

Norbert Bartelme

gis Technologie

Geoinformationssysteme,
Landinformationssysteme
und ihre Grundlagen



Springer-Verlag Berlin Heidelberg GmbH



Norbert Bartelme

GIS Technologie

Geoinformationssysteme,
Landinformationssysteme
und ihre Grundlagen

Mit 126 Abbildungen

Springer-Verlag Berlin Heidelberg GmbH

Dr. NORBERT BARTELME
Abteilung für Mathematische
und Datenverarbeitende Geodäsie
Technische Universität Graz
Rechbauerstr. 12
A-8010 Graz

ISBN 978-3-540-50410-8 ISBN 978-3-662-07494-7 (eBook)
DOI 10.1007/978-3-662-07494-7

CIP-Titelaufnahme der Deutschen Bibliothek. Bartelme, Norbert: GIS-Technologie : Geoinformationssysteme, Landinformationssysteme und ihre Grundlagen / Norbert Bartelme. – Berlin ; Heidelberg ; New York ; London ; Paris ; Tokyo : Springer, 1988.

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der Fassung vom 24. Juni 1985 zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Springer-Verlag Berlin Heidelberg 1989
Ursprünglich erschienen bei Springer-Verlag Berlin Heidelberg New York 1989

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

2132/3130-543210 – Gedruckt auf säurefreiem Papier

VORWORT

Der Einsatz digitaler Methoden zur Verwaltung, Darstellung und Auswertung ortsbezogener Informationen erschließt neue Wege für ein tieferes Verständnis der komplexen Zusammenhänge, welche unseren Lebensraum prägen. Entsprechende Informationssysteme werden uns als Werkzeuge für den Vergleich raumbezogener Phänomene, für das Aufzeigen von Verflechtungen und Abhängigkeiten und für die sich daraus ergebenden Entscheidungen auf politischer, administrativer und wirtschaftlicher Ebene zur Verfügung stehen. Es ist zu erwarten, daß raumbezogene Informationssysteme schon in naher Zukunft jenen Stellenwert einnehmen werden, der heute bereits alphanumerischen Datenbanken und deren Verarbeitung eingeräumt wird. Die Anschaulichkeit graphischer Darstellungen und die sich ständig verbessernden technischen Voraussetzungen für deren Realisierung werden bewirken, daß die Bedeutung von Geo- und Landinformationssystemen weiter sprunghaft anwachsen wird.

Die Realisierung der einzelnen Komponenten solcher Systeme kann jedoch - aus vielerlei Gründen - nur in kleinen Schritten erfolgen; dies gilt gleichermaßen für die Daten, die Werkzeuge zu deren Bearbeitung und die Hardware. Die Qualität der Gesamtlösung setzt daher ein klares allumfassendes Konzept voraus, das die einzelnen Komponenten und deren Verknüpfungen definiert. Teile aus diesem logischen Gesamtkonzept können dann zu verschiedenen anwendungsspezifischen Realisierungen zusammengefügt werden. Deren Spektrum reicht von Aufgaben des Katasters, die sich durch Detailreichtum und Punktgenauigkeit auszeichnen, bis zu großräumigen rasterorientierten Anwendungen im Rahmen der Geowissenschaften und der Umweltplanung.

In den vergangenen sechs Jahren hat sich - bedingt durch meine Vorlesungen zu den Gebieten Geoinformationswesen, Digitalkartographie und Graphische Datenverarbeitung - eine Fülle von Material angesammelt. Praktische Erfahrungen kamen im Rahmen meiner Mitarbeit am System KERN - INFOCAM hinzu. Das vorliegende Buch ist das Ergebnis der Sichtung, Ordnung und Sammlung all dieser Erfahrungen. Ich habe mich bemüht, es so allgemein zu halten, daß damit ein breiter Leserkreis angesprochen wird. So trägt es dem Umstand Rechnung, daß die neuen Methoden nicht nur innerhalb der einzelnen Fachbereiche, sondern auch interdisziplinär neue, ungeahnte Perspektiven eröffnen.

Graz, im Herbst 1988.

Norbert Bartelme

INHALTSVERZEICHNIS

KAPITEL 1: EINFÜHRUNG

1

1.1.	Karten und Pläne: Hintergründe und Voraussetzungen	2
1.2.	Raumbezogene Informationssysteme: Begriffe und Gliederungen	5
1.3.	Die Vorteile der digitalen Speicherung	8
1.4.	Die Daten eines raumbezogenen Informationssystems	10
1.5.	Das Instrumentarium eines raumbezogenen Informationssystems	13
1.6.	Beispiele für raumbezogene Informationssysteme	15

KAPITEL 2: GEOMETRIE UND TOPOLOGIE

17

2.1.	Die Modellbildung	18
2.2.	Vektor- und Rastermodelle: eine Gegenüberstellung	20
2.3.	Die Geometrie und Topologie in Vektormodellen	26
2.3.1.	Der Punkt	26
2.3.2.	Die Kante	27
2.3.3.	Die Metrik in einer Kanten-Knoten-Struktur	30
2.3.4.	Flächen und Inseln	31
2.3.5.	Analytisch beschreibbare Flächen	35
2.3.6.	Netze	36
2.3.7.	Die Geometrie und Topologie von Symbolen	39
2.3.8.	Die Geometrie und Topologie von Texten	41
2.3.9.	Zusammengesetzte geometrisch-topologische Strukturen	42
2.3.10.	Geometrisch-topologische Bearbeitungsauszüge	43
2.4.	Die Geometrie und Topologie in Rastermodellen	44
2.5.	Hybride Modelle	47
2.6.	Fraktale Modelle	49

KAPITEL 3: THEMATIK

59

3.1.	Begriffsbestimmungen und Gliederungen	60
3.2.	Die Thematik in Rastermodellen	66
3.3.	Die Thematik in Vektormodellen	70
3.4.	Thematische Hierarchien	76
3.5.	Thematische Attribute	80
3.6.	Hybride Modelle der Thematik	82
3.7.	Thematik und topologische Konsistenzbedingungen	84

KAPITEL 4: LANGZEITSPEICHERUNG

89

4.1.	Datenbanken: Definitionen und Ziele	90
4.2.	Datenbank-Schemata	92
4.3.	Beziehungen zwischen Datengruppen	94
4.4.	Schlüssel, Attribute und Zugriffspfade	95
4.5.	Das hierarchische Datenbankmodell	99
4.6.	Das Netzwerkmodell	100
4.7.	Das relationale Datenbankmodell	101
4.8.	Voraussetzungen für stabile Datenstrukturen	103
4.8.1.	Die erste Normalform	104
4.8.2.	Die zweite Normalform	106
4.8.3.	Die dritte Normalform	107
4.9.	Die relationale Algebra	108
4.9.1.	Elementaroperationen der relationalen Algebra	108
4.9.2.	Sprachelemente für den Zugriff auf eine relationale Datenbank	111
4.10.	Die Einbindung in ein Anwenderprogramm	115
4.11.	Relationale Datenbanken und Anwenderkomfort	116

KAPITEL 5: TRANSAKTIONEN, RAUMBEZUG UND**LOKALE SPEICHERUNG**

119

5.1.	Raumbezug und Langzeitspeicherung: ein Gegensatz?	120
5.2.	Speicherung von Rasterstrukturen	123
5.3.	Speicherung von Vektorstrukturen	126
5.3.1.	Baumstrukturen	128
5.3.2.	Directory-orientierte Methoden	131
5.3.3.	Verallgemeinerung auf Linien- und Flächenobjekte	134
5.4.	Speicherung nach topologischen Kriterien	139
5.5.	Datenbanken und lokale raumbezogene Methoden	140
5.5.1.	Variante A: die ganzheitliche Lösung	141
5.5.2.	Variante B: Trennung von Geometrie und Thematik	143
5.5.3.	Variante C: Raumbezogene Transaktionen	144
5.5.4.	Andere Varianten	146

KAPITEL 6: WERKZEUGE

147

6.1.	Übersicht	148
6.2.	Geometrische Transformationen	149
6.2.1.	Drehung, Skalierung und Verschiebung	149
6.2.2.	Homogene Koordinaten	150
6.2.3.	Perspektive	152
6.3.	Schnittaufgaben	154
6.3.1.	Relative Lage von Punkten und Kanten	154
6.3.2.	Relative Lage zweier Kanten	155
6.3.3.	Verschneidung mit Polygonen	156

6.4.	Rasteroperationen	157
6.4.1.	Übersicht	157
6.4.2.	Elementare Operationen	158
6.4.3.	Verdicken und Verdünnen	160
6.4.4.	Füllen vorgegebener Flächen	162
6.4.5.	Abstandstransformierte und Skelett	163
6.4.6.	Filterung mittels Konvolution	164
6.4.7.	Filterung mittels Fourier-Transformation	168
6.4.8.	Geometrische Transformationen von Rasterdaten	169
6.4.9.	Effizienz von Rasteralgorithmen	169
6.5.	Interpolation und Approximation von Kurven	171
6.5.1.	Modellbildung	171
6.5.2.	Interpolation	173
6.5.3.	Approximation	177
6.6.	Interpolation und Approximation von Flächen	179
6.6.1.	Modellbildung (digitales Geländemodell)	179
6.6.2.	Lineare und bilineare Interpolation	183
6.6.3.	Gleitende Mittelbildung	187
6.6.4.	Lineare Prädiktion	189
6.6.5.	Interpolation über finite Elemente	190
6.6.6.	Dreiecksvermaschung	192
6.6.7.	Interpolation von Schichtenlinien in einem DGM	194
6.7.	Generalisierung	195
6.8.	Weitere Werkzeuge	197
6.8.1.	Ausgleichsprobleme	197
6.8.2.	Pufferzonen und Freistellungsflächen	198
6.8.3.	Automatische Flächenbildung und Aggregieren	199
6.8.4.	Netz- und Flußoptimierung	199
6.8.5.	Mustererkennung	199
6.8.6.	Karten- und Plangestaltung	200
6.8.7.	Umwandlung von Datenformaten	200

KAPITEL 7: DIE SICHT DES ANWENDERS

201

7.1.	Das raumbezogene Informationssystem als "black box" für den Anwender	202
7.2.	Anwenderprofil und Tabellen	203
7.2.1.	Die Zugriffsberechtigung	204
7.2.2.	Thematische Hierarchien	205
7.2.3.	Thematische Konsistenzvorgaben	205
7.2.4.	Wertebereichsvorgaben	206
7.2.5.	Genauigkeitsvorgaben	206
7.2.6.	Numerische Formate	207
7.2.7.	Sprachen und Dialekte	208
7.2.8.	Graphische Formate	208
7.2.9.	Dialoggestaltung	209
7.2.10.	Standardeinstellungen	210
7.3.	Datenselektion und Projektbildung	211
7.4.	Projektbeschreibung und Projektparameter	215
7.5.	Multi-User-Betrieb	216
7.6.	Netze und verteilte Datenbanken	217
7.7.	Benützerspezifische Funktionen	219

KAPITEL 8: AUSBLICKE	223
8.1. Künstliche Intelligenz: von der Information zur Koordination	224
8.2. Formen der künstlichen Intelligenz	226
8.3. Expertensysteme	227
8.4. Wissensgestützte Informationssysteme	231
8.5. Graphentheoretische Konzepte: UND-ODER-Graphen	235
8.6. Der kognitive Zugang: Semantische Netze und Frames	236
8.7. Derzeitige und künftige AI-Anwendungen in raumbezogenen Informationssystemen	239
8.7.1. Erkennen von Strukturen	239
8.7.2. Datenkonsistenz	240
8.7.3. Kartengestaltung	241
8.7.4. Natürliche Sprache	242
8.7.5. Anwenderprofil	243
8.7.6. Training	244
8.7.7. Anwenderspezifische Techniken	244

ANHANG A: LITERATUR

245

ANHANG B: GLOSSAR

255

ANHANG C: SACHVERZEICHNIS

273



KAPITEL 1
EINFÜHRUNG

1.1. Karten und Pläne: Hintergründe und Voraussetzungen

Karten und Pläne vermitteln ein Bild unserer Umwelt. In diesem Bild stellen topographische Merkmale, Gebäude und Verkehrswege ein Bezugssystem dar, das uns die Orientierung erleichtert. Darauf aufbauend werden eine Reihe von Themen behandelt, wie etwa die Aufteilung in Verwaltungseinheiten, ein Netz von Versorgungsleitungen, die Verteilung von Lagerstätten, aber auch infrastrukturelle Gegebenheiten und ökologische Themen. Wir können diese Themen miteinander in Beziehung setzen; der Raumbezug als gemeinsamer Nenner aller Themen ermöglicht es uns, vielfältige Verflechtungen und Wechselwirkungen zu erkennen und diese Erkenntnisse zur Grundlage unserer Entscheidungen zu machen.

Zwischen den beiden Begriffen "Karte" und "Plan" gibt es keine allgemein anerkannte Trennlinie; meist wird jedoch die Karte als Darstellung großräumiger Gebiete verstanden (etwa Gemeinden, Regionen, Bezirke, Länder), in denen geographische, geologische oder infrastrukturelle Themen behandelt werden. Der Plan hingegen umfaßt ein beschränktes Gebiet und erreicht dort ein hohes Maß an Genauigkeit, wie etwa im Katasterbereich, bei Leitungsplänen usw. Im englischen Sprachgebrauch trifft man für beide Kategorien die Begriffe "map" und "mapping" an. Wir sind der Meinung, daß sehr viele Probleme und Lösungsansätze - speziell bei grundsätzlichen Fragen, und diese sollen ja in diesem Buch aufgezeigt werden - auf beide Bereiche zutreffen, so daß wir im folgenden wieder auf eine Differenzierung verzichten können.

In den letzten Jahren hat die Datenverarbeitung auch in dem Bereich, der uns im folgenden interessieren soll - wie auch in vielen anderen Bereichen - neue Perspektiven eröffnet, und so werden die uns vertrauten Begriffe "Karte" und "Plan" immer häufiger in neue Wortschöpfungen eingebunden. Es entstanden Begriffe wie

- *digitale Karte,*
 - *digitaler Plan,*
 - *Landinformationssystem (LIS),*
 - *Geoinformationssystem (GIS),*
 - *raumbezogenes Informationssystem (RIS),*
 - *Mehrzweck- und Leitungskataster,*
- um nur einige Begriffe zu nennen.

