (Kap. 3.3.4), das Navigationspfade aufzeichnet, und «Gruppen» (Kap. 3.5.4), das die Gruppen für die selektive Darstellung von Komponenten beinhaltet, verwiesen.

3.6.2 Hyperlinks

Geo3D soll es ermöglichen, zusätzliche Informationen an Komponenten eines geometrischen Modells anzubinden. Eine sehr flexible, modellunabhängige Lösung bietet das Internet mit seinem Adressen-Konzept an. Analog zu einem *Hypertext* können Komponenten eines geometrischen Modells in **Geo3D** mit einem *Hyperlink* versehen werden. Da nur die Adresse der Komponente zugeordnet wird, spielt das Format, in dem die Informationen abgelegt sind, keine Rolle. Die Verbindung zwischen **Geo3D** und dem Programm, das die Informationen verwerten/darstellen soll, übernimmt die Mac-Systemerweiterung *Internet Config*²².

Da Internetadressen meist wenig über den Inhalt aussagen, wird zusätzlich eine Kommentarzeile angeboten. Internetadressen samt Kommentar (im folgenden einfach Adressen genannt) können in einem 3DMF-Dokument einem Objekt zugewiesen werden. In **Geo3D** sollen Adressen geometrischen Komponenten zugewiesen, editiert und aufgerufen werden können. **Geo3D** unterstützt dabei beliebige Arten von Adressen. Zusätzlich werden relative lokale Adressen unterstützt²³. Der für relative Adressen notwendige Quellordner (Basisadresse und Pfad zum Ordner) kann frei gewählt werden. Die Selektion der geometrischen Komponente erfolgt durch Anklicken. Im Inter-

22. Internet Config (Internet Configuration System, <http://www.quinn.echidna.id.au/Quinn/Config>) wurde entwickelt, um die Voreinstellungen internetbezogener Applikationen an einem zentralen Ort zusammenfassen zu können. Unter anderem können sogenannte «Helper»-Applikationen für bestimmte Adressen-Präfixe (z.B. *http://*) festgelegt werden. Eine an Internet Config gesendete Internet-Adresse wird zur Helper-Applikation weitergeleitet und von dieser geöffnet.

23. Lokale Adressen haben den Präfix *file:///* und beziehen sich auf lokale Datenträger (Festplatten etc.). Eine absolute Adresse beschreibt den vollständigen Pfad von der Basisadresse zur Datei (*file:///HD/Geo3D/Manual/navigation.html*). Relative Adressen (*Manual/navigation.html*) beziehen sich auf einen kontextabhängigen Ordner (*file:///HD/Geo3D/*).

face muss ersichtlich sein, ob eine geometrische Komponente eine Adresse besitzt, und falls dies zutrifft, wie sie lautet. Wenn der Mauszeiger über einer Komponente, die eine Adresse enthält, ist, zeigt der Mauszeiger dies durch ein Symbol an. Die betroffene Komponente wird als Ganzes durch Aufleuchten sichtbar gemacht. Zudem wird im Fenster die Adresse angezeigt. Adressen können per *Drag-and-Drop* aus einem Web-Browser oder aus einer beliebigen Textdatei der Komponente zugewiesen werden. Daneben wird ein Editier- und ein Aufrufmodus implementiert. Im Editiermodus wird durch Doppelklick ein Editierdialog geöffnet. Adressen können hier eingegeben, editiert und gelöscht werden. Im Aufrufmodus wird durch Doppelklick die Adresse via Internet Config an die entsprechende Helperapplikation weitergereicht. Die Hyperlinks für Komponenten wurden in dieser Form implementiert.

3.6.3 Externe Bedienung von «Geo3D»

Da **Geo3D** als Visualisierungswerkzeug für Daten, die andernorts generiert werden, konzipiert ist, ist es notwendig, Schnittstellen zu definieren, die eine externe Steuerung von **Geo3D** erlauben. So soll beispielsweise die Kamera von einem andern Programm aus gesteuert werden können. Das auf dem Mac übliche Protokoll für die Kommunikation zwischen Applikationen heisst *Apple Events*²⁴. Die hier vorgeschlagenen Basis-Funktionen, die von einem andern Programm aufgerufen werden können, umfassen:

- Öffnen und schliessen von **Geo3D**,
- Einlesen einer 3DMF-Datei und Darstellung des Inhalts in einem Fenster,
- Sichern einer modifizierten Datei,
- Übergabe der variablen Kameraparameter und des Zentrums (für Modus «Zentrum»). Die Parameter können als absolute oder relative Werte übermittelt werden.
- Einlesen von Animationsdateien und flexibler Zugriff auf die eingelesenen Kameradaten.

Die aufgeführten Punkte erlauben, auf vielfältige Weise durch eine beliebige 3DMF-Datei zu navigieren, ohne dass direkt mit **Geo3D** gearbeitet werden muss. Diese Basis-Funktionen wurden, zusammen mit weiteren Funktionen (Gruppen, Modi etc.), implementiert.

24. Applikationen brauchen auf dem Mac u.a. Apple Events, um Dienstleistungen und Informationen von andern Applikationen zu verlangen oder um Dienstleistungen und Informationen als Antwort auf eine Anfrage anzubieten. Die sogenannte *Client Application* sendet einen Apple Event zur *Server Application*, die entweder eine Dienstleistung ausführt und/oder, falls verlangt, eine Antwort zurücksendet. Entscheidend für das Gelingen einer Anfrage ist, ob die Server-Applikation einen bestimmten Apple Event empfangen kann und ob sie die geforderte Dienstleistung oder Information anbietet. (*Apple Computer 1993*)

Grundsätzlich kann für jede Funktion der Applikation eine Apple-Event-Schnittstelle errichtet werden. Aus Zeitgründen wurde darauf verzichtet. In Kap. 4.3 wird anhand eines realisierten Beispieles gezeigt, welchen Nutzen die Kommunikation zwischen Applikationen haben kann.

4 «Geo3D» im Visualisierungsprozess

Die Philosophen haben die Welt nur verschieden interpretiert; es kommt aber darauf an, sie zu verändern.

KARL MARX

4.1 Einsatz und Grenzen von «Geo3D»

Der Einsatz von Visualisierungswerkzeugen variiert in unterschiedlichen Stadien des Forschungsprozesses. DiBIASE 1990 unterscheidet zwischen den Stadien Exploration, Bestätigung (*Confirmation*), Synthese und Präsentation. Die Art der Visualisierung verlagert sich dabei vom *Visual Thinking* im privaten Rahmen zur öffentlichen *Visual Communication*.

Geo3D kann grundsätzlich in allen Stadien seine Dienste leisten kann, wobei aber die in Kap. 2.1.3 erwähnten Modellvoraussetzungen zu erfüllen sind (potentielle Sichtbarkeit der Komponenten, an Hardware angepasste Modellgrösse, berücksichtigte kognitive Schwierigkeiten). Zudem kann **Geo3D** keine wirklich fotorealistische Darstellung bieten. Raytracing, Schlagschatten, Antialiasing etc. können allenfalls über Plug-in-Renderer verwirklicht werden – allerdings unter Verlust der interaktiven Navigation. Alle Einstellungsmöglichkeiten in **Geo3D** wirken auf das ganze Modell (Ausnahme: Internetadressen). Einzelne Komponenten können zwar in **Geo3D** ein- und ausgeblendet, aber nicht editiert werden. «Seeing the unseen» kann nur dann erfolgreich sein, wenn Möglichkeiten und Grenzen des Werkzeugs berücksichtigt werden.

Sind die Voraussetzungen erfüllt, kann **Geo3D** innerhalb seiner Möglichkeiten in allen Stadien mit Gewinn verwendet werden. Gerade aber im Bereich des *Visual Thinking* ist die schnelle, weitgehend automatisierte Generierung

der 3-D-Modelle besonders wichtig, da oft unzählige Varianten geprüft werden müssen. Kap. 4.3 beschreibt eine ausgereifte Lösung aus der Höhlenforschung unter Verwendung von **Geo3D**. Für die Visual Communication mag ein grösserer Aufwand in Kauf genommen werden. Schnell hergestellte Vorschau-Bilder sind jedoch wünschenswert.

Um die Möglichkeiten des Einsatzes von **Geo3D** zu konkretisieren, seien hier ausgewählte Anwendungen kurz erläutert. Die Liste ist keinesfalls abschliessend. Es werden nur implementierte Features berücksichtigt.

Überblick: Durch die freie Wahl der Ansichten können Daten schnell überblickt werden. Dazu ist der Navigationsmodus «Center» besonders geeignet, auch in Form vordefinierter Sequenzen (Kap. 3.3.3). Nützlich sind auch Mehrfachansichten.

Präsentation: iGeo3D bietet die Möglichkeit an, 3-D-Daten zu präsentieren. Genügt die verfügbare Rendering-Qualität den Ansprüchen, können die Animationsfeatures (Kap. 3.3.4) eingesetzt werden. Interaktive Präsentation ist aber mit ein wenig Übung durchaus realisierbar. Da **Geo3D** statische (durch Weiterverarbeitung auch dynamische) 2-D-Ansichten exportieren kann, ist z. B. der Einsatz in der Präsentationsvorbereitung möglich.

Optimaler Betrachtungs-Standort²⁵: Die vielen Navigationsmöglichkeiten bieten sich für die Suche an.

Beleuchtung: Beleuchtung ist wichtig für das Wahrnehmen der Räumlichkeit einer Darstellung. Sie kann auch für die visuelle Analyse eingesetzt werden. Interaktive Veränderung der Beleuchtung ist ein Mittel, Oberflächenstrukturen zu finden (z. B. Artefakte der Geländemodellierung). Die Besonnung eines Geländes kann visualisiert werden. Durch den Einsatz verschiedenfarbiger, horizontal in allen vier Himmelsrichtungen angeordneter Lichtquellen kann die Exposition eines Geländes visualisiert werden.

Studium der geometrischen Struktur: Die geometrische Struktur kann mittels Drahtgitterdarstellung oder Ausschalten der Interpolation studiert werden.

Geeignete Darstellungsmethoden: Geeignete Darstellungsmethoden für bestimmte Arten von 3-D-Modellen können geprüft werden. Das Wahrnehmen der Oberflächenstruktur eines Geländemodells kann z. B. durch einen eindimensionalen Farbverlauf in Abhängigkeit der Höhe verbessert werden.

Modellvergleiche: Durch Überlagerung von geometrischen Modellen können Abweichungen sichtbar gemacht werden. Durch Gruppierung können die Modelle einzeln oder gemeinsam dargestellt werden. Verschiedene

25. Sieber 1996 untersucht in seiner Arbeit, welche Kriterien oder Elemente einer Darstellung für Wahl eines optimalen Standorts essentiell sind.

Methoden der Geländemodellierung können verglichen werden. Der Einsatz von Transparenz kann hier sinnvoll sein.

Distanzmessungen: Durch Distanzmessungen im 3-D-Raum können beispielweise Höhendifferenzen auf einfache Art bestimmt werden.

Externes Rendering: Vorschaubilder für aufwändige statische Visualisierung durch spezielle Applikationen erstellen (z.B. Fotorealismus) ist zeitsparend. Die Kameradaten können für diesen Zweck importiert und exportiert werden.

Animationspfade: Geo3D eignet sich für die Suche und Aufzeichnung eines geeigneten Animationspfades. Dieser kann entweder direkt in Geo3D für Präsentationen, oder als Vorschau für aufwändige dynamische Visualisierung verwendet werden. Pfade können ebenso importiert und exportiert werden.

Einfaches Informationssystem: Der Aufbau eines einfachen Informationssystems wird durch Verknüpfung von externen Daten mit geometrischen Objekten ermöglicht (Kap. 3.6.2). Das Modell «Erdmann» auf der beiliegenden CD-ROM ist ein einfaches Beispiel. Diverse Links zu Fotografien auf dem Internet (<http://www.geo.unizh.ch/spelaion/Dinkelberg/Erdmann/>) sind eingebettet.

«Geo3D» als externes Visualisierungs-Tool: Geo3D kann als externes Visualisierungs-Tool verwendet oder direkt in ein erweitertes Visualisierungssystem integriert werden. Kap. 4.3 befasst sich mit dieser Option.

4.2 Zur Visualisierung von Geländemodellen

Die meisten Geländemodelle liegen entweder als rechteckiges Grid oder als TIN (Triangulated Irregular Network; PEUCKER/FOWLER/LITTLE/MARK 1978) vor. TINs haben gegenüber Grids zwei Vorteile: Erstens kann das Gelände bei gleichwertiger Auflösung mit weniger Punkten repräsentiert werden, und zweitens ist die Datenstruktur (ein Netz von Dreiecken) eindeutig. Grids dagegen bestehen aus einem Netz von Rechtecken. Die Rechtecke müssen für das Rendering in Dreiecke zerlegt werden²⁶. Wird die Zerlegung automatisch durchgeführt, entsteht eine im Detail irreführende Darstellung. Die *Trigrid*-Primitive von QD3D wird z.B. nach einem «Zickzack»-Muster zerlegt.

Sind die Geländemodelle für die Visualisierung auf Lowend-Computern zu gross – was im Fachbereich Geographie oft der Fall ist – können die Daten durch Verfahren wie *Adaptive Traingular Mesh (ATM) Filtering* (HELLER

26. Ohne diese Triangulation wäre die 3-D-Darstellung eines Grid «klötzchenförmig». Eine differenzierte Schattierung wäre nicht möglich und jeder Punkt müsste durch vier Punkte repräsentiert werden.