Dabei kommt es nicht nur auf das Ersetzen* herkömmlicher manueller Zeichenmethoden durch computer-gesteuerte Zeichengeräte (Plotter) an; die Inhalte, die zum Entstehen einer Karte Anlaß geben, werden langfristig als Daten eines Informationssystems abgelegt und können so bei Bedarf mit anderen Daten zu immer neuen Kombinationen, Auswertungen und Vergleichen herangezogen werden. Zum Aspekt der Darstellung

kommen somit die - langfristig viel bedeutsameren - Aspekte der **Evidenzhaltung und Konsistenz** der Informationen eines solchen Systems hinzu. Die Funktionen dieses Systems müssen beiden Aspekten Rechnung tragen: nach *innen* hin müssen sie die Konsistenz wahren, und nach *außen* hin müssen sie sich nach jener Vorgangsweise richten, die wir bei der manuellen Erstellung und Auswertung einer Karte beachten. Wir wählen aus der Vielzahl der Darstellungs- und Interpretationsmöglichkeiten jene aus, die der jeweils zugrundeliegenden **Thematik** am besten entsprechen. Ein Stadtplan, eine Wanderkarte, die Wetterkarte im Fernsehen, die graphische Wiedergabe von Gewässergütezonen rufen in uns unterschiedliche Assoziationen hervor; beinahe unbewußt verwenden wir die jeweilige Thematik als Kriterium für die Art der Informationen, das Maß der Genauigkeit und für das Spektrum von möglichen Auswertungen.

Wir übersehen in diesem Zusammenhang oft, wie groß der Anteil der dabei notwendigen **Abstraktionsprozesse** ist, die von der Realität eines thematischen Sachverhaltes bis zu seiner systematischen graphischen Ausdrucksform führen. Dieser Anteil ist bei den eben erwähnten Beispielen unterschiedlich hoch. Er wird noch weit höher, wenn wir uns etwa hoch-spezialisierte Pläne von Leitungsnetzen vor Augen halten. Die systematische Analyse der Informationsinhalte, der Problemstellungen und der Lösungsmethoden stellt somit eine wesentliche Voraussetzung für den Aufbau eines raumbezogenen Informationssystems dar; wir müssen auch den Abstraktionsprozeß nachvollziehen, der zu den Formen der Karteninterpretation geführt hat, die uns heute geläufig sind.

Die Entwicklung des Menschen wurde und wird entscheidend von seiner Fähigkeit geprägt, seine Wahrnehmungen, Empfindungen und Schlußfolgerungen zu **abstrahieren** und **weiterzuvermitteln**. Er ist imstande, seine Erfahrungen zu formalisieren; er kann zwischen verschiedenen Bereichen Beziehungen herstellen und Querverbindungen bewußt ausnützen; das wesentliche daran ist, daß er dieses abstrakte Wissen auch weitergeben kann. Einzelerfahrungen können somit kollektiv nutzbar werden. Dieser Abstraktionsprozeß weist Sprungstellen auf. An diesen Sprungstellen hat sich das Selbstverständnis der Menschheit entscheidend verändert (Sinding-Larsen 1987). Die Entwicklung der *Sprache* war die erste größere Stufe: der Mensch konnte ein Objekt benennen, und der Name dieses Objektes wurde auch von den anderen Angehörigen seiner Gruppe verstanden. Später dehnte sich der Wortschatz auch auf abstrakte Begriffe aus. In einer nächsten Stufe lernte der Mensch, seine Sicht der ihn umgebenden Umwelt **graphisch** wiederzugeben; eine weitere Formalisierung der zeichnerischen Darstellung mündete in das Entstehen der *Schrift*, die sich weit von ihrer ursprünglichen bildhaften Bedeutung entfernt hat.

Auf einer anderen Abstraktionsebene entstanden Karten und Pläne, die zunächst eine systematische Abbildung der Umgebung zum Ziel hatten und natürliche Phänomene (etwa Flüsse) ebenso beschrieben wie von Menschenhand geschaffene Objekte (Bauwerke und Straßen). Das älteste uns erhaltene Beispiel für eine solche Karte ist ein 3500 Jahre alter Plan der mesopotamischen Stadt Nippur, der auf eine Tontafel eingeritzt wurde. Ebenso wie der Mensch Tausende von Jahren zuvor abstrakte Bezeichnungen in seinen Sprachschatz eingebracht hatte, so ließ er nun abstrakte Themen in die Karte einfließen: es wurden Hoheitsbereiche eingetragen, wichtige Gebäude markiert und oft in quasi-räumlichen Ansichten gezeichnet; geschichtliche Ereignisse und die Orte ihres Geschehens wurden bildlich festgehalten; Darstellungen von Schiffahrtswegen und Häfen wurden durch Schiffe und allerlei Meerestiere belebt. (Istituto Geografico Militare 1986.)

Ein Thema wurde somit bei Bedarf der rein geometrischen Darstellung überlagert, um bestimmte Sachverhalte besonders hervorheben zu können. Erst die Thematik (bzw. ihre Umsetzung in eine graphische Ausdrucksform) erlaubt es uns, die Karteninformation zu interpretieren, einzelne Karteninhalte miteinander zu vergleichen, und die daraus gezogenen Schlußfolgerungen als Entscheidungshilfen zu benutzen.

Allmählich wurden die darzustellenden Sachverhalte immer komplizierter und umfangreicher; die Thematik mußte daher in immer größerem Maße abstrahiert und durch Farben und Signaturen umgesetzt werden, ähnlich wie dies Jahrtausende zuvor bei der Abstraktion der Schriftzeichen geschah. Dieselbe Thematik wurde - je nach den Anforderungen - durch verschiedene Signaturen dargestellt. Aber auch die Geometrie wurde zum Träger mehrerer thematischer Bedeutungen.

Der Abstraktionsprozeß ist heute sehr weit fortgeschritten. Nach wie vor ist die Thematik so dominant, daß wir ohne genügende Hintergrundinformation über das jeweilige Thema nicht imstande sind, die Karte zu interpretieren; dies äußert sich besonders kraß bei der Darstellung innerstädtischer Versorgungsleitungen oder beim Plan der Einflugschneisen eines Flughafens. Trotzdem ist dieser Aspekt nicht auf unsere heutige hochtechnisierte Umwelt (und auch nicht auf graphische Darstellungen) beschränkt, wie ein interessanter Artikel von P. Damerow, R. Englund und H. Nissen (1988) beweist: auf archaischen Tontafeltexten, die zwischen 3200 und 3000 v.Chr. in der - ebenfalls mesopotamischen - Stadt Uruk entstanden, wurden zur Angabe der Größe von Feldern und zu den darauf geernteten Getreidemengen Symbole verwendet, die keinesfalls einheitlich, sondern - je nach der zugrundeliegenden Thematik - unterschiedlich zu interpretieren waren.

1.2. Raumbezogene Informationssysteme:

Begriffe und Gliederungen

Ein *raumbezogenes Informationssystem (RIS)* dient der **Erfassung, Speicherung, Verarbeitung und Darstellung** aller Daten, die einen Teil der Erdoberfläche und die darauf befindlichen technischen und administrativen Einrichtungen sowie ökonomische und ökologische Gegebenheiten beschreiben. Es kann als Überbegriff über all die im vorangegangenen Abschnitt genannten Wortschöpfungen verwendet werden. Es ist üblich, den Begriff **Geo-Informationssystem (GIS)** zu verwenden, wenn man eher die Funktion einer Karte vor Augen hat, und **Landinformationssystem (LIS)**, wenn Pläne verarbeitet werden. Dementsprechend werden auch die Begriffe **digitale Karte** und **digitaler Plan** verwendet. So wie wir im einleitenden Abschnitt starke Überlappungen zwischen den Funktionen von Karten und Plänen festgestellt haben, stellt sich auch hier die Frage, ob eine starre Trennung sinnvoll ist; im englischen Sprachgebrauch setzt sich für beide Bereiche die Bezeichnung GIS durch.

Schließlich wollen wir noch auf Begriffe wie **Grundstücksdatenbank (GDB)**, **Automatisierte Liegenschaftskarte (ALK)**, **Amtliches Topographisch-Kartographisches Informationssystem (ATKIS)**, **Mehrzweckkataster (MZK)** und **Leitungskataster** eingehen. Es sind dies Untermengen eines GIS bzw. LIS, die auf spezielle Anwendererfordernisse ausgerichtet sind und eine in sich abgeschlossene Einheit bilden.

Nach dieser Gliederung gemäß Anwenderkriterien wollen wir nun näher auf die Funktionen eines **Informationssystems** eingehen. Ein solches Informationssystem besteht aus Daten, die zu einer **Datenbank** zusammengefaßt sind, und aus einer Reihe von **Werkzeugen** zur Verarbeitung dieser Daten (Conzett 1983, Pflieger 1982). Es besteht somit aus mehreren Schichten. Zunächst liegen die Daten. Diese Sammlung von Daten wird erst durch ein **Datenbankverwaltungssystem (data base management system - DBMS)** zu einer Datenbank; das DBMS gruppiert den auf Dauer angelegten Datenbestand, schützt ihn und macht ihn verschiedenen Benützern zugänglich. Um diese Datenbank gruppieren sich eine Reihe von **Software-Werkzeugen** wie Transformationen, Algorithmen zur Verschneidung, zur Interpolation, zur graphischen Aufbereitung usw. Der Anwender hat die Möglichkeit, sowohl einzelne Daten wie auch einzelne Werkzeuge, die ihm das Informationssystem bietet, in seine Anwendung einzubauen (siehe Abb. 1.1).

Wir gehen noch einen Schritt weiter, wenn wir die Werkzeuge und Methoden zu einer **Methodenbank** zusammenfassen, ähnlich wie wir bereits die Daten zu einer Datenbank zusammengefaßt haben. Eine solche Formalisierung des Wissens führt uns in

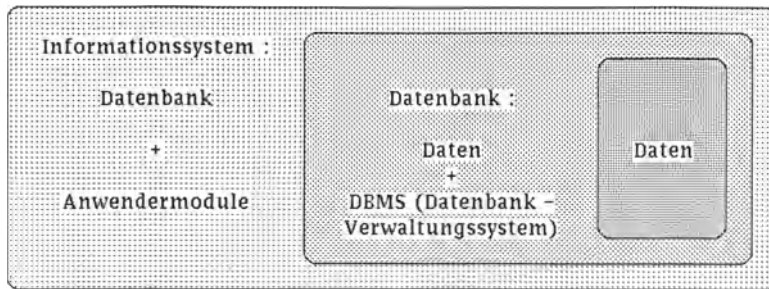


Abb. 1.1. Komponenten eines Informationssystems

den Bereich der *künstlichen Intelligenz* und der *Expertensysteme*, auf die wir in Kapitel 8 näher eingehen werden.

Der Begriff "Informationssystem" könnte den Eindruck erwecken, daß es sich um ein in sich geschlossenes, zentrales System von Hardware- und Softwarekomponenten handelt. Dies ließe sich angesichts des doch sehr allgemeinen Anspruches, der in der obigen Definition zum Ausdruck kommt, kaum in befriedigender Weise realisieren (siehe dazu auch Fischer 1986). Vielmehr wollen wir darunter ein **logisches Konzept** verstehen, das sich in einer Reihe von unterschiedlichen **Realisierungen** äußert, etwa in einem Netzwerk von Teilsystemen; jedes Teilsystem geht auf spezifische Belange ein, kann aber über **kompatible Schnittstellen** Daten und Werkzeuge mit anderen Netzteilnehmern austauschen. Ein solches **verteiltes System** kann optimal auf lokale Verhältnisse und auf geeignete Hardwarekonfigurationen eingehen (Abb. 1.2). Ein Konzept für ein solches System stellt beispielsweise der Entwurf zur Reform der Amtlichen Vermessung in der Schweiz (Eidg. Justiz- und Polizeidepartement 1987) dar.

Der Unterschied zwischen dem logischen Gesamtkonzept und der Palette von tatsächlichen oder möglichen Realisierungen kann anhand eines Beispiels aus dem EDV-Alltag deutlich gemacht werden: der Graphik-Standard GKS ("graphisches Kernsystem") sieht ein **logisches Arbeitsplatzkonzept** vor: verschiedene Arbeitsplätze können vom Programm angesprochen werden, ohne daß zunächst Details der tatsächlichen Realisierung angegeben werden müssen; erst bei der Durchführung des Programmes ist es notwendig, diese Details festzulegen. In diesem Sinn kann ein raumbezogenes Informationssystem durchaus aus verschiedenen Komponenten bestehen, die - jede für sich - mehr dem Typus eines LIS, eines GIS oder einer Untermenge davon entsprechen. Wichtig ist jedoch,

- daß die Komponenten in ein logisches Gesamtkonzept passen,

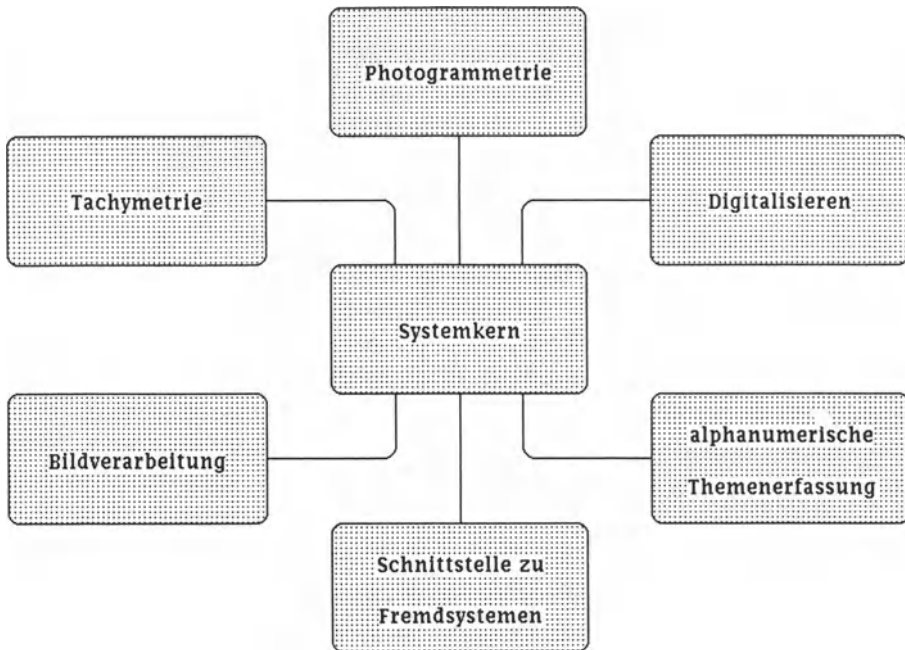


Abb. 1.2. Informationssystem als Netz von spezialisierten Teilsystemen

- daß das Gesamtsystem offen gegenüber künftigen Entwicklungen und Veränderungen ist,
- daß es möglichst allgemeingültig ist und nicht zu sehr auf ein bestimmtes Problem oder ein bestimmtes Anwenderprogramm ausgerichtet ist,
- daß es einen kontinuierlichen Datenfluß von der Datenerfassung bis zur Auswertung ermöglicht (*vertikale Integration*),
- und daß es Querverbindungen zwischen einzelnen Themen und zu anderen Informationssystemen erlaubt (*horizontale Integration*).