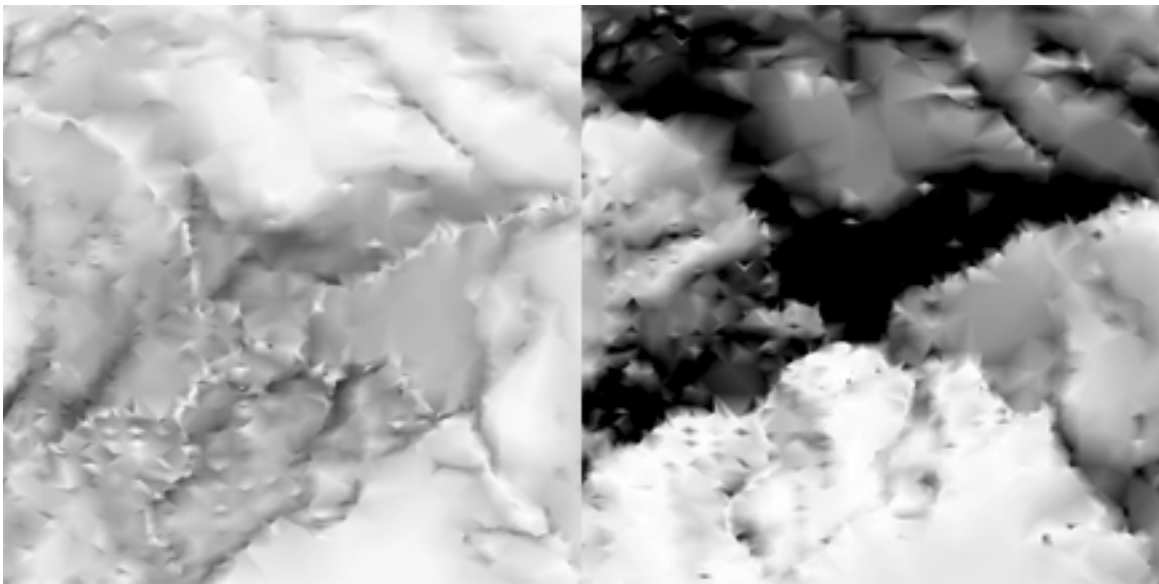


Abb. 4-1 Einfluss der Lichtrichtung auf die Wahrnehmung der Geländeform. Einfallswinkel links 90°, rechts 45° bei gleicher Lichtstärke (Datenquelle M. Heller)

1990) reduziert werden. Daneben bringen rein technische Optimierungen grosse Beschleunigung. Statt einzelner Dreiecke können z.B. geeignete Polygonnetze (Kap. 2.3.2) verwendet werden.

Geeignete Beleuchtung, bez. die daraus resultierende Schattierung ist äusserst wichtig für die Wahrnehmung der Geländeform (Abb. 4-1). Flächen, die auf der Schattseite liegen, erscheinen in einheitlicher Farbe oder, falls das Gelände texturiert ist, als abgedunkelte Textur. Dieser Effekt ist auch auf Luftbildern zu sehen. Leuchtet die parallele Lichtquelle von oben, ist die gesamte sichtbare Oberfläche differenziert schattiert. Die Schattierung widerspiegelt die Hangneigung (hell bis dunkel: flach bis steil). Bei Schräglichtbeleuchtung kann durch den Einsatz weiterer schwächerer Lichtquellen die Schattseite leicht aufgehellt werden. Die interaktiven Einstellungsmöglichkeiten der Beleuchtung in **Geo3D** (siehe Abb. 3-4) sind ein effizientes Mittel, die Beleuchtung zu optimieren.

Ein vertikaler Farbverlauf (Textur) bietet zusätzliche Hilfe, die Geländeform wahrzunehmen. Der Farbverlauf sollte, um die Schattierung nicht zu beeinträchtigen, eine einheitliche Helligkeit aufweisen. Der Einsatz von Nebel hilft, die Tiefenwahrnehmung zu verbessern. Weiter entfernte Flächen nähern sich dabei der Hintergrundfarbe an.

4.3 «Toporobot» und «Geo3D»

Die Applikation *LimeLight* ist der Kern von Martin Hellers Höhlenkartographie- und -visualisierungssystem *Toporobot* (<http://www.geo.unizh.ch/~heller/toporobot/>, HELLER 1994). Toporobot wurde seit 1972 fortlaufend weiterentwickelt und gilt seit langem als State-of-the-Art in diesem Bereich.

Durch die enge Zusammenarbeit mit Martin Heller wurden **Geo3D** und *LimeLight* über Schnittstellen verbunden. Dabei werden seitens **Geo3D** die in Kap. 3.6 besprochenen Schnittstellen verwendet. Diese Schnittstellen sind aber keineswegs speziell auf *LimeLight* zugeschnitten, sondern allgemeiner Art und könnten von jeder Datenmodellierungsapplikation genutzt werden. Die Kommunikation zwischen **Geo3D** und *LimeLight* ist dreiteilig:

- *LimeLight* generiert für **Geo3D** geometrische Modelle mit verschiedenen Attributen.
- *LimeLight* kann **Geo3D** starten, verschiedene Dateien einlesen lassen und die Kamera steuern.
- Durch Picken von Komponenten in **Geo3D** werden in *LimeLight* die zugehörigen Daten aufgerufen und tabellarisch dargestellt.

Geometrische Modelle können im einfachsten Fall durch einen Menübefehl in *LimeLight* generiert und automatisch in **Geo3D** eingelesen werden. Geometrische Modelle lassen sich aber auch an spezifische Bedürfnisse anpassen. Geometrische Auflösung, Modellierungsalgorithmen wie auch rendering-bezogene Parameter können verändert werden. *LimeLight* bietet vielfältige strukturbezogene Farbattributierung und Gruppierung (Kap. 3.5.4) der Komponenten. Beliebige Internet-Adressen können an Komponenten angefügt werden.

LimeLight bietet eigene Visualisierungsmöglichkeiten an. Über einen 2-D-Plan des Höhlenmodells in *LimeLight* kann die Kamera von **Geo3D** gesteuert werden. Ausgangspunkt und Ziel lassen sich interaktiv steuern, parallele Verschiebung der Kamera ist ebenso möglich wie strukturbezogene Navigation entlang von Gängen. Abb. 4-2 zeigt die Steuerung der **Geo3D**-Kamera über den 2-D-Plan von *LimeLight*. Zwischen der Navigation über den 2-D-Plan und direkter Navigation in **Geo3D** kann beliebig hin und her gewechselt werden.

LimeLight generiert für jede Komponente der 3DMF-Datei eine spezielle Internetadresse, die auf die Daten der einzelnen Komponente in *LimeLight* verweist (Präfix `speleo:///`). Durch Picken einer Komponente in **Geo3D**, der eine Adresse vom Typ `speleo` zugewiesen wurde, werden in den Datenfenstern von *LimeLight* die der Komponente zugrunde liegenden Daten tabellarisch angezeigt.

Es hat sich gezeigt, dass die beschriebenen Interaktionsmöglichkeiten zwischen einer Modellierapplikation und einem 3-D-Visualisierungs-Tool die 3-D-Visualisierung wesentlich erleichtern. Da die Modellierapplikation 3-D-

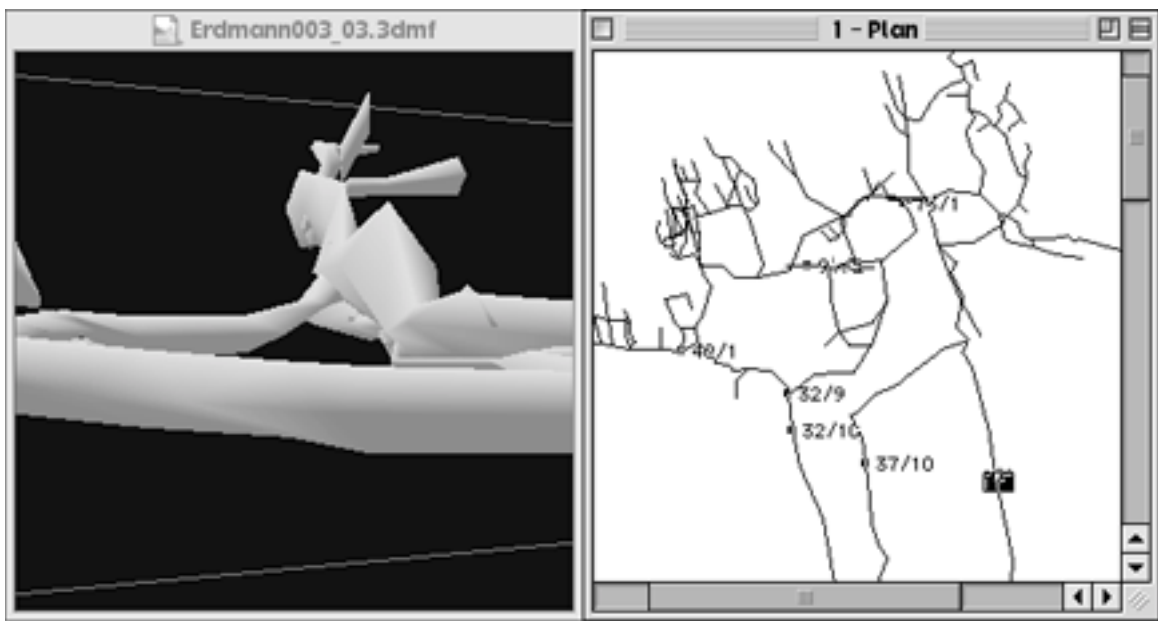


Abb. 4-2 Steuerung der Geo3D-Kamera auf dem 2-D-Plan von LimeLight (Datenquelle K. Zimmermann)

Daten generieren und exportieren kann, wird die 3-D-Visualisierung sehr einfach. Sie wird zum alltäglichen Werkzeug, das nach Bedarf eingesetzt wird. Ebenso ist die Verzahnung verschiedener Visualisierungsmöglichkeiten über Applikationsgrenzen hinaus problemlos machbar (hier Steuerung der 3-D-über die 2-D-Darstellung). Der Zugriff auf die dem 3-D-Modell zugrunde liegenden Daten löst ein weiteres Problem: Die durch die Visualisierung gewonnenen Erkenntnisse können direkt in die Datenmodellierung einfließen. Die Identifikation der Komponenten entfällt – die Fehleranfälligkeit sinkt.

5 Schlussfolgerungen und Ausblick

5.1 Fazit

Die für diese Arbeit formulierten Ziele (Kap. 1.2) lauten kurzgefasst: Die Möglichkeiten interaktiver 3-D-Visualisierung auf Lowend-Computern werden durch einen Viewer mit Schwerpunkt Navigation erweitert. Die für räumliche geographische Daten notwendigen Features werden in einem Konzept zusammengestellt. Wesentliche Teile des Konzepts werden stabil implementiert und fortlaufend veröffentlicht.

Die Konzeption und Implementation von **Geo3D** kann (mit gewissen Einschränkungen) in verschiedener Hinsicht als erfolgreich betrachtet werden. Das Konzept beinhaltet eine Reihe von derzeit auf Lowend-Computern realisierbaren Features. **Geo3D** ist stabil implementiert und frei von (mir bekannten) Fehlern, das User-Interface ist laut verschiedenen Usern ohne allzu grossen Lernaufwand nachvollziehbar und die Funktionalität, soweit implementiert, ausgereift. Eine vollständige Implementation aller vorgestellten Features hätte den Rahmen dieser Arbeit gesprengt, wird aber nach Abschluss der Arbeit ins Auge gefasst. Einschränkend ist Tatsache, dass **Geo3D** anhand von Höhlen- und Geländemodellen entwickelt wurde. Weitere Modelle aus andern Fachbereichen wären wünschenswert. Und nur im Bereich der Höhlenmodellierung war das anregende und bestätigende Feedback genügend gross²⁷. Während die implementierte Funktionalität von **Geo3D** für die Visualisierung von Höhlenmodellen optimal ist, fehlen für Geländemodelle zwei sehr nützliche (im Konzept enthaltene) Features: Kollisionserkennung zwischen Kamerastandort und Modell, und oberflächenabhängige Navigation.

27. Interessanterweise bekam ich sonst vor allem aus Architekturkreisen Hinweise auf die Eignung von **Geo3D** als Visualisierungswerkzeug, was ich wegen mangelnden Kenntnissen in diesem Fachbereich nur beschränkt auswerten konnte.

Räumliche Daten sind im Fachbereich Geographie zuhauf vorhanden. Dass die Möglichkeiten der interaktiven 3-D-Visualisierung dennoch eher selten genutzt werden, liegt unter anderem am Fehlen geeigneter Schnittstellen zwischen Modellier- und Visualisierungsapplikationen. Je einfacher 3-D-Modelle hergestellt werden können, desto öfter werden sie auch eingesetzt. In der Praxis ist eine Visualisierungsapplikation auf Applikationen, die 3-D-Modelle exportieren, angewiesen. Die Erfahrungen mit LimeLight (Kap. 4.3) zeigen, dass reine Exportfunktionen nicht voll befriedigen können. Die enge Verzahnung von Applikationen zur Datenmodellierung und -visualisierung erweist sich hingegen als zukunftssträchtig. Die in **Geo3D** verfolgte Strategie, Visualisierung möglichst allgemein zu formulieren – geeignet für Modelle verschiedener Art – erlaubt ähnliche Interaktionen mit einer Vielzahl von Applikationen.