Einzelne Teile eines solchen logischen Gesamtkonzeptes können zu einer realen Arbeitsplatzkonfiguration zusammengefaßt werden. Die zentrale Rolle in einem raumbezogenen Informationssystem wird sicher der **graphisch-interaktive Arbeitsplatz (GIAP)** spielen, der die Sichtung und Korrektur der Daten auf einem graphischen Bildschirm erlaubt. Daneben gibt es - je nach den Ein- und

Ausgabesituationen, die von einem RIS unterstützt werden - Arbeitsplätze für die Bereiche

- *Digitalisierung,*
- *Aufbereitung photogrammetrischer Daten,*
- *Aufbereitung tachymetrischer Daten,*
- *Aufbereitung von Scannerdaten (Bildverarbeitung) und*
- *Ausgabe auf Plottern und Druckern.*

Schließlich wird auch oft die Ein- und Ausgabe archivierter Daten über Magnetbänder zu einer organisatorischen Einheit, einem "Metafile-Arbeitsplatz", zusammengefaßt. In Abb. 1.3 ist eine typische Arbeitsplatzkonfiguration für eine Anwendung aus dem Bereich des Vermessungswesens dargestellt.

1.3. Die Vorteile der digitalen Speicherung

Die Vorteile einer digitalen Speicherung und Verarbeitung raumbezogener Informationen sind vielfältig:

- Der **Variante**nreichtum der Darstellungsmöglichkeiten ist praktisch unbegrenzt. Farben, Strichstärken, Zeichensätze und Maßstäbe können beliebig variiert und dem Zweck der Darstellung untergeordnet werden.
- Graphische Informationen können mit alphanumerischen Informationen verbunden werden; so können etwa alle Daten, die für ein bestimmtes Gebäude gespeichert sind (Name des Eigentümers, Baujahr, Zustand, Versicherungsnummer, Anzahl der Mieter), bei Bedarf eingeblendet werden; es ist somit eine **integrierte Informationsvermittlung** möglich.
- Die **Geschwindigkeit** in der **Fortführung** ist ein wesentlicher Aspekt. Im Bereich eines RIS sind Datenbestände eher statisch: Eigentümerverhältnisse ändern sich nur selten, die Topographie ändert sich praktisch nie. Eine Straßenbegradigung etwa beeinträchtigt nur einen geringen Prozentsatz von Grundstücken. Wir können daher sehr schnell verschiedene Varianten einer neuen Trassenführung ausarbeiten und damit unsere Entscheidungen besser untermauern.
- Dieselben Daten können einer **Vielzahl von Auswertungen** zugeführt werden. So kann etwa die Bodenbeschaffenheit in eine Statistik von Grundstückspreisen einfließen. Daneben kann sie als Kriterium für die Wahl von Anbaugebieten dienen, die eine bestimmte Weinsorte begünstigen. Sie spielt auch für das Ausmaß einer Gefährdung durch Erosion eine Rolle. Langfristige Trends, wie etwa eine fortschreitende Versandung, können erkannt werden.

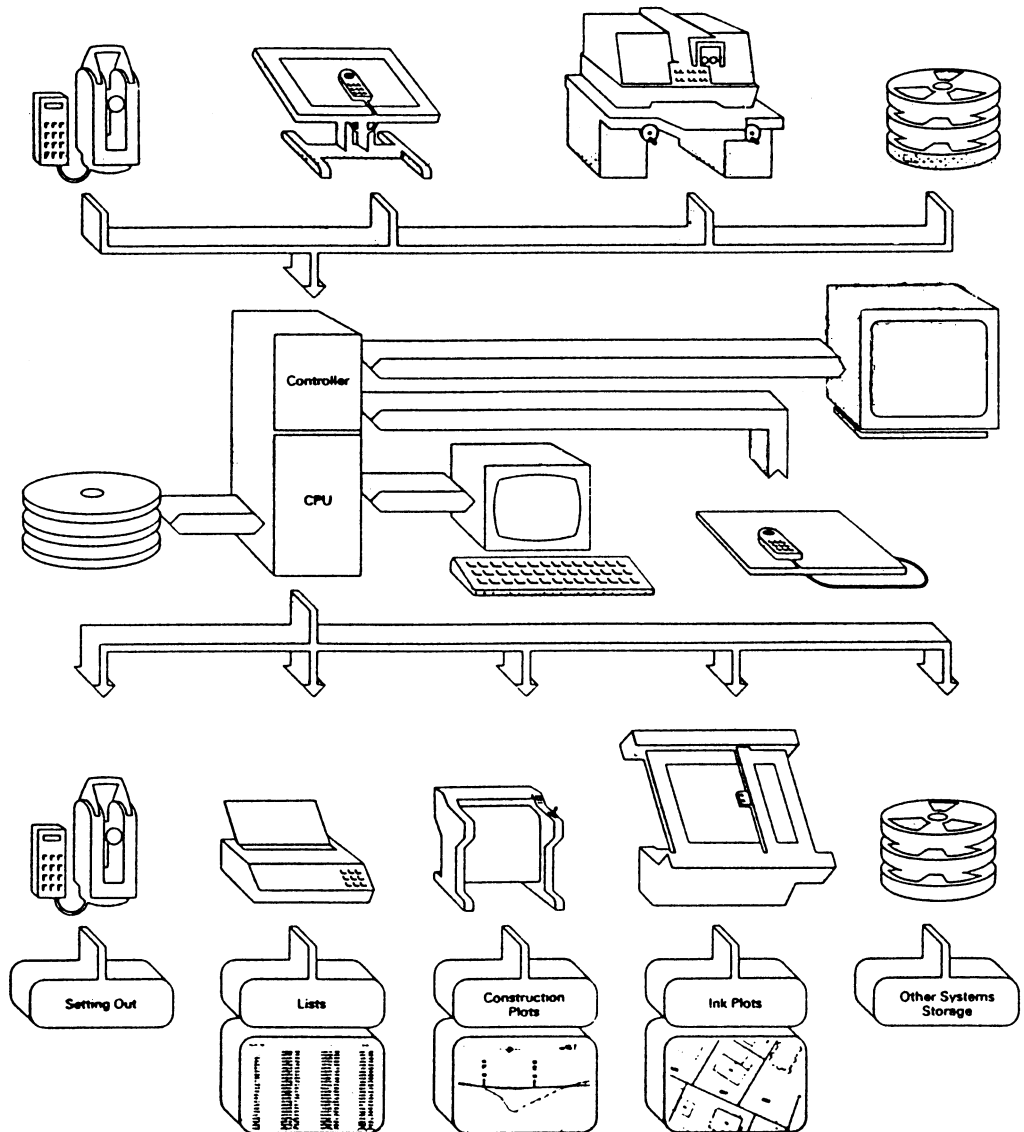


Abb. 1.3. Beispiel einer Arbeitsplatzkonfiguration im Vermessungswesen

- Themen können beliebig miteinander **kombiniert** werden. Dabei ergeben sich Verschneidungen, die als Grundlage für Entscheidungen dienen. So haben beispielsweise alle Stadtverwaltungen das Problem, daß Aufgrabungen und Arbeiten an unterirdischen Leitungssträngen besser koordiniert werden

sollen. Umweltschutzmaßnahmen bieten ein weiteres Beispiel: die Verschneidung von Lärmpufferzonen mit Wohnsiedlungen oder von Emissionszonen mit Waldgebieten ergibt wertvolle Hinweise über Prioritäten von Maßnahmepaketen. Ein wichtiger Aspekt ist in diesem Zusammenhang die Forderung nach einem **offenen System**: es ist zum Zeitpunkt der Konzipierung noch nicht abzusehen, welche Kombinationen von Themen notwendig sein werden; trotzdem soll das System flexibel genug sein, auf solche zukünftigen Anforderungen zu reagieren. Wenn etwa der Baumbestand in verschiedenen Gebirgstälern erhoben wird, und parallel dazu Nächtigungszahlen und lawinengefährdete Zonen registriert werden, so kann ein Auswerteprogramm ergründen, ob es einen statistisch gesicherten Zusammenhang zwischen dem überhandnehmenden Fremdenverkehr, der damit verbundenen Abholzung und der steigenden Lawinengefährdung gibt.

Abschließend muß noch gesagt werden, daß der rationelle Einsatz eines Informationssystems nur bei einer möglichst weitgehenden Ausnützung der eben angeführten Vielfalt in den Darstellungen, Auswertungen und Interpretationen gegeben ist. Erst wenn dieselben Daten und Methoden sehr oft und von verschiedenen Benutzern verwendet werden können, sind die Hardware- und Softwarekosten eines solchen Systems gerechtfertigt. Herkömmliche Methoden sind hingegen vorzuziehen, wenn es sich nur um einige wenige Pläne und Karten handelt, wenn diese nur selten gebraucht werden, oder wenn zu ihrer Erstellung Methoden benötigt werden, die kaum verallgemeinerungsfähig sind. Wir wollen auch noch erwähnen, daß wir in diesem Buch natürlich nur die technische Seite eines solchen Informationssystems beschreiben können. Die hier nicht behandelten *rechtlichen* und *administrativen* Aspekte sind freilich von mindestens ebenso großer Bedeutung.

1.4. Die Daten eines raumbezogenen Informationssystems

Die Daten eines raumbezogenen Informationssystems beschreiben **reale** oder **abstrakte Objekte** unserer Umwelt. Diese Beschreibung (das **Modell**) kann mehr oder minder genau sein. Wollen wir den Verlauf einer Straße beschreiben, so kann das entsprechende geometrische Modell ein Polygonzug sein, dessen Zwischenpunkte durch Geraden verbunden werden; wir können aber auch eine glatte Kurve durch diese Punkte legen; die Straße kann andererseits - in größeren Maßstäben - als Fläche auftreten. Für den Rand dieser Fläche sind selbst wieder verschiedene Modelle denkbar. Andere Daten liegen gar nicht in Vektorform, sondern in Rasterform vor, und das Modell muß dies berücksichtigen.

Neben den Verfeinerungsstufen des *geometrischen Modells* sind auch verschiedene *thematische Verfeinerungsstufen* denkbar: So können wir die Straßen eines Gebietes in Straßen mit getrennten Richtungsfahrbahnen (Autobahnen) und in sonstige Straßen einteilen. Die sonstigen Straßen könnten wieder in Fernverkehrsstraßen und Straßen mit regionaler und lokaler Bedeutung eingeteilt werden, usw.

Wir sehen also, daß die Daten eines raumbezogenen Informationssystems unterschiedliche Eigenschaften der zu beschreibenden Objekte wiedergeben. Es sind dies

- a) **geometrische** Eigenschaften,
- b) **topologische** Eigenschaften und
- c) **thematische** Eigenschaften.

Zu a): Die Geometrie kommt im **Raumbezug** zum Ausdruck, den alle Objekte in einem mehr oder minder starken Ausmaß aufweisen; sie erfüllen also Voraussetzungen bezüglich der **Lage** und der **Ausdehnung**: für Gebäude, Grundstücke, Flüsse oder Wasserleitungen ist dies offensichtlich; aber auch Nutzungen, Eigentümerverhältnisse, Netzkapazitäten beziehen sich auf bestimmte geometrisch abgegrenzte Bereiche, so daß auch ihnen ein - wenn auch etwas schwächerer - Raumbezug zukommt.

Zu b): Topologische Eigenschaften äußern sich in Beziehungen der **Nachbarschaft**, des **Enthaltenseins**, der **Überschneidung** und ähnlichem. Welche Grundstücke grenzen an eine Straße, welche Leitungen verlaufen in der Nähe von Gebäuden, welche Waldgebiete sind aufgrund ihrer Nähe zu kalorischen Kraftwerken oder stark befahrenen Straßen besonders gefährdet? All dies sind topologische Fragestellungen; sie sind nicht (ausschließlich) durch exakte Koordinatenangaben zu interpretieren, denn der Begriff der "Nähe" ist unterschiedlich zu deuten, wenn es um eine möglichst kurze Anschlußleitung zum Fernwärmenetz, um die Nahversorgung von Konsumenten oder um Autobahnanschlüsse geht.

Zu c): Schließlich gehört jedes Objekt einem bestimmten **Thema** an. Wir können nur eine Auswahl denkbarer Themen angeben:

- **Topographie**: Gebirge, Talsenken, Gewässer;
- **Grundstückseigentum**: Grundbuch, Kataster;
- **Ökologische Voraussetzungen**: Klima (Temperatur, Niederschlag, Besonnung), Geologie (Lagerstätten), Bodengüte, Gewässergüte und Luftgüte;
- **Technische Einrichtungen**: Verkehrsanlagen, Projekte, Ver- und Entsorgungsnetze;

- **Nutzung:** Flächennutzung, Siedlungsdichte, Infrastruktur;
- **Bevölkerungsdaten:** Bevölkerungsbewegung (Abwanderung, Zuzug), Statistiken über den Anteil an Pendlern, Arbeitslosen, Jungfamilien;
- **Daten der Wirtschaft:** Bodenertrag, Waldflächen, Viehbestand, Betriebsstruktur, Fremdenverkehr.