Gerade geographische Datensätze sind oft sehr umfangreich. Trotz gewaltiger Leistungssteigerungen der Hardware liegt die maximale Zahl der interaktiv darstellbaren Dreiecke auf Lowend-Computern noch im fünfstelligen Bereich. Interessanterweise spielt neben der Rendering-Qualität (Kap. 3.4) ebenso sehr die Topologie des geometrischen Modells eine Rolle. Gelände-modelle verlangen tendenziell mehr Rechenleistung als die «löchrigen» Höhlenmodelle. Bei Höhlenmodellen reduzieren im Durchschnitt Clipping und Tiefensortierung (Kap. 2.2.2) den Rechenaufwand stärker.

Geo3D wurde schon in einem frühen Entwicklungsstadium übers Internet der Öffentlichkeit zur Verfügung gestellt. Öffentlichkeit im Internet entsteht jedoch nur durch Vernetzung. Erst mit gewisser «Werbung» wurden Links von gut frequentierten Websites zur **Geo3D**-Site erstellt. Dadurch stellte sich das erwünschte Feedback ein. Der jetzige Bekanntheitsgrad von **Geo3D** wurde v.a. durch die weite Verbreitung von Toporobot und Ankündigungen auf Mac-spezifischen News-Sites erreicht. **Geo3D** leistet inzwischen einen Beitrag zur Verbreitung von 3-D-Anwendungen im (nicht nur) geographischen Umfeld.

QuickDraw 3D erwies sich als ein mächtiges Werkzeug für den dreidimensionalen Teil der Programmierung. Besonders wertvoll habe ich die hochentwickelten Methoden, Dateien zu lesen und zu schreiben, die einfach zu bewerkstelligende zweidimensionale Darstellung auf dem Bildschirm, die flexiblen und schnellen Picking-Methoden und das gute Angebot von mathematischen Methoden im Bereich der Vektoren- und Matrizen-Operationen empfunden. Gute Kenntnisse der Grundlagen der 3-D-Computergraphik sind auch bei hochentwickelten 3-D-Architekturen wie QD3D unabdingbar.

5.2 Ausblick

Zweifellos wird die Leistungssteigerung der Computer-Hardware auch in Zukunft in hoher Geschwindigkeit weitergetrieben. Spezielle 3-D-Hardware gehört inzwischen bei Lowend-Computern zum Standard. Mehr noch: Die 3-D-Leistung durchschnittlicher Lowend-Computer ist in letzter Zeit überproportional gesteigert worden.

Softwareseitig herrscht hingegen zur Zeit eine eigentümliche Stille. VRML – vor zwei Jahren als das 3-D-Metaformat der Zukunft gepriesen – hat sich nicht im erwarteten Mass durchgesetzt. *Extensible 3D* (X3D) als Nachfolgerin von VRML ist noch weit davon entfernt, auch nur testweise implementiert zu werden²⁸. Die Zukunft von QuickDraw 3D (und damit von 3DMF) ist mehr als fraglich²⁹. Darauf reagierend begannen verschiedene Software-Entwickler *Quesa*³⁰ zu entwickeln. Quesa soll zu QD3D 1.6 vollständig kompatibel werden. Silicon Graphics und Microsoft arbeiten gemeinsam an *Fahrenheit*³¹, einem neuen 3-D-API (*Application Programmer's Interface*). Doch die Entwicklung von Fahrenheit kommt nur langsam voran. Die Zukunft der bestehenden Technologien der an Fahrenheit beteiligten Firmen ist von dieser Entwicklung abhängig. Einzig *Java 3D*³², einem neuen High-Level-API für Java, ist bereits auf gewissen Plattformen erhältlich. Ob Java 3D das Leistungsniveau vergleichbarer APIs erreichen wird, ist zumindest fraglich.

Die Konzeption und Entwicklung von **Geo3D** basiert auf dem Vergleich von bestehenden Applikationen und v. a. auf Vorschlägen und Ideen einer kleinen Gruppe von Leuten. Die Testdaten aus dem geographischen Umfeld waren auf Höhlensysteme und digitale Geländemodelle beschränkt. Die Vor- und Nachteile interaktiver 3-D-Visualisierung gegenüber herkömmlichen Methoden müssten für verschiedene Arten von Daten und Anwendungen abgeklärt werden. Dazu wäre es wünschbar, dass 3-D-Modelle aus weiteren Gebieten generiert würden. Ich denke da an Geologie, Meteorologie, Klimatologie, Glaziologie, aber auch an die Darstellung abstrakter Daten aus der Statistik³³.

28. Das Web3D-Consortium beschreibt den Entwicklungsprozess von X3D unter http://www.vrml.org/technicalinfo/x3d/x3d_process.htm und http://www.web3d.org/technicalinfo/x3d/x3d_process.htm.

29. Apple Computer hat QuickDraw 3D kürzlich aus der Carbon-Spezifikation (http://developer.apple.com/techpubs/macosx/Carbon_Specification/CarbSpecWebIntro.html) gestrichen. Das Carbon-API erlaubt es, Applikationen zu schreiben, die zu Mac OS 8/9 wie auch zu Mac OS X, dem geplanten neuen Betriebssystem von Apple, kompatibel sind. Da QuickDraw 3D Bestandteil von QuickTime (<http://www.apple.com/quicktime/>), dem derzeitigen Industriestandard für digitales Video, ist, bleiben einige Fragen zur Zukunft offen.

30. Siehe <http://quesa.designcommunity.com/>. Quesa ist ein Open-Source-Projekt. Es ist zur Zeit nicht absehbar, wann Quesa fertiggestellt sein wird.

31. Siehe <http://www.sgi.com/fahrenheit/>. Die Informationen sind allerdings veraltet.

32. Siehe <http://java.sun.com/products/java-media/3D/>. Zu Java 3D und VRML siehe <http://www.web3d.org/WorkingGroups/vrml-java3d/>

33. Um den Bogen weiter zu spannen: Ich habe über das Internet viele 3-D-Modelle aus Architektur, Chemie, Biologie und Mathematik gefunden. Viele Visualisierungsprobleme sind vermutlich nicht fachspezifisch.

Fragen bezüglich notwendigen Visualisierungs-Features und zugehörigem User-Interface müssten breit abgestützt untersucht werden. Die Features sollten klassiert und soweit abstrahiert werden, dass sie möglichst breit nutzbar sind. Eine Herausforderung ist auch die Datenkonvertierung. Geländemodelle aus einem GIS liessen sich beispielsweise relativ einfach in ein beliebiges 3-D-Datenformat konvertieren. Doch ohne entsprechende Exportfunktionen oder Konvertierungstools wird interaktive 3-D-Visualisierung nicht zur alltäglichen Methode. In Zukunft muss das Augenmerk auf die Verzahnung von Datenmodellierung und -visualisierung im interaktiven 3-D-Bereich gerichtet werden.

Der Schwerpunkt von **Geo3D** liegt in der Bewegungsdynamik, dem Navigieren der Kamera durch den Raum. Folgende Features sind bei einer Weiterentwicklung des **Geo3D**-Konzepts vorrangig zu implementieren: Objektanimation und Objektdeformation.