Geometrische, topologische und thematische Aspekte von Daten beeinflussen einander. Bei der manuellen Erstellung wie auch bei der Interpretation von Karten und Plänen beachten wir diese Verflechtungen stillschweigend: Straßen haben (im großmaßstäblichen Bereich) immer einen linken und einen rechten Straßenrand; diese beiden Ränder dürfen einander nicht schneiden. Häuser stehen immer auf Grundstücken; sie sind von geschlossenen Linienzügen umrahmt und dürfen einander nicht überlappen. Grundstücke haben immer mindestens einen Eigentümer. Flüsse dürfen keine geschlossenen Schleifen aufweisen; Leitungsnetze müssen zusammenhängend sein.

Die Liste dieser **Konsistenzbedingungen** ließe sich beliebig fortsetzen. Ihrer Beachtung kommt in einem raumbezogenen Informationssystem eine besondere Bedeutung zu; dessen Wirkungsgrad wird entscheidend davon abhängen, wie gut wir die Erfahrung eines geübten Kartographen nachvollziehen können, und wie weit sein Wissen um Daten und deren Zusammenhänge formalisiert werden kann. Das Verhindern von Inkonsistenzen ist in einem System, das der Langzeitspeicherung dient, von großer Bedeutung. Natürlich ist es in einer EDV-Umgebung notwendig, die Konsistenzbedingungen *explizit* zu definieren; dies hat aber auch den Vorteil, daß sämtliche Anwenderprogramme auf diesen Voraussetzungen aufbauen können.

Daten bilden den Kern eines raumbezogenen Informationssystems. Ihnen wird daher in diesem Buch ein breiter Raum gewidmet. Die geometrisch-topologischen Aspekte werden in Kapitel 2 behandelt, die thematischen Aspekte in Kapitel 3. In diesen Kapiteln geht es hauptsächlich um die Modellbildung. Die tatsächliche Speicherung und die dabei verwendeten Ansätze werden in den Kapiteln 4 und 5 beschrieben. Der Unterschied zwischen der **konzeptionellen** Betrachtungsweise von Daten und der **externen** Sicht wird in Kapitel 7 erläutert: während die erstere eine interne Einheitlichkeit aller Daten eines Informationssystems garantiert und neutral gegenüber verschiedenen Anwendungen ist, beschreibt die letztere eben die persönlich gefärbte Sicht, die ein bestimmter Anwender bezüglich der Daten entwickelt.

1.5. Das Instrumentarium eines raumbezogenen

Informationssystems

Wie wir bereits in Abschnitt 1.2 festgestellt haben, beschränkt sich ein Informationssystem nicht nur auf eine Datenbank. Erst wenn ein Instrumentarium an Anwendermethoden, Algorithmen und Funktionen zur Verfügung gestellt wird, können wir von einem Informationssystem sprechen. Die Daten sind zwar der Treibstoff, den wir zum Betrieb eines solchen Systems brauchen. Mindestens ebenso wichtig aber sind die Kanäle, durch die dieser Treibstoff fließt, der Vergaser, der Motor und die Kraftübertragung. Daten allein sind wenig aussagekräftig, wenn es keine effizienten Werkzeuge zur Bearbeitung derselben gibt.

Es lohnt sich immer, wenn wir uns vor Augen halten, welche gedanklichen Prozesse beim Betrachten einer Karte angestoßen werden. Nehmen wir zum Beispiel eine Wanderkarte zur Hand. Wenn uns das Gelände vertraut ist, so holen wir aus dem "Langzeitspeicher", also aus unserem Gedächtnis, ein Modell dieses Geländes vor unser inneres Auge. Dabei generalisieren wir, das heißt, wir lassen alle derzeit unwichtigen Details außer acht. Anhand von "Paßpunkten", also von Merkmalen, die wir aus eigener Anschauung kennen, orientieren wir uns in der Karte; dies bedeutet, daß wir eine Paßpunkttransformation zwischen der Karte und unserem gedanklichen Modell vornehmen. Wenn wir den Verlauf eines bestimmten Wanderweges verfolgen, so schneiden wir ihn mit den eingezeichneten Schichtenlinien; wir interpolieren zwischen den Schnittpunkten und können dadurch die Steigung feststellen.

Dies sind alles Vorgänge, die größtenteils unbewußt ablaufen. In einem digitalen Umfeld müssen wir diese Vorgänge nachvollziehen, und dazu ist es notwendig, explizit Werkzeuge zu definieren, die dies unterstützen. Deshalb haben wir das Kapitel 6 der Beschreibung einzelner Werkzeuge gewidmet; so werden etwa folgende Bereiche näher beschrieben:

- Transformationen und Projektionen,
- Schnittberechnungen,
- Rasteralgorithmen,
- Interpolation und Approximation von Kurven,
- digitale Höhenmodelle, und
- Ansätze zur Generalisierung.

Natürlich kann dies nur ein Ausschnitt aus dem Spektrum von Anwendermodulen sein, die wir für ein raumbezogenes Informationssystem ins Auge fassen; so wären etwa die folgenden Teilbereiche ebenso abzudecken:

- *Bearbeitung tachymetrischer Daten,*
- *Bearbeitung photogrammetrischer Daten,*
- *Bearbeitung digitalisierter Daten,*
- *Bearbeitung von Scannerdaten,*
- *Schnittstellen zu Fremdsystemen,*
- *Bildverarbeitungsalgorithmen,*
- *Konstruktionen,*
- *graphische Aufbereitung,*
- *numerische Aufbereitung (in Listen usw.).*

Abbildung 1.4 zeigt ein Diagramm möglicher Anwendermodule, die in Verbindung mit einer Datenbank ein raumbezogenes Informationssystem bilden.

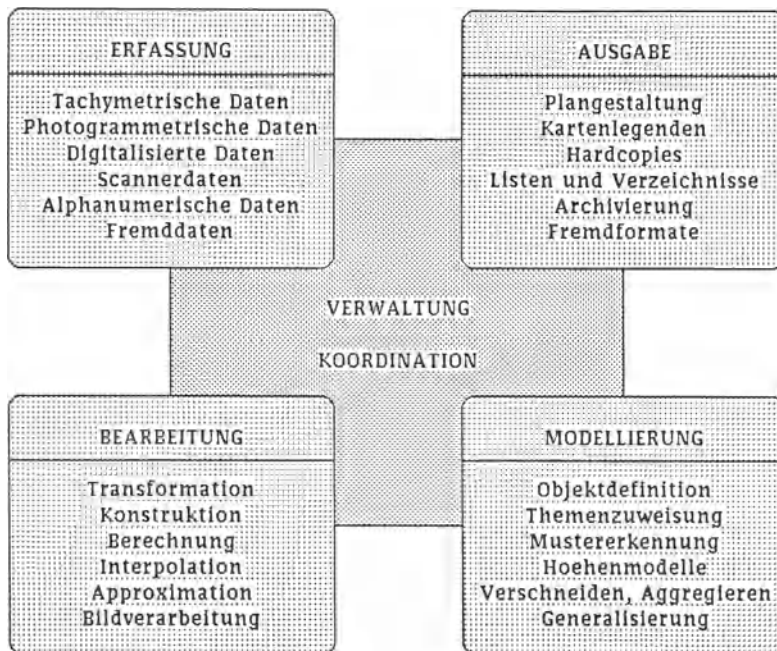


Abb. 1.4. Ein RIS (raumbezogenes Informationssystem) und seine Anwendermodule

Neben Kapitel 6 sind auch Teile der Kapitel 7 und 8 dem Instrumentarium an Anwenderfunktionen gewidmet, die von einem raumbezogenen Informationssystem zur Verfügung gestellt werden. In Kapitel 7 befassen wir uns näher mit der Frage, wie ein Anwender diese Methoden in seine Problemlösung einbauen kann, und Kapitel 8 ist den zukünftig zu erwartenden Entwicklungen auf diesem Gebiet gewidmet.

1.6. Beispiele für raumbezogene Informationssysteme

Wir wollen nun einige Anwendungsgebiete aufzählen, die derzeit bereits in ein raumbezogenes Informationssystem eingebunden werden. Dies kann natürlich keine vollständige Aufzählung sein. Vielmehr wollen wir jeweils einen typischen Repräsentanten aus jeder Gruppe von möglichen Anwendungen herausgreifen.

- **A) Anwendungen im Katasterbereich:** Als Beispiel möge die *ALK (Automatisierte Liegenschaftskarte)* in der BRD dienen. Zwischen den Ländern Hessen, Nordrhein-Westfalen und Niedersachsen wurde eine Verwaltungsvereinbarung getroffen, die den Aufbau und die Weiterführung eines Informationssystems zur Verwaltung von Flurstücken und Gebäuden vorsieht. Das System ist als logisches Gesamtkonzept zu betrachten, das verschiedenartige Erfassungssysteme und -methoden unterstützt. Mit der Erfassung der Daten wurde bereits begonnen. Neben den technischen Aspekten tritt hier besonders das Problem der Erarbeitung von allgemeingültigen Vorschriften für die Vorgangsweise in den Vordergrund, müssen doch verschiedene Verwaltungsstrukturen und Methoden berücksichtigt werden (Mittelstraß 1987).

- **B) Mehrzweckkataster:** Im Kanton Basel-Stadt wird stufenweise ein Informationssystem aufgebaut, das ein leistungsfähiges Instrumentarium für Planungs-, Entscheidungs- und Kontrollaufgaben darstellt (Messmer 1983). Eine Vielzahl von Themen wird dabei berücksichtigt, angefangen von Grundstücks- und Gebäudeverzeichnissen über Leitungskataster, Einwohner- und Fahrzeugstatistiken, Steuerverwaltung, Handelsregister usw. Ähnliche Informationssysteme entstehen auch in vielen anderen Städten. Sie enthalten beispielsweise auch Angaben über die Infrastruktur, die Zusammensetzung der Einwohner nach Altersgruppen, Einsatzpläne für die Feuerwehr, Netzanalyse- und Netzoptimierungsmodule und vieles mehr. (Willmersdorf 1986). Oft werden solche Mehrzweckkataster auch von kommunalen und regionalen Versorgungsunternehmen angelegt (Ponschab 1985).

- **C) Umweltkataster:** Dem Schutz unseres bedrohten Lebensraumes ist eine Fülle von Projekten gewidmet. Das komplexe Zusammenspiel verschiedenster Einflußfaktoren kann durch ein Informationssystem erkannt und ausgewertet werden. Der Bogen der Einsatzmöglichkeiten spannt sich vom lokalen und regionalen Bereich (Hönig 1984, Baumann et al. 1985) bis zu einem gesamteuropäischen Umweltinformationssystem (Wiggins et al. 1987).

- **D) Topographie:** Das *Amtliche Topographisch-Kartographische Informationssystem ATKIS* der BRD hat die Aufgabe, geometrische und thematische Informationen der topographischen Landeskartenwerke (Gelände, Gewässer, Verkehr, Besiedlung, Boden, Grenzen, geographisches Namensgut) zur Verfügung zu stellen (Appelt 1987, Harbeck 1987). Diese interessensneutral angebotenen Informationen können dann in verschiedene Auswertungen eingebunden werden.

- **E) Verkehrsleitsysteme:** Es gibt bereits Pilotprojekte, welche die Möglichkeiten für eine elektronische Steuerung des Straßenverkehrs untersuchen. Die Grundlage eines solchen Systems muß die digitale Abbildung eines Straßennetzes sein, wobei jeder Straße thematische Zusatzinformationen (Verkehrsaufkommen, Belastbarkeit, Zustand usw.) zugeordnet werden (Hentschel 1987). Ähnliche Bemühungen gibt es etwa auch für die Koordinierung des Schiffsverkehrs (Berger 1986).

Diese Aufzählung der Einsatzbereiche von raumbezogenen Informationssystemen erhebt keinen Anspruch auf Vollständigkeit. Andere Anwendungsbereiche betreffen etwa

- *Digitale Geländemodelle (DGM), Dichte- und Geoidmodelle,*
- *Strukturplanung (Pfleger 1982),*
- *Vegetationsanalysen (Reiner 1988),*
- *geologische Gegebenheiten,*
- *Untersuchungen des Wasserhaushaltes und*
- *bodenkundliche Bestandsaufnahmen.*

Für Berichte über weitere Anwendungen möge der interessierte Leser auf die Reihe "Nachrichten aus dem Karten- und Vermessungswesen" (Verlag des Institutes für Angewandte Geodäsie, Frankfurt a.M.) zurückgreifen, die sehr viele Beiträge enthält, welche die Problematik aus der Anwendersicht beleuchten. Die Reihe "International Journal of Geographical Information Systems" (Verlag Taylor & Francis, London) legt das Schwergewicht eher auf die Konzeption solcher Systeme. Eine ausführliche Darstellung der Problematik aus verschiedenen Blickwinkeln bieten auch die "Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984" (Geographisches Institut, Abteilung Kartographie/EDV, Universität Zürich-Irchel). Interessante Möglichkeiten für den Einsatz raumbezogener Informationssysteme im Bereich der Geologie, Hydrologie, der Lagerstättenforschung und verwandter Wissensgebiete werden im Protokoll der Interdisziplinären Fachtagung GeoLIS (TU Wien, April 1986) aufgezeigt.



KAPITEL 2
GEOMETRIE UND TOPOLOGIE

2.1. Die Modellbildung

Raumbezogene Informationssysteme speichern eine ungeheuer große Zahl unterschiedlicher Informationen. Um dieser Fülle von Informationen Herr zu werden, müssen wir sie *strukturieren*. Wir benützen dazu zwei Strategien, die einander entgegengesetzt sind: einerseits teilen wir die Informationen in ihre Bestandteile auf und klassifizieren diese entsprechend; diesen Vorgang bezeichnen wir als **Analyse**. Andererseits formen wir aber auch aus einzelnen Teilen komplexe Gebilde (**Synthese**).

In einem solchen Informationssystem wollen wir bestimmte Teile der Wirklichkeit in ein (mehr oder weniger genaues) Modell *abbilden*. Betrachten wir als Beispiel den Verlauf eines Flusses (Abb. 2.1). Ein einfaches Modell dieses Flusses ist ein Polygon, also eine Folge von Koordinatenpaaren, die durch gerade Linien verbunden werden. Dies kann natürlich nur ein kleiner Teil dessen sein, was die Information "Fluß" in Wirklichkeit ausmacht.