Objekte beliebiger Art sollen interaktiv im Raum platziert und animiert werden können. Dazu kurz zwei Vorschläge von **Geo3D**-Benützern: Geologische Schichtgrenzen könnten explorativ gezogen werden; die Besonnung eines Geländes könnte tages- oder jahreszeitlich simuliert werden.

Interaktive Deformation von geometrischen Komponenten in 3-D würde neue Anwendungen erschliessen. WEIBEL/HELLER (1990, S. 226) haben das «interactive sculpting» von Geländemodellen vorgeschlagen. BÄR 1996 hat Werkzeuge zur interaktiven Bearbeitung von Geländeoberflächen entwickelt, allerdings in 2-D. Sein Ansatz könnte um eine Dimension erweitert werden.

Quellen

Literatur³⁴

AMES, A. L./NADEAU, D. R./MORELAND, J. L. 1997: *VRML 2.0 Sourcebook*. New York: Wiley (2nd ed.)

APPLE COMPUTER 1993: *Inside Macintosh: Interapplication Communication*. Reading MA: Addison-Wesley

APPLE COMPUTER 1995: *3D Graphics Programming With QuickDraw 3D*. Reading MA: Addison-Wesley

APPLE COMPUTER 1995a: *Macintosh Human Interface Guidelines*. Reading MA: Addison-Wesley (auch <http://developer.apple.com/techpubs/macos8/HumanInterfaceToolbox/HumanInterfaceGuide/humaninterfaceguide.html>)

APPLE COMPUTER 1996: *QuickDraw 3D Preliminary User Interface Guidelines*. Cupertino CA: Apple Computer (nicht mehr erhältlich; siehe CD-ROM)

APPLE TECHNICAL PUBLICATIONS 1997: *Mac OS 8 Human Interface Guidelines*. Cupertino CA: Apple Computer (<http://developer.apple.com/techpubs/macos8/HumanInterfaceToolbox/HumanInterfaceGuide/humaninterfaceguide.html>)

APPLE TECHNICAL PUBLICATIONS 1997a: *3D Graphics Programming With QuickDraw 3D 1.5.4. Including 3D Metafile Reference and Renderer Acceleration Virtual Engine*. Cupertino CA: Apple Computer (<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/PDF.htm>)

34. Internet-Quellen in elektronischer Buchform werden unter Literatur aufgeführt. Das für solche Dokumente übliche PDF-Format kann mit dem frei erhältlichen *Acrobat Reader* (<http://www.adobe.com/products/acrobat/readermain.html>) gelesen und gedruckt werden. Die auf das Dokument verweisende Internet-Adresse wird zusätzlich angegeben. Die Adressen wurden anfangs Oktober 1999 geprüft. Alle hier aufgeführten, über das Internet erhältlichen QD3D-Dokumente sind auch auf der beiliegenden CD-ROM zu finden.

- BÄR, H. 1996: *Interaktive Bearbeitung von Geländeoberflächen*. Konzepte, Methoden, Versuche. Geoprocessing Series, Vol. 25, Geographisches Institut, Universität Zürich
- BUI-TUONG, PHONG 1975: Illumination for Computer-Generated Images. In: *Communications of the ACM*, Vol. 18, No. 6, S. 311-317
- DI BIASE, D. W. 1990: Scientific Visualization in the Earth Sciences. In: *Earth and Mineral Sciences*, Vol. 59, No. 2, S. 13-18
- DIERKES, R./LUCE, S./GREENSTONE, B. 1999: *QuickDraw 3D 1.6*. New API Features and Improvements. Cupertino CA: Apple Computer
(<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/PDF.htm>)
- DUDEN 1997: *Fremdwörterbuch*. Mannheim/Wien/Zürich: Dudenverl. (6. Aufl.)
- DÜNOW, J. 1999: Pixelkunst. 3D-Grafik zwischen Prozessor und Monitor. In: *c't*, 2/99, S. 130-135
- DÜNOW, J. 1999a: Polygon-Treibsätze. Hardware-Ansätze zur 3D-Geometriebeschleunigung. In: *c't*, 2/99, S. 188-195
- FALCO, P./MCBRIDE, P. 1996: Generating QuickTime VR Movies From QuickDraw 3D. In: *develop, The Apple Technical Journal*, issue 25, Cupertino CA: Apple Computer, S. 5-25 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)
- FERNICOLA, P./THOMPSON, N. 1995: A New Dimension for Macintosh Graphics. In: *develop, The Apple Technical Journal*, issue 22, Cupertino CA: Apple Computer, S. 6-28 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)
- FERNICOLA, P./THOMPSON, N. 1996: The Basic of QuickDraw 3D Geometries. In: *develop, The Apple Technical Journal*, issue 23, Cupertino CA: Apple Computer, S. 30-51 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)
- FLOHR, U. 1996: 3-D for Everyone. In: *BYTE*, October 1996 (auch <http://www.byte.com/art/9610/9610.htm>)
- FLUSSER, V. 1992: Virtuelle Räume – Simultane Welten. Vilém Flusser im Gespräch mit Sabine Kraft und Phillipp Oswalt. In: *Arch+*, Heft 111, März 1992, S. 33-52
- FOLEY, J. D./VAN DAM, A./FEINER, S. K./HUGHES, J. F. 1996: *Computer Graphics, Principles and Practice*. Reading MA: Addison-Wesley (2nd ed. in C)
- FOLEY, J. D./VAN DAM, A./FEINER, S. K./HUGHES, J. F./PHILLIPS R. L. 1994: *Grundlagen der Computergraphik. Einführung, Konzepte, Methoden*. Bonn: Addison-Wesley