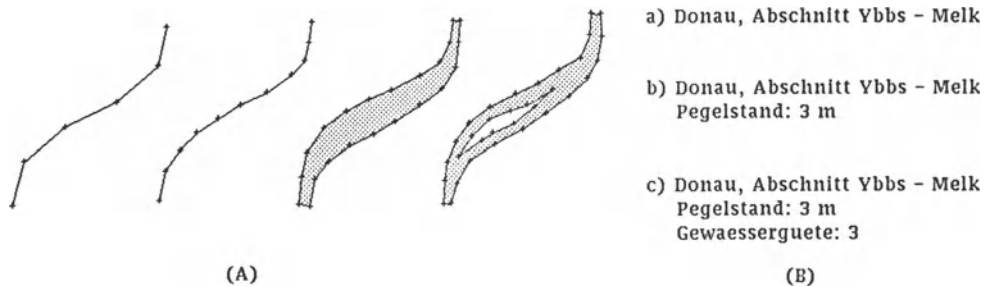


Abb. 2.1. Modell eines Flusses. Verfeinerung der Topologie (A) und der Thematik (B)

So beschränkt sich dieses Modell auf die **geometrisch-topologische Komponente** des Flusses. Wir können es nun verfeinern, indem wir aus dem Bereich der **Thematik** den Flußnamen hinzunehmen. Andere thematische Komponenten (etwa der Verschmutzungsgrad, die Schiffbarkeit, die durchschnittliche Wassertiefe usw.) werden nicht einbezogen. Trotzdem wird diese einfache Modellbildung in vielen Fällen ausreichen.

Natürlich können wir die Geometrie unseres Modells verfeinern, indem wir Höhen mit einbeziehen, indem wir zusätzliche Stützpunkte des Polygons einführen, und indem wir eine möglichst glatte Kurve durch diese Stützpunkte legen. Handelt es sich um eine Karte in großem Maßstab, so würden wir breite Flüsse nicht in ein linienförmiges Modell abbilden, sondern vielmehr Flächen verwenden.

Inseln im Bereich eines solchen Gewässers blieben bisher unberücksichtigt. Sie gehören zur **Topologie** des Flusses. (Auch die Reihenfolge der Punkte in unserem Polygon ist als topologische Eigenschaft anzusehen. Das gleiche gilt für den Umstand, daß das Polygon zusammenhängend ist, also nicht etwa einen Karstfluß widerspiegelt, dessen oberirdischer Verlauf von Zeit zu Zeit unterbrochen ist.)

Wir sehen also, daß wir dieses Modell immer mehr verfeinern können. Der Grad der Verfeinerung hängt von den Anwendungen ab, denen wir dieses Modell zugrunde legen wollen. Allerdings sind uns Grenzen gesetzt, und zwar aus verschiedenen Gründen:

- der Speicherplatz und die erreichbare Genauigkeit sind begrenzt; dies zwingt uns, die Verfeinerung nach endlich vielen Schritten abzubrechen.
- je feiner das Modell wird, desto unhandlicher wird seine Bearbeitung, und desto weniger flexibel ist es im Hinblick auf neue, unvorhergesehene Anwendungen.
- eine extreme Verfeinerung führt uns in den Bereich des Paradoxons. So hat etwa Mandelbrot (1983) nachgewiesen, daß die Küstenlinie Englands unendlich lang wird, wenn man diese Verfeinerung beliebig lange fortsetzt. Wir wollen uns in Abschnitt 2.6 näher mit den Auswirkungen dieser Überlegungen beschäftigen.

Ein Aspekt der Realität kann, aus verschiedenen Blickwinkeln betrachtet, zu verschiedenen Modellen Anlaß geben. Frank (1983a) erklärt dies anhand eines Beispiels aus der Alltagssprache: das gesprochene Wort löst in uns unterschiedliche Assoziationen aus; jeder, der das Wort "Haus" hört, bildet dazu in Gedanken ein anderes Modell dieses Teiles der ihn umgebenden Wirklichkeit; ein Kind wird ein viel einfacheres Modell eines Hauses gebrauchen als ein Bauingenieur; sogar ein- und dieselbe Person wird zu verschiedenen Zeiten und zu verschiedenen Anlässen unterschiedliche Modelle gebrauchen.

Die Modellbildung geschieht in einem Bereich, der zwischen der Wirklichkeit und der tatsächlichen Speicherung liegt. Wo etwa die Koordinaten der Stützpunkte für den oben erwähnten Flußlauf abgelegt werden, ob dies in Form einer verketteten Liste oder sequentiell geschieht, wie viele Stellen für jede Koordinate vorgesehen werden, dies sind Fragen, die dem *internen Schema* eines Datenverwaltungssystems zugeordnet werden müssen. Das Modell hingegen entspricht dem *konzeptionellen Schema*. Es gibt den Rahmen der künftigen Anwendungen an. So fehlen etwa im obigen Beispiel des einfachsten Modells für den Fluß die Höhenangaben. Es wird daher nie Anwenderprogramme geben können, welche auf die Höhe von Punkten Bezug nehmen.

Neben dem internen und dem konzeptionellen Schema gibt es dann - je nach Anwendung - mehrere externe Schemata. Für diese werden jeweils Teilmengen des konzeptionellen Schemas auf bestimmte Anwendergruppen zugeschnitten. Eine nähere Erläuterung dieser drei Schemata wird in Kapitel 4 gegeben.

Raumbezogene Informationen haben also immer einen **geometrischen Aspekt** (Koordinaten, aber auch die Form von Verbindungen, wie etwa Gerade, Kreisbogen u.a.), einen **topologischen Aspekt** (Zusammenhang, Inseln, Verschneidungen, Überlappung, Nachbarschaft) und daneben noch **thematische Aspekte**. Wir können ein RIS als *Puzzle* ansehen, das aus vielen kleinen Teilchen besteht. Bereits im Kindesalter haben wir gelernt, beim Zusammenfügen eines Puzzles sowohl auf geometrisch-topologische Aspekte (die Form der einzelnen Teile) wie auch auf thematische Aspekte (den Bildinhalt) zu achten. Beide Komponenten ergänzen einander.

In diesem Kapitel wollen wir die geometrischen und topologischen Aspekte raumbezogener Informationen näher durchleuchten. In Kapitel 3 befassen wir uns dann mit der Thematik.

2.2. Vektor- und Rastermodelle: eine Gegenüberstellung

Das geometrisch-topologische Modell als (mehr oder weniger genaues) Bild eines Teiles der Wirklichkeit kann zwei verschiedenen Kategorien angehören. Das **Vektormodell** beinhaltet Punkte, linienförmige Verbindungen dieser Punkte, sowie Flächen, die von solchen Linien eingerahmt werden. Ein Vektormodell eines Grundstückes etwa besteht aus den Koordinaten der Eckpunkte, ferner aus der Vorschrift, daß diese Eckpunkte in einer bestimmten Reihenfolge so zu verbinden sind, daß ein geschlossenes Polygon entsteht, welches sich nicht selbst schneidet und demnach eine Fläche einrahmt. In Abb. 2.2 ist das Vektormodell dieses Grundstückes dargestellt.

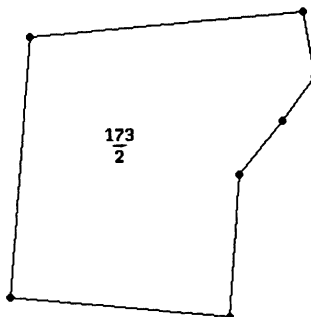


Abb. 2.2. Vektormodell der Topologie eines Grundstückes

Die Fläche selbst kann nur in Verbindung mit Linien bestehen, ebenso wie eine Linie nur auf bestehende Punkte aufbauen kann. Unter Umständen kann die Fläche eine oder mehrere **Inseln** haben. In diesem Fall gibt es neben den äußeren Grenzen auch *innere Grenzlinien*, welche die Fläche gegen diese Inseln abgrenzen. Hier sei als Beispiel das Bundesland Niederösterreich genannt, in dem das Bundesland Wien eine Insel bildet (Abb. 2.3).



Abb. 2.3. Inseln im Vektormodell. Die Grenzen von Niederösterreich und Wien

Bilden wir hingegen ein **Rastermodell**, so abstrahieren wir die Eigenschaften eines Objektes derart, daß wir sie innerhalb von rechteckigen und regelmäßig in Rasterform angeordneten Bereichen ("Zellen") als homogen auffassen. Rastermodelle sind somit für die Beschreibung flächiger Sachverhalte weit besser geeignet als Vektormodelle (die ihrerseits wieder eine Stärke für linienhafte Verbindungen aufweisen). Man bezeichnet Rastermodelle deshalb auch oft als **areale Modelle**, im Gegensatz zu den als **lineale Modelle** bezeichneten Vektormodellen.

An dieser Stelle sei auf die Funktion graphischer Ein- und Ausgabegeräte verwiesen: es gibt Geräte, die nach dem Vektorprinzip arbeiten (Digitalisieretisch, Stiftplotter), und solche, die nach dem Rasterprinzip arbeiten (Scanner, Matrixnadelldrucker, Laserdrucker, Tintenstrahldrucker, sowie die meisten neueren graphischen Bildschirme). Dort entspricht die kleinste homogene Rasterzelle einem *Pixel*. Dies ist auch der Fall, wenn es sich um das Modell eines Luft- oder Satellitenbildes handelt. Oft jedoch kann diese kleinste Rasterzelle in einem RIS auch eine allgemeinere Bedeutung haben, etwa wenn wir ein Gebiet in regelmäßig angeordnete Zonen gleicher Bedeutung aufteilen. Als Beispiel sei der *Schweizer Hektarraster* genannt, der das gesamte Schweizer Bundesgebiet in einem 100 m x 100 m -

Raster einteilt (Abb. 2.4). In jeder Rasterzelle werden die Nutzung, die Gemeindezugehörigkeit, die mittlere Neigung usw. erhoben (Hase et al. 1972).

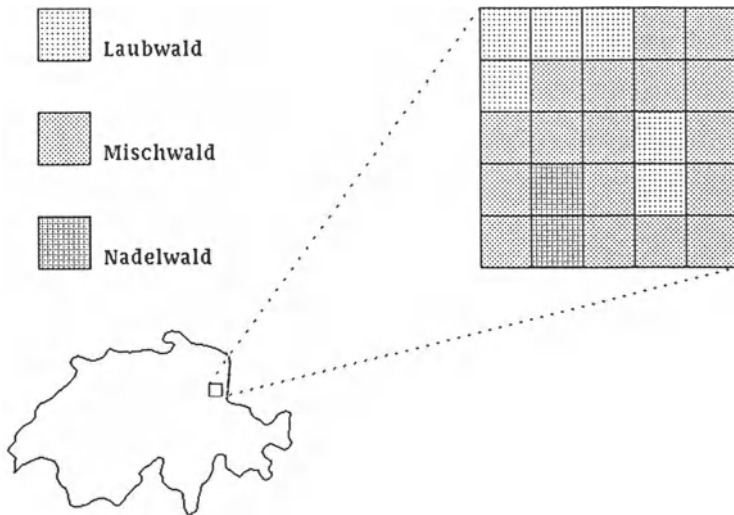


Abb. 2.4. Beispiel eines Rastermodells

Wir haben uns bisher auf zweidimensionale Modelle beschränkt, denn ein Hauptanliegen raumbezogener Informationssysteme ist eine Beschreibung eines Teiles der *Erdoberfläche*, und diese ist eben zweidimensional. Ein solches System wird zwar Höhenangaben verwalten, aber diese haben gegenüber den Lageangaben eine eher untergeordnete Bedeutung; ihr Charakter ist mehr der einer zusätzlichen Beschreibung (eines Attributes), während die Lageangabe oft als Zugriffskriterium (Schlüssel) verwendet wird. Die meisten Sachverhalte, die ein solches System beschreibt, lassen sich also ohne Informationsverlust auf eine Fläche abbilden; sie sind demnach zweidimensional.

Soll jedoch ein raumbezogenes Informationssystem im Bereich der Geologie, der Hydrologie, der Lagerstättenforschung usw. eingesetzt werden, so wird die dritte Dimension an Bedeutung gewinnen, und wir müssen Modelle in Betracht ziehen, die dieser dritten Dimension entgegenkommen.

Eine Verallgemeinerung des Vektormodells auf drei Dimensionen wird oft auch als **Drahtmodell** (wire frame model) bezeichnet. Dieser Ausdruck kommt aus dem CAD-Bereich, wo dieses Modell (neben anderen dreidimensionalen Modellen) Verwendung findet. Stellt man alle Verbindungslinien eines Drahtmodells dar, so erscheint es durchsichtig.

Das Rastermodell hingegen verallgemeinert sich auf das **Volumenmodell**, das regelmäßig angeordnete Quader homogener Bedeutung (ein sogenanntes *Voxel*, analog zum *Pixel*) beinhaltet. Der Vollständigkeit halber wollen wir noch erwähnen, daß es neben den angeführten Modellen noch weitere gibt, die aber hauptsächlich im CAD-Bereich Anwendung finden. So etwa könnte man zweidimensionale Objekte durch eine Kombination von einfachen Grundfiguren (Linien, Kreise, Quadrate) modellieren; dreidimensionale Modelle würden dementsprechend Quader, Kugeln, Zylinder, Kegel usw. enthalten.

Im dreidimensionalen Raum gibt es aber naturgemäß eine höhere Zahl von Varianten im Rahmen der Modellbildung. So würden wir die Darstellung eines Körpers über seine Oberflächen als eine Mischform zwischen Raster- und Vektorsichtweise ansiedeln. In diesen Bereich würde etwa auch der Globus als Darstellung der Erdoberfläche fallen. Wollten wir beispielsweise eine altägyptische Pyramide modellieren, so würde das Drahtmodell auf ihre Kanten aufbauen; das Flächenmodell würde die Grundfläche und die vier Seitenflächen beinhalten; ein Volumenmodell wäre vielleicht aus einzelnen Basisquadern zusammengesetzt, die den tatsächlich verwendeten Steinblöcken entsprechen (Abb. 2.5).

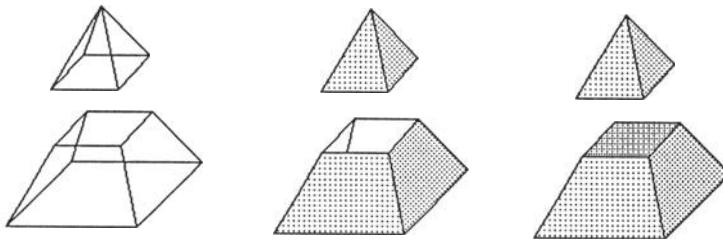


Abb. 2.5. Drahtmodell, Oberflächenmodell und Volumenmodell

Die Beantwortung der Frage, welches Modell besser für bestimmte Anwendungen geeignet ist, hängt von einer Reihe von Kriterien ab. Grundsätzlich sind uns beide Strategien geläufig. Wir zeichnen, lesen und schreiben nach dem Vektorprinzip, wir lassen Bilder nach dem Rasterprinzip auf uns einwirken. Stricken und Weben sind ebenfalls Rastertechnologien.

Wenn wir die Vor- und Nachteile von Vektormodellen und Rastermodellen einander gegenüberstellen, so kommt es natürlich in erster Linie auf das Objekt bzw. die Klasse von Objekten an, die wir modellieren wollen; wie wir bereits früher gesagt haben, lassen sich eindimensionale Verbindungen (etwa Leitungen) besser in Vektormodelle

abbilden, während zweidimensionale Zusammenhänge (zum Beispiel Waldgebiete) eher durch ein Rastermodell dargestellt werden.

Die Entscheidung ist nicht immer so einfach. Es kommt auch auf die Art der Anwendungen an, die wir im Auge haben, und ebenso auf die Speicherungsmethode und die interne Gruppierung der Daten. In diesem Zusammenhang müssen wir auf den Flaschenhals im jeweiligen Modell zu sprechen kommen. Der Engpaß im Rastermodell ist die Datenmenge. Denken wir an die (ca.) 250000 Bildpunkte eines Fernsehbildes.

Wollen wir den Speicherplatzbedarf in Grenzen halten, müssen wir Abstriche in punkto Genauigkeit machen. Auf einen einfachen Nenner gebracht bedeutet dies, daß Rastermodelle nicht so genau wie Vektormodelle sein können. In unserem Anwendungsspektrum eignen sie sich eher für großflächige Zusammenhänge, wie etwa im obigen Beispiel der Waldzonen. Grundstücke hingegen sind zwar auch Flächen, ihre Begrenzungspunkte weisen aber Genauigkeiten im cm-Bereich auf. Ein Rastermodell eines solchen Grundstückes müßte demnach ebenfalls auf Rasterzellen im cm-Bereich aufbauen.

Vektormodelle zeichnen sich durch Sparsamkeit im Umgang mit dem Speicherplatz aus. Ein Grundstück, bestehend aus vier Grenzpunkten, benötigt nicht viel mehr Speicherplatz als vier Paare von Koordinaten.

Wie bereits erwähnt, spielen auch die geplanten Anwendungen eine Rolle. Wenn wir geometrische Operationen betrachten, so ist es klar, daß Koordinatentransformationen für Vektormodelle höchst einfach, für Rastermodelle hingegen schwierig (zumindest zeitaufwendig) sind (siehe Kapitel 6). Andererseits ist die Ermittlung von Schnitten und Nachbarschaften in Rastermodellen trivial: jede (innere) Rasterzelle hat genau vier Nachbarn, die leicht ansprechbar sind.

Dieses Problem kann bei Vektormodellen beinahe unlösbar werden: denken wir an zwei aneinanderstoßende Kartenblätter, die sich ein Waldgebiet mit vielen Verästelungen, Einbuchtungen und Waldlichtungen (in unserer Terminologie sind dies Inseln) teilen; für ein Vektormodell müssen zusammenhängende (Teil-)Polygone gefunden werden, die dann zu geschlossenen Umrandungen von Flächen aufgefädelt werden müssen; Inseln müssen gefunden und entsprechend markiert werden; Inseln können selbst wieder Inseln beinhalten (eine Baumgruppe auf einer Waldlichtung) usw. (siehe Abb. 2.6).

Beziehungen zwischen Nachbarn können in Rastermodellen implizit über benachbarte Rasterzellen hergestellt werden; in Vektormodellen muß diese Beziehung explizit definiert werden. Wollen wir etwa das obige Waldgebiet mit der

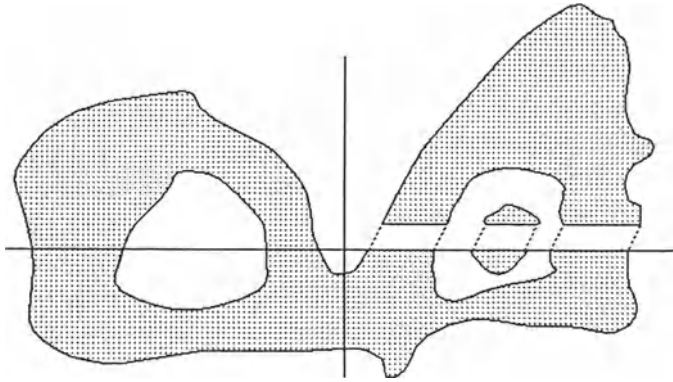


Abb. 2.6. Aufschneiden einer Fläche mit Inseln

Emissionszone eines Schadstofferregers schneiden, und wollen wir den Flächeninhalt des Durchschnitts errechnen, so ist dies bei Verwendung von Rastermodellen wieder höchst einfach: wir summieren einfach alle Rasterzellen, die in beiden Flächen auftreten (Abb. 2.7). In einem Vektormodell benötigen wir jedoch komplexe Schnitt- und Flächenberechnungsalgorithmen.

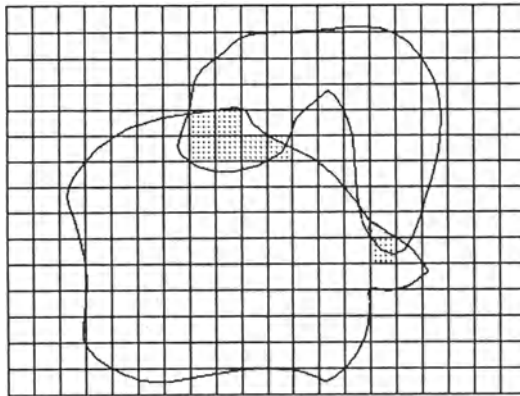


Abb. 2.7. Verschneidung im Rastermodell

Die implizite Herstellung einer Verbindung über die geometrisch-topologische Nachbarschaft hat also entscheidende Vorteile. Sie kann aber im Bereich der *Thematik* auch entscheidende Nachteile mit sich bringen. Obzwar wir thematische Strukturen erst im nächsten Kapitel behandeln, wollen wir hier vorwegnehmen, daß thematische Beziehungen sehr oft von der Geometrie unabhängig sind.

Denken wir etwa an den oder die Eigentümer eines Grundstückes. Selbst die Grundstücke eines Eigentümers liegen meist räumlich verstreut.

Thematische Beziehungen sind auch meist vielschichtiger als geometrisch-topologische Beziehungen. Wir können sie uns als ein Gewirr von Fäden vorstellen, die einzelne thematische Begriffe untereinander verbinden. Fäden sind eindimensional, und tatsächlich eignen sich Vektormodelle besser für thematische Beziehungen.

Natürlich ist damit nicht gesagt, daß dieser Umstand uns nun auch zwingt, unser topologisches Modell vektormäßig aufzubauen; denn erstens können (und sollen) die Geometrie und Topologie streng von der Thematik getrennt bleiben; und zweitens können wir immer Übergänge zwischen den beiden Varianten schaffen, indem wir ein Vektormodell rastern oder ein Rastermodell vektorisieren (siehe Kapitel 6). Aber letztlich wird die Entscheidung für ein bestimmtes topologisches Modell von der darauf aufbauenden Thematik mitgeprägt.

2.3. Die Geometrie und Topologie in Vektormodellen

2.3.1. Der Punkt

In Vektormodellen ist der Punkt der *Träger der geometrischen Information*. Alle höheren Strukturen (Linien, Flächen usw.) bauen auf dem Punkt auf. Ebenso lassen sich aus den Koordinaten der Punkte sämtliche geometrischen Aussagen für höhere Strukturen ableiten, wie etwa die Länge von Verbindungen, der Flächeninhalt, der Abstand zweier geometrischer Figuren usw.

In vielen Anwendungen sind die Koordinaten die einzigen Bestandteile an Information, die an einem Punkt interessieren. In vermessungstechnischen Anwendungen jedoch kommen zu den Koordinaten noch eine Reihe anderer Informationen hinzu, die als **Punktattribute** bezeichnet werden können:

- die **Punkthöhe** (falls sie nicht schon in den Koordinaten subsumiert war),
- die **Punktnummer** (die oft auch als Schlüssel verwendet wird),
- der **Punkttyp** (trigonometrischer Punkt, Polygonpunkt, Detailpunkt, photogrammetrischer Punkt, digitalisierter Punkt, konstruierter Punkt usw.),

- der **Punktfehler** (die zugestandene Genauigkeit) für die Lage bzw. die Höhe,
- die **Verlässlichkeit** (kontrolliert, nicht kontrolliert),
- die **Fixierung** (darf der Punkt verändert werden oder nicht?),
- das **Datum** der Erfassung bzw. der letzten Änderung und der Name des jeweiligen Bearbeiters,
- "**historische**" Koordinaten (die Geschichte des Punktes),
- **zusätzliche Bemerkungen.**

Im Interesse der Konsistenz des Modells müssen wir fordern, daß die Koordinaten einen Punkt eindeutig festlegen, daß es also nicht an ein- und derselben Stelle im Raum zwei verschiedene Punkte geben kann. Die Umkehrung, nämlich mehrere Koordinatenpaare bzw. -tripel pro Punkt, können wir hingegen zulassen.

2.3.2. Die Kante

In einem Vektormodell werden Punkte zu linienhaften Strukturen verbunden. Neben den (nulldimensionalen) Punkten tritt daher noch als deren (eindimensionales) Analogon die Kante auf. Sie verbindet jeweils zwei Punkte miteinander. Durch diese Verbindung wird eine topologische Beziehung zwischen den beiden Punkten hergestellt, eben die Beziehung des "Verbunden-Seins" (Adjazenz von Knoten bzw. Inzidenz von Kanten).

Diese topologische Beziehung kann mehrere geometrische Ausprägungen haben. So denken wir in erster Linie zunächst an eine geradlinige Verbindung zwischen den Punkten. Es ist aber ebenso eine kreisbogenförmige Verbindung denkbar, die, vom topologischen Standpunkt aus gesehen, mit der ersteren Verbindung identisch ist. Alle Verbindungskurven zwischen zwei Punkten sind zueinander **topologisch äquivalent**, d.h. sie können stetig ineinander abgebildet werden. Sie unterscheiden sich durch ihren **Formparameter**. Ein Beispiel: ein Kreis ist zu einem Quadrat topologisch äquivalent, ebenso zu einer Ellipse, nicht aber zu einem Kreisring. Eine treffende Formulierung lautet: "*Topologie ist das, was von einem Luftballon übrigbleibt, dem die Luft ausgeht*" (Frank 1983a).

Ein weiteres vertrautes Beispiel für die Hervorhebung topologischer Aspekte und die Unterdrückung der Geometrie ist die schematische Darstellung des Netzes innerstädtischer Verkehrsmittel (Abb. 2.8). Die wesentliche Information einer solchen Darstellung besteht darin, wie man vom Punkt A zum Punkt B kommt. Jene Haltestellen, wo sich zwei oder mehrere Linien kreuzen, nehmen die Rolle von Knoten an, die Teilstrecken zwischen diesen Knoten sind die Kanten. Die Geometrie spielt in diesem Modell eine völlig untergeordnete Rolle: die Lage der Knoten, ihre Entfernung voneinander, sowie die Form der Kanten stellt nicht unbedingt ein maßstabsgetreues Abbild der Wirklichkeit dar. Ändert sich beispielsweise die Geometrie, weil eine Autobuslinie in einem Streckenabschnitt in eine Parallelstraße verlegt wird, so bleibt das topologische Modell gleich - es sei denn, es entstehen dadurch neue Kreuzungspunkte und Verschneidungen mit anderen Linien. Ebensowenig äußert sich ein gerader oder kurvenreicher Verlauf einer Strecke in der Topologie, wohl aber eine Streckenunterbrechung oder ein in sich geschlossener Rundkurs.

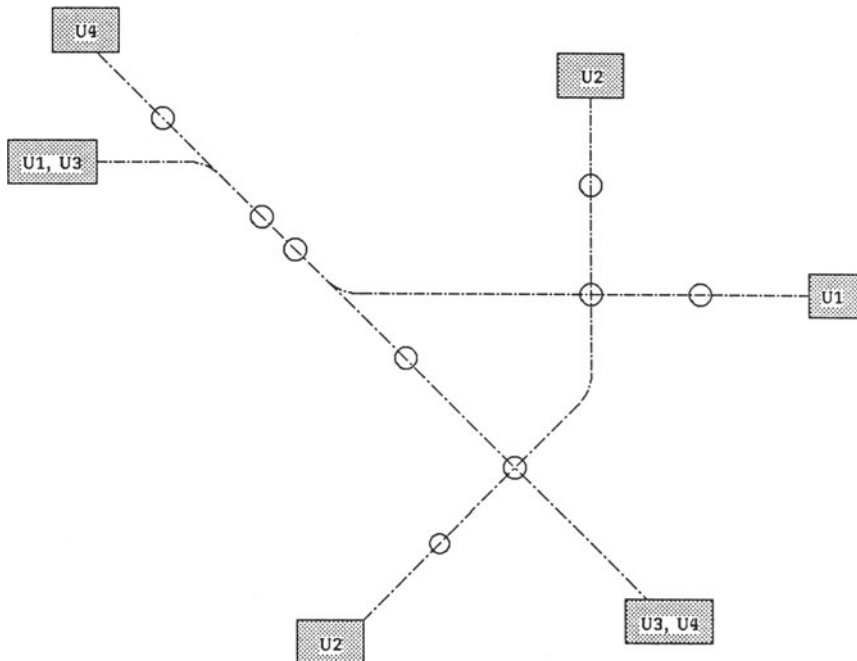


Abb. 2.8. Topologie eines städtischen Verkehrsnetzes

Man kann den Kanten eine Richtung als zusätzliches Attribut zugestehen. Diese Richtung hat nichts mit der intern natürlich immer vorhandenen Reihenfolge in der Speicherung der beiden Endpunkte einer Kante zu tun; sie hängt vielmehr von den jeweiligen Anwendungen ab. So ist eine Richtung von Kanten im Bereich eines Wasserleitungsnetzes durchaus sinnvoll, während dies bei den Kanten eines Grundstückkatasters nicht der Fall ist. Die Richtung ist demnach ein thematisches Attribut und sollte dementsprechend behandelt werden (siehe Kapitel 3).