- GREENSTONE, B. 1998: *Making Cool QuickDraw 3D Applications!* Cupertino CA: Apple Computer
(<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/PDF.htm>)
- GOURAUD, H. 1971: Computer Display of Curved Surfaces. In: *IEEE Transactions on Computers*, Vol. C-20, No. 6, S. 623-629
- HELLER, M. 1990: Triangulation Algorithms for Adaptive Terrain Modeling. In: *Proceedings Forth International Symposium on Spatial Data Handling*, Zürich, Vol. 1, S. 163-174
- HELLER, M. 1994: Toporobot, Logiciel d'aide à la topographie. In: *XX Congrès de la Fédération Française de Spéléologie - (1992)*, Saint Gaudens, Spélunca Mémoires 21, S. 75-81
- KUECHLE, S. 1997: Optimizing QuickDraw 3D 1.5.3 Applications For Maximum Performance. *Technote 1109*, Cupertino CA: Apple Computer
(<http://developer.apple.com/technotes/tn/tn1109.html>)
- MAC EACHREN, A. M. in collaboration with BUTTENFIELD B. P./CAMPBELL J. B./DiBIASE D. W./MONMONIER M. 1992: Visualization. In: Abler, R. F./Marcus M. G./Olson J. M. (eds.), *Geography's Inner Worlds. Pervasive Themes in Contemporary American Geography*. New Brunswick NJ: Rutgers University Press, S. 99-137
- MCCORMICK, B. H./DEFANTI T. A./BROWN, M. D. 1987: *Visualization in scientific computing*. Report to the National Science Foundation by the panel on Graphics, Image Processing, and Workstations. Baltimore: ACM SIGGRAPH Computer Graphics, Vol. 21, No. 6
- PEUCKER, T. K./FOWLER R. J./LITTLE J. J./MARK D. M. 1978: The Triangulated Irregular Network. In: *Proceedings of the ASP Digital Terrain Models (DTM) Symposium*, Falls Church VA, S. 516-540
- SCHNEIDER, P. J. 1996: New QuickDraw 3D Geometries. In: *develop, The Apple Technical Journal*, issue 28, Cupertino CA: Apple Computer, S. 32-56
(auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)
- SCHNEIDER, P. J. 1996a: NURB Curves: A Guide for the Uninitiated. In: *develop, The Apple Technical Journal*, issue 25, Cupertino CA: Apple Computer, S. 48-74 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)
- SIEBER, R. 1996: *Visuelle Wahrnehmung dreidimensionales parametrisierter Objekte und Objektgruppen*. Eine empirische Untersuchung zur Bestimmung eines optimalen Betrachterstandorts. Geoprocessing Series, Vol. 26, Geographisches Institut, Universität Zürich
- SUTER, M. 1997: *Aspekte der interaktiven real-time 3D-Landschaftsvisualisierung*. Remote Sensing Series, Vol. 29, Remote Sensing Laboratories, University of Zurich

THOMPSON, N. 1997: Easy 3D With the QuickDraw 3D Viewer. In: *develop, The Apple Technical Journal*, issue 29, Cupertino CA: Apple Computer, S. 4-27 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)

THOMPSON, N./FERNICOLA, P./DAVIDSON, K. 1996: Adding Custom Data to QuickDraw 3D Objects. In: *develop, The Apple Technical Journal*, issue 26, Cupertino CA: Apple Computer, S. 80-98 (auch <http://devworld.apple.com/dev/techsupport/develop/bysubject/quickdraw3d.html>)

THOMPSON, N./KUECHLE, S. 1998: Supporting Plug-in Renderers in QuickDraw 3D 1.5.3 Applications. *Technote 1110*, Cupertino CA: Apple Computer (<http://developer.apple.com/technotes/tn/tn1110.html>)

THOMPSON, T. 1996: Must-See 3-D Engines. In: *BYTE*, June 1996 (auch <http://www.byte.com/art/9606/9606.htm>)

WEIBEL, R./ HELLER M. 1990: A Framework for Digital Terrain Modeling. *Proceedings Forth International Symposium on Spatial Data Handling*, Zürich, Vol. 1, S. 219-229

ZEDI, R. 1998: *Dynamische und interaktive Darstellung geographischer Objekte*. Diplomarbeit. Geographisches Institut der Universität Zürich (auch <http://www.geo.unizh.ch/~zedi/diplom/>)

Internet³⁵

Acrobat Reader: <http://www.adobe.com/products/acrobat/readermain.html>

Adobe Systems: <http://www.adobe.com>

Apple Computer: <http://www.apple.com>

Bare Bones Software: <http://web.barebones.com>

Carbon-Spezifikation: http://developer.apple.com/techpubs/macosx/Carbon_Specification/CarbSpecWebIntro.html

Cosmo Player: <http://www.cosmosoftware.com>

Erdmannshöhle: <http://www.geo.unizh.ch/spelaion/Dinkelberg/Erdmann/>

Extensible 3D (X3D): http://www.vrml.org/technicalinfo/x3d/x3d_process.htm,
http://www.web3d.org/technicalinfo/x3d/x3d_process.htm

Fahrenheit: <http://www.sgi.com/fahrenheit/>

Fly Through: <http://www.halcyon.com/jennings/FlyThrough.html>

35. Die im Text erwähnten Internet-Adressen werden hier unter dem betreffenden Stichwort aufgelistet. Die Adressen wurden anfangs Oktober 1999 geprüft.

Geo3D: <http://www.datacomm.ch/luzern/>

Geo3D-Manual (deutsch): <http://www.datacomm.ch/luzern/D/Manual/>

GeoCon: <http://www.datacomm.ch/luzern/>

Internet Config: <http://www.quinn.echidna.id.au/Quinn/Config/>

Java 3D: <http://java.sun.com/products/java-media/3D/>

Java 3D und VRML: <http://www.web3d.org/WorkingGroups/vrml-java3d/>

Inklination: <http://www.inklination.com>

LightWork Design: <http://www.lightwork.com>

LimeLight: <http://www.geo.unizh.ch/~heller/toporobot/>

Mathemaesthetics: <http://www.mathemaesthetics.com>

Metrowerks: <http://www.metrowerks.com>

Microsoft: <http://www.microsoft.com>

Netscape Communications: <http://home.netscape.com>

Pangea Software: <http://www.pangeasoft.net>

Qualcomm: <http://www.qualcomm.com>

Quesa: <http://quesa.designcommunity.com>

QuickDraw-3D-API: <http://developer.apple.com/sdk/>

QuickTime: <http://www.apple.com/quicktime/>

QuickTime 3 Animation and 3D File Formats:
<http://www.apple.com/quicktime/authoring/ffan.html>

Stairways Software: <http://www.stairways.com>

Toporobot: <http://www.geo.unizh.ch/~heller/toporobot/>

WorldView: <http://www.worldview.com>

VRML: <http://www.vrml.ch>, <http://www.vrml.org>, <http://www.web3d.org>

Anhang

A Inhalt der CD-ROM

Die beiliegende CD-ROM (Oktober 1999) enthält die vorliegende Arbeit («Geo3D.pdf») im *Adobe Portable Document Format* (PDF). Im Ordner «Geo3D» befindet sich das Programm **Geo3D** (Version 1.5.2) in deutscher, französischer und englischer Sprache (<http://www.datacomm.ch/luzern/>). Ein detailliertes deutschsprachiges Manual im HTML-Format erläutert alle Features (<http://www.datacomm.ch/luzern/D/Manual/>). Der Ordner «Modelle» enthält die in den Abbildungen verwendeten Höhlenmodelle im 3DMF-Format. In «Programme/Systemerweiterungen» sind erstens die aktuellsten Versionen der Systemerweiterungen, die von **Geo3D** verlangt werden, und zweitens die Programme zum Lesen der verschiedenen Dokumente dieser CD, enthalten. «QD3D-Dokumentation» enthält u.a. die unter der Literatur aufgeführten, nur über das Internet erhältlichen Dokumente zum Thema QuickDraw 3D.

Die CD-ROM ist hybrid (Mac/ISO 9660). Die vorliegende Arbeit und die QD3D-Dokumente sind plattformunabhängig. Die andern Dateien sind nur auf einen Mac lesbar.