Auch ein Polygonzug kann als Kante interpretiert werden, die den Anfangspunkt mit dem Endpunkt verbindet. Ein solcher Polygonzug hat im allgemeinen auch Zwischenpunkte. Denken wir an das Beispiel eines Polygonnetzes, wo sich mehrere Polygonzüge in bestimmten Punkten verknöten. In einem solchen Netz teilen wir meist die Punkte in zwei Kategorien: in **Knoten** und in **Zwischenpunkte**. Letztere sind dadurch gekennzeichnet, daß von ihnen genau zwei Verbindungen ihren Ausgang nehmen. Alle anderen Punkte sind Knoten. Zwischenpunkte werden anders als Knoten behandelt. So kann man für viele Netzuntersuchungen die Zwischenpunkte vernachlässigen. Auch in unserem Fall spielen sie nur die Rolle eines weiteren Formparameters. Sie "stützen" die Verbindung zwischen zwei Knoten.

Das Konzept der Kanten und Knoten (oft auch als **Kanten-Knoten-Struktur** bezeichnet) kommt aus der Graphentheorie. Damit ist es uns möglich, die wesentlichen topologischen Aspekte eines Modells hervorzuheben (Knoten und deren Verbindungen). In einer Kanten-Knoten-Struktur ist der *Punkt* (bzw. Knoten) der *Träger der geometrischen Information* und die *Kante* der *Träger der topologischen Information*.

Ebenso wie für Punkte müssen wir auch für Kanten die Eindeutigkeit der Lage fordern. Ein (teilweise) gemeinsamer Verlauf von Kanten ist daher nicht erlaubt. Der Grenzverlauf zwischen zwei benachbarten Grundstücken wird durch *eine* Kante beschrieben. Ebenso muß der Fall, daß ein Haus direkt an der Grundstücksgrenze steht, durch eine entsprechende Aufteilung in Teilkanten gelöst werden. Wir sprechen in diesem Zusammenhang von der **Eindeutigkeit von Kanten**: an einer Stelle des Raumes darf nur eine Kante sein. Dies ist eine Verallgemeinerung der Forderung nach der Eindeutigkeit von Punktkoordinaten. Zur Veranschaulichung betrachten wir nochmals Abb. 2.8: Die Linien 1, 3 und 4 haben streckenweise einen gemeinsamen Verlauf. Es darf also dort nur *eine* Kante geben.

2.3.3. Die Metrik in einer Kanten-Knoten-Struktur

Um Abstände, Flächen und dergleichen berechnen zu können, benötigen wir eine Metrik. Der Abstand zwischen Punkten bereitet keine Probleme, wohl aber muß der Abstand zwischen Kanten neu definiert werden, denn erstens sind Kanten nicht unbedingt geradlinig, und selbst wenn sie geradlinig sind, bestehen sie nur aus einem endlichen Teilstück einer Geraden.

Da wir die uns geläufige *Euklidische Metrik* nicht mehr anwenden können, benötigen wir einen neuen Abstandsbegriff. Er muß den Gesetzen einer Metrik genügen:

- Der Abstand zwischen zwei Elementen darf nie negativ werden. (Er muß *positiv-definit* sein.)
- Der Abstand zwischen zwei Elementen A und C darf nie größer als die Summe der Abstände zwischen A und B sowie B und C werden. (Die *Dreiecksungleichung* muß gelten.)

Eine mögliche Definition für den Abstand eines Punktes von einer Kante ist durch folgende Formel gegeben:

$$\text{dist}(\text{Punkt}, \text{Kante}) = \min [\text{dist}(\text{Punkt}, \text{Fußpunkt})] \quad (2-3-1)$$

Das Minimum ist über alle möglichen Fußpunkte längs der Kante zu erstrecken. In Sonderfällen kann dies zur orthogonalen Projektion auf die Kante führen, so wie wir dies von der Euklidischen Metrik her kennen. Es können aber auch andere Fälle eintreten, wie die Beispiele in Abb. 2.9 zeigen. In ähnlicher Weise kann man auch den Abstand zwischen zwei Kanten definieren. Zwei Kanten, die einander schneiden, haben in diesem Sinn einen Abstand "Null".

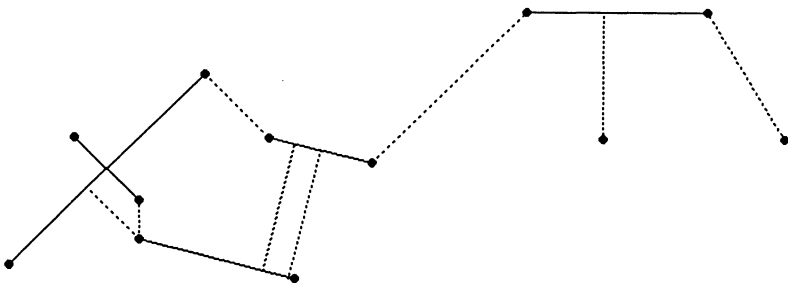


Abb. 2.9. Metrik von Kanten und Knoten

Ausgerüstet mit diesem Abstandsbegriff können wir nun etwa den Abstand einer Hochspannungsleitung von einem Haus berechnen oder die Frage klären, welche Grundstücke aufgrund der Neuprojektierung einer Straße abgelöst werden müssen (weil sich eine Grundstückskante mit einer Straßenkante schneidet, der Abstand in unserem Sinn daher Null ist), und welche Grundstücke eine Wertminderung erfahren, weil sie in einem Pufferbereich von 50 m liegen.

2.3.4. Flächen und Inseln

In einem Vektormodell werden Flächen meist auf ihre Umrißpolygone bezogen. Wird eine Kanten-Knoten-Struktur im strengen Sinn ausgelegt, so sind Flächen implizit über ihre umgebenden Kanten definiert und brauchen nicht als eigenständige Einheiten behandelt zu werden. Alle Fragen, die eine solche Fläche betreffen, können über die umgebenden Kanten und deren Knoten bzw. Zwischenpunkte gelöst werden.

So können wir etwa die Einteilung eines Staates in Länder durch eine Kanten-Knoten-Struktur beschreiben (Abb. 2.10). Die Knoten sind die "Dreiländerecken", während die Zwischenpunkte zwar für die Geometrie (z.B. für die Flächenberechnung) wichtig sind, nicht aber für die Topologie: denn es entscheidet sich an den Dreiländerecken, welche Länder aneinander grenzen; die Zwischenpunkte haben darauf keinen Einfluß mehr.

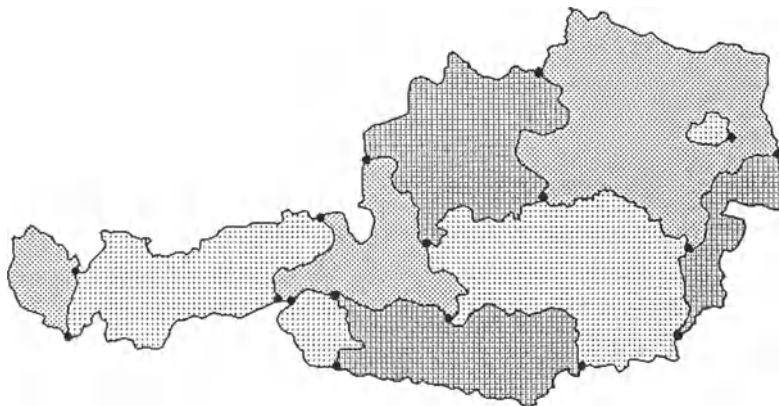


Abb. 2.10. Beispiel einer Kanten-Knoten-Struktur

Die Einfärbung einer Landkarte mit möglichst wenigen Farben, so daß aneinandergrenzende Länder verschieden gefärbt sind, ist somit ein topologisches Problem. Ändert man die Form

der Verbindungskanten, so ändert sich zwar die Geometrie (z.B. der Flächeninhalt), nicht aber die Topologie (z.B. die Nachbarschaft), es sei denn, man verschiebt eine Kante so weit, daß sie eine andere Kante schneidet; an den Schnittpunkten entstehen neue Knoten, und damit eine neue Topologie.

Die Genugtuung darüber, daß wir zwei Fliegen mit einem Streich erledigt haben (eindimensionale und zweidimensionale Modellteile werden gleichermaßen durch Kanten beschrieben), wird entscheidend geschmälert, wenn wir uns bewußt werden, daß Inseln in unserem Modell unberücksichtigt bleiben.

Inseln sind Aussparungen innerhalb einer Fläche, die keine Verbindung (d.h. Kante) zum umgebenden Polygon haben. Beispiele für Inseln sind in unserem Anwendungsgebiet leicht zu finden: Bundesländer, die ganz innerhalb eines anderen Landes liegen (Wien / Niederösterreich in Abb. 2.3), aber auch die unbebaute und die bebaute Teilfläche eines Grundstückes, die Lichtungen in einem Waldgebiet, Industriezonen inmitten einer Agrarlandschaft, natürlich auch Inseln im ursprünglichen umgangssprachlichen Sinn des Wortes.

Eine Fläche kann auch mehrere Inseln haben (wie an den Beispielen der Länder Niedersachsen / Bremen oder der Kantone St. Gallen / Appenzell ersichtlich ist). Eine Insel kann selbst wieder Inseln haben (auf einer Waldlichtung steht eine Baumgruppe). Inseln prägen entscheidend die Topologie ihres "Umlandes". Wie wir bereits weiter oben festgestellt haben, sind ein Kreis und ein Kreisring topologisch verschiedene Gebilde.

Im Sinne unseres Vektormodells können wir eine Insel natürlich wieder nur durch deren Umgebungskanten repräsentieren. Sie können als innerer Ring der Umgebungsfläche aufgefaßt werden. Jede Fläche hat somit genau einen **äußeren Ring** und keinen, einen oder mehrere innere Ringe (Abb. 2.11). Die Bezeichnung "Ring" steht hier für eine geordnete in sich geschlossene Kantenfolge, die topologisch zu einem Kreis äquivalent ist.

Wir müssen nun einen Bezug zwischen dem äußeren Ring und den inneren Ringen einer Fläche herstellen. Dazu gibt es mehrere Möglichkeiten:

1) Wir führen **Pseudokanten** ein, die einen Knoten des äußeren Ringes mit einem Knoten eines inneren Ringes verbinden. Dies entspricht der Einführung eines zusätzlichen Formparameters "pseudo". Eine solche Kante muß genau zweimal, und zwar jeweils in entgegengesetzter Richtung, durchlaufen werden. Wenn wir nun vereinbaren, daß der äußere Ring immer *im Uhrzeigersinn*, und innere Ringe immer *gegen den Uhrzeigersinn* durchlaufen werden, und ferner im

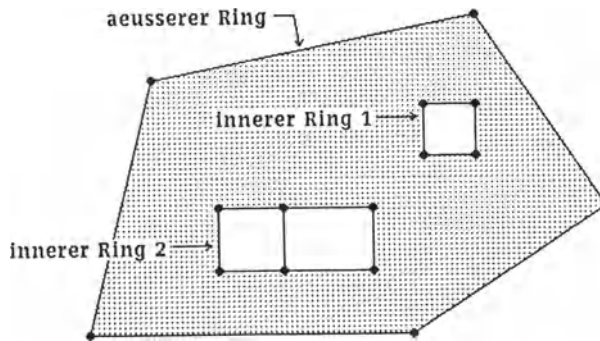


Abb. 2.11. Äußerer Ring und innere Ringe einer Fläche

Fall von Mehrdeutigkeiten in einem Knoten immer jene Kante als nächste verfolgt wird, die mit der zuletzt durchfahrenen Kante den *kleinsten Brechungswinkel* einschließt, so können wir in einem einzigen Durchgang alle inneren und äußeren Begrenzungen einer Fläche durchlaufen (Abb. 2.12). Diese Vorgangsweise ermöglicht das problemlose Funktionieren vieler Flächenberechnungs- und Schraffurprogramme auch im Fall des Auftretens von Inseln.

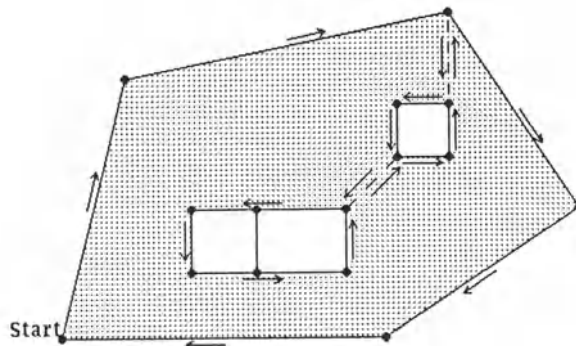


Abb. 2.12. Durchlaufen von Kanten und Pseudokanten

2) Wir führen Flächen als eigenständige Elemente in unserem Vektormodell ein. Damit gelangen wir zu einer Dreierbeziehung Knoten - Kante - Fläche, in der folgende Abhängigkeiten bestehen:

- jede Kante wird von zwei Knoten begrenzt;
- in jedem Knoten beginnen bzw. enden mehrere Kanten;
- jede Kante hat eine linke und eine rechte Fläche (siehe Abb. 2.13);
- jede Fläche wird von einem äußeren Ring und mehreren inneren Ringen begrenzt.

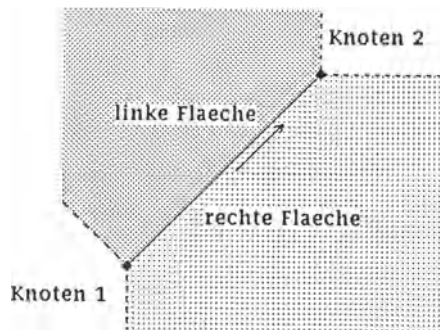


Abb. 2.13. Zusammenspiel zwischen Knoten, Kanten, Flächen

Damit sind implizit auch noch folgende Beziehungen gegeben:

- jeder Knoten ist Eckpunkt mehrerer Flächen;
- jede Fläche hat mehrere (innere und äußere) Knoten.