B Systemanforderungen von Geo3D

Geo3D verlangt einen Computer mit PowerPC-Prozessor (Power Mac); optional, aber empfehlenswert ist eine 3-D-Beschleunigerkarte. Mac OS ab Version 8.5 plus QuickDraw 3D 1.6 ist die optimale Umgebung. Mindestens verlangt wird Mac OS ab Version 7.5.1 plus Appearance Extension («Erscheinungsbild-Erweiterung») Version 1.0.3 oder Mac OS ab Version 8. Weiter wird QD3D ab Version 1.5.4 und für die Internetfunktionen Internet Config ab Version 2 benötigt. Mac OS bis Version 7.5.5 sowie alle Erweiterungen sind kostenlos erhältlich.

C Verwendete Software

Die folgenden Programme wurden für diese Arbeit verwendet (nur gesichtete und getestete Software werden nicht aufgeführt):

Betriebssystem:

Mac OS. Cupertino CA: Apple Computer (<http://www.apple.com>)

Programmierung:

CodeWarrior Pro. Austin TX: Metrowerks (<http://www.metrowerks.com>)

Resorcerer. Boulder CO: Mathemaesthetics (<http://www.mathemaesthetics.com>)

Text und 2-D-Graphik:

FrameMaker. San Jose CA: Adobe Systems (<http://www.adobe.com>)

Photoshop. San Jose CA: Adobe Systems (<http://www.adobe.com>)

Acrobat. San Jose CA: Adobe Systems (<http://www.adobe.com>)

BBEdit. Bedford MA: Bare Bones Software (<http://web.barebones.com>)

3-D-Visualisierung:

Toporobot. Zürich: M. Heller (<http://www.geo.unizh.ch/~heller/toporobot/>)

3DMF Optimizer. Austin TX: Pangea Software (<http://www.pangeasoft.net>)

FineArt 3D. Vancouver BC: Inklination (<http://www.inklination.com>)

Kommunikation:

Eudora Pro. Qualcomm (<http://www.qualcomm.com>)

Anarchie Pro. Stairways Software (<http://www.stairways.com>)

Navigator. Netscape Communications (<http://home.netscape.com>)

Internet Explorer. Microsoft (<http://www.microsoft.com>)

Folgende Software wurde im Rahmen dieser Arbeit entwickelt:

3-D-Visualisierung:

Geo3D, GeoCon, Anatas (<http://www.datacomm.ch/luzern/>)

D Proprietäre Geo3D-Dateiformate

Gruppen (Groups): Die Datei muss den Mac-Filetyp «TEXT» aufweisen. Der Dateiname sollte mit dem Suffix «.grp» enden. Zur Identifizierung muss am Anfang der Datei ##Geo3D_Groups stehen. Kommentare beginnen mit «#». Ein Kommentar wird implizit durch das Zeilenende beendet. Der Befehl «Menuitem» beginnt mit !M. Auf der gleichen Zeile folgt der Name, der ins Menü eingetragen wird. Er hat die Form "Name". Anschliessend folgen die Nummern der Objekte. Die Zahlen können auf verschiedenen Zeilen stehen und müssen durch einen Leerschlag voneinander getrennt werden (z.B. 3 6 10 11). Aufeinanderfolgende Objekte können mit «-» zusammengefasst werden (z.B. 13-16). Der Befehl wird implizit durch den folgenden Befehl oder das Dateiende abgeschlossen. Eine Trennlinie im Menü wird durch !S (Separator) erreicht. Abb. A-1 zeigt eine Groups-Datei.

Abb. A-1
Beispiel für das
Dateiformat «Gruppen»:
Die Datei wurde für das
Höhlenmodell «Cach-
tice» von *M. Sluka* in
LimeLight erstellt.

```
##Geo3D_Groups
!S
!M "Surface" 1
!M "Surface with Boxes" 1 -3
!M "Surface with Caves" 1 4-180
!S
!M "Cachtice" 2 4-174
!M "Stepnica" 3 175-180
!S
!M "Exploration 1956-57" 4-52 57-63 75-98 103-120 123-124
!M "First Surveying 1960" 4-6 19 21-22 78-79 88-92 95-96 103-114
!M "Explorations 1978-1999" 53-56 64-74 99-102 121-122 125-174
!M "Box" 2
```

Animation: Die Datei muss den Mac-Filetyp «TEXT» aufweisen. Der Dateiname sollte mit dem Suffix «.geo» enden. Zur Identifizierung muss am Anfang der Datei `##Geo3D_Animation` stehen. Kommentare beginnen mit «#». Ein Kommentar wird implizit durch das Zeilenende beendet. Kommentare werden beim Lesen der Datei ignoriert. Die Kameraparameter stehen jeweils auf einer Zeile und sind durch Leerschläge oder Tabulatoren getrennt. Abb. A-2 zeigt einen Ausschnitt aus einer von **Geo3D** exportierten Animations-Datei.

```
##Geo3D_Animation
# File: Sichle.geo
# Scene: Sichle.3dmf
# Creation date: 1.6.1999, 18:28 Uhr
# Number of Frames: 258
# Data: Camera Location x/y/z, Point Of Interest x/y/z, Up Vector x/y/z, Field Of View [°]
184.10 2609.84 -477.07 -6679.42 180.29 -1987.93 -0.32 0.95 -0.07 17.76 # 1
180.77 2608.66 -477.80 -6682.75 179.11 -1988.66 -0.32 0.95 -0.07 17.76 # 2
177.43 2607.48 -478.54 -6686.09 177.93 -1989.40 -0.32 0.95 -0.07 17.76 # 3
174.10 2606.30 -479.27 -6689.42 176.75 -1990.13 -0.32 0.95 -0.07 17.76 # 4
169.10 2604.53 -480.37 -6694.42 174.98 -1991.23 -0.32 0.95 -0.07 17.76 # 5
164.09 2602.76 -481.47 -6699.42 173.21 -1992.33 -0.32 0.95 -0.07 17.76 # 6
159.09 2600.99 -482.57 -6704.43 171.44 -1993.43 -0.32 0.95 -0.07 17.76 # 7
152.42 2598.63 -484.04 -6711.10 169.08 -1994.90 -0.32 0.95 -0.07 17.76 # 8
145.75 2596.27 -485.51 -6717.77 166.72 -1996.37 -0.32 0.95 -0.07 17.76 # 9
139.09 2593.90 -486.98 -6724.44 164.35 -1997.84 -0.32 0.95 -0.07 17.76 # 10
132.42 2591.54 -488.44 -6731.10 161.99 -1999.30 -0.32 0.95 -0.07 17.76 # 11
125.75 2589.18 -489.91 -6737.77 159.63 -2000.77 -0.32 0.95 -0.07 17.76 # 12
[...]
```

Abb. A-2 Beispiel für das Dateiformat «Animation»