3) Bisher haben wir die Beziehung zwischen einer Fläche und ihren Inselflächen (ein zweidimensionales Problem) auf dem Umweg über (eindimensionale) Kanten gelöst. Wir haben aber auch die Möglichkeit, diese Beziehung direkt zwischen Flächen herzustellen. Wir führen also die Beziehungen zwischen Knoten, Kanten und Flächen wie in Variante 2) ein, mit der Ausnahme, daß wir nur eine Beziehung zwischen der Fläche und ihrem äußeren Ring herstellen. Zusätzlich vermerken wir noch zu jeder Fläche die sie umgebende Fläche (diese Beziehung ist eindeutig, d.h. zu jeder Fläche kann es höchstens eine Umgebungsfläche geben, während die Umkehrung nicht gilt, denn jede Fläche kann mehrere Inseln haben).

4) Schließlich sei auch die Möglichkeit erwähnt, eine Fläche nicht durch (äußere und innere) Kanten zu repräsentieren, sondern durch ihr **Skelett** (Abb. 2.14). Das Skelett ist der geometrische Ort aller internen Flächenpunkte, bei denen das Minimum aller Entfernungen zu inneren bzw. äußeren Begrenzungskanten für mindestens zwei dieser Kanten gleichzeitig eintritt.

Die Generierung eines Skelettes kann man sich als einen Prozeß vorstellen, der die Begrenzungslinien der Fläche wellenartig in das Innere fortsetzt. Dabei hält man jede Kante an den Endknoten fest und baucht sie nach innen aus, bis sie an eine andere Wellenfront stößt. Das Skelett ergibt sich als Grenzfall dieses Prozesses. Für die Wellenfronten verwendet man Polygone oder Umhüllende von Kreisbögen (Brassel et al. 1984).

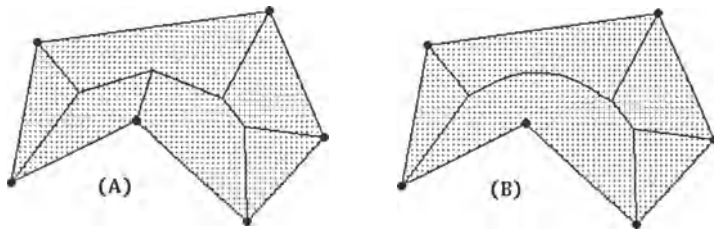


Abb. 2.14. Skelett einer Fläche. Bildung über polygonartige (A) oder kreisbogenförmige Wellenfronten (B)

Das Skelett spiegelt gut die topologische Beschaffenheit der Fläche wider. So gibt etwa das Vorhandensein von Zyklen Aufschluß über Inselflächen. Die Verästelungen des Skelettes dienen als Brücken sowohl zu Nachbar- wie auch zu Inselflächen. Ein Skelett bietet die Möglichkeit, die Topologie einer Struktur in kompakter Weise (in Kurzschrift) darzustellen. Die eigentliche Topologie kann aus einem Skelett jederzeit (durch Umkehrung der oben erwähnten Algorithmen) rekonstruiert werden.

Es sei hier noch angemerkt, daß die Skelettierung auch in Rastermodellen verwendet wird und dort gute Dienste leistet. Wir werden uns in Abschnitt 2.4 mit der Skelettierung von Rastermodellen auseinandersetzen.

2.3.5. Analytisch beschreibbare Flächen

Flächen, deren Umriß sich durch einen analytischen Ausdruck angeben läßt (wie etwa die Kreisfläche, Ellipsenflächen usw.), nehmen eine Sonderstellung ein. In unseren Anwendungen treten Kreise meistens im Rahmen von kartographischen Symbolen auf. Kreise können aber beispielsweise auch als geometrisches Modell einer Schadstoff-Emissionszone gewählt werden.

Zur internen Repräsentation genügen wenige Parameter (z.B. Zentrum und Radius). Die Problematik von Inseln und Schnitten vereinfacht sich für solche analytisch beschreibbare Flächen wesentlich, so daß wir hier nicht näher darauf einzugehen brauchen. Allerdings ist anzumerken, daß die Komplexität der Datenstruktur zunimmt, wenn wir eine eigene Kategorie von Modellen für diese speziellen Flächen einführen. So müssen in sehr vielen Verarbeitungsschritten (zum Beispiel im Bereich der Schnittalgorithmen) eigene Programmverzweigungen geschaffen werden, die diese Kategorie mit allen anderen Kategorien in

Beziehung bringt. Viele Systeme vermeiden aus diesem Grund ein gesondertes Modell, sondern sehen Kreise beispielsweise als in sich geschlossene Kanten mit dem Formparameter "Kreisbogen" an. Natürlich ist dazu die Definition eines Kreispunktes nötig, der als Anfangs- und Endknoten auftritt, etwa der Punkt, der genau im Osten vom Kreiszentrum liegt. Dadurch wird aber wieder dessen Punktlage gegenüber den anderen Koordinaten auf der Peripherie künstlich aufgewertet; eine Aufwertung, die ihr aber aufgrund der analytischen Definition nicht zukommt. Man hat hier, wie in so vielen anderen Bereichen, Vorteile und Nachteile der verschiedenen Varianten gegeneinander abzuwägen.

2.3.6. Netze

Die vorhin besprochenen *elementaren geometrisch-topologischen Strukturen* Punkt / Knoten, Kante und Fläche können wir nun zu komplexen Gebilden zusammenfassen. So können wir etwa das Netz der Versorgungsleitungen eines Elektrizitätswerkes aus den Grundelementen Knoten und Kante aufbauen. Eine andere Variante eines solchen Netzes ergibt sich, wenn wir alle Grundstücke einer Gemeinde im Rahmen einer Kanten-Knoten-Struktur darstellen. Als dritte Variante betrachten wir schließlich die Dreiecksvermaschung eines Höhenmodells. Die drei Varianten sind in Abb. 2.15 auszugsweise wiedergegeben.

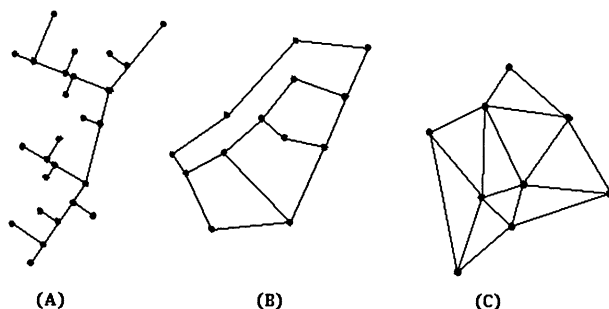


Abb. 2.15. Varianten topologischer Netze

Aus dem Vergleich der drei Figuren ergeben sich einige grundsätzliche Sachverhalte: im Grundstückskataster (B) sind wir an den Flächen interessiert, die von den Polygonen umrahmt werden; dabei dominieren die Flächen, während dem Umriß nur eine stützende Rolle zukommt. Grundstücke können auch Inseln haben. In der Dreiecksvermaschung (C) tritt die Bedeutung der Dreiecksfläche stark hinter die des Randes zurück; im wesentlichen wird die Information des Randes

(z.B. die Höhe der Eckpunkte des Dreieckes) in das Innere fortgesetzt. Andererseits können im Dreiecksnetz keine Inseln auftreten; das Netz ist topologisch *zusammenhängend*. Im Versorgungsnetz (A) hingegen treten im allgemeinen gar keine geschlossenen Polygone auf; dort, wo sie auftreten, ist dies eher zufällig, und die Thematik dieser Polygone setzt sich nicht in ihr Inneres fort, ja, sie haben gar kein Inneres.

Netze als komplexe Konglomerate einfacher Grundstrukturen zerfallen demnach in Gruppen, die jeweils unterschiedlich behandelt werden. Die drei Beispiele führen zu folgenden Schlüssen:

- es gibt linienförmige und flächenförmige Netze;
- bei flächenförmigen Netzen wird die Fläche unterschiedlich stark bewertet;
- die Entscheidung über die jeweils gültige topologische Variante (d.h. ob ein linienförmiges oder flächenförmiges Netz vorliegt, bzw. ob die Fläche als eigenständiger Teil des Modells behandelt werden soll oder nur über umgebende Kanten oder Skelette) geht immer von der Thematik aus.

Die bislang erwähnten Kriterien bilden nicht die einzigen Entscheidungsgrundlagen im Rahmen der von uns angestrebten Einteilung der Netze in Gruppen. So können wir etwa linienförmige Netze noch zusätzlich untergliedern in

- **zusammenhängende Netze** und
- **nicht zusammenhängende Netze.**

Geodätische Punktnetze beispielsweise müssen immer *zusammenhängend* sein, wenn sie einer Ausgleichung unterworfen werden sollen. Ebenso sollten Gewässernetze *zusammenhängen* (ausgenommen im Karstgebiet). Für Versorgungsleitungen müssen wir dies nicht fordern.

Bei flächigen Netzen wiederum kann es erforderlich sein, daß sie *flächendeckend* angelegt sind: jeder Punkt eines betrachteten Gebietes gehört genau zu einer Fläche, d.h. es gibt keine Überlappungen von Flächen, aber auch keine nicht überdeckten Restgebiete.

Ein interessanter Spezialfall eines flächendeckenden Netzes ist die **DIME-Struktur** (DIME ist eine Abkürzung für Dual Independent Map Encoding). Sie beinhaltet Knoten (sogenannte **0-Zellen**), Kanten (**1-Zellen**) und Flächen (**2-Zellen**) eines flächendeckenden Netzes. Alle diese Zellen sind durch eindeutige Nummern gekennzeichnet. Benachbarte Knoten - die durch eine Kante verbunden werden - und benachbarte Flächen - die durch eine Kante voneinander getrennt werden - stehen in einem Zusammenhang, den man als geordnetes **Quadrupel** schreiben kann:

Knoten 1, Knoten 2; Fläche A, Fläche B (2-3-2)

In Worten ausgedrückt bedeutet dies: "Die Knoten 1 und 2 werden durch eine Kante verbunden; geht man längs dieser Kante von 1 nach 2, so befindet sich die Fläche A zur rechten und die Fläche B zur linken Hand."

Dazu gibt es einen dualen Sachverhalt: "Überschreitet man die Grenze von der Fläche A in die Fläche B, so hat man den Knoten 1 zur linken und den Knoten 2 zur rechten Hand."

Wir können also zu jeder Kante eine dazu duale 1-Zelle finden, die das Überschreiten dieser Kante ausdrückt. Außerdem haben nun 0-Zellen und 2-Zellen die Rollen getauscht: im dualen Graphen werden die Flächen zu Knoten und die Knoten zu Flächen (Abb. 2.16).

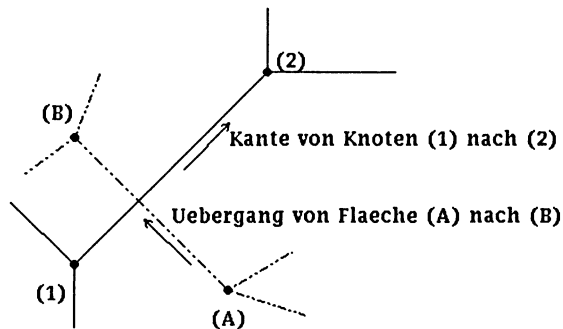


Abb. 2.16. 0-Zellen, 1-Zellen und 2-Zellen in zwei zueinander dualen Graphen

Duale Entsprechungen pflanzen sich auch auf abgeleitete Aussagen fort. Wollen wir etwa die Eindeutigkeit von Kanten überprüfen, so lautet die Frage im Sinne der DIME-Topologie: "Wird jede 1-Zelle von genau zwei 0-Zellen berührt?", und dazu ist die Frage "Wird jede 1-Zelle von genau zwei 2-Zellen berührt?" dual; sie überprüft, ob es zu jeder Kante genau eine linke und eine rechte Nachbarfläche gibt.

Der duale Graph zu einer gültigen Kanten-Knoten-Struktur wird oft auch als **Thiessen-Polygon** oder **Voronoi-Graph** bezeichnet. Im Rahmen der graphischen Datenverarbeitung wird auch der Ausdruck **Region adjacency graph (RAG)** verwendet. Der RAG dient zur expliziten Darstellung der Nachbarschaft von Flächen (Abb. 2.17).

Auch die im vorigen Abschnitt erwähnte Skelettmethode ist in diesem Zusammenhang zu erwähnen. Fügt man nämlich die Skelette aller Flächen eines Netzes zusammen, so entsteht eine duale Struktur (ein Skelett des gesamten Netzes), bei

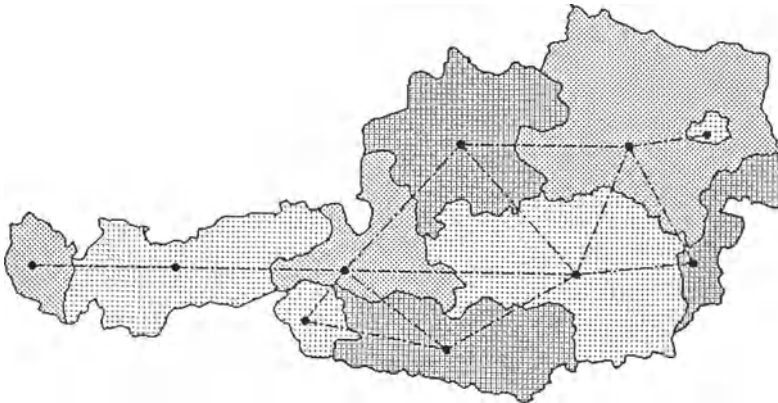


Abb. 2.17. Adjazente Regionen und ihr Graph (RAG)

der allerdings neben dem topologischen (graphentheoretischen) Aspekt auch noch der geometrische Aspekt im Rahmen der Metrik berücksichtigt ist.

2.3.7. Die Geometrie und Topologie von Symbolen

In der Kartographie wird dem kartographischen Symbol ein besonderer Stellenwert zugemessen, denn es dient einer kurzen und doch prägnanten Darstellung eines (möglicherweise komplexen) Sachverhaltes. Die Interpretation einer Karte bzw. eines Planes hängt im wesentlichen von einer geeigneten Symbolisierung ab. Ein Symbol ist also nichts anderes als eine Abstraktion eines thematischen Sachverhaltes, ein *Modell* eines Teiles der uns umgebenden Wirklichkeit. Je nach dem Verfeinerungsgrad des Modells können wir

- **Punktsymbole,**
- **linienförmige Symbole und**
- **Flächensymbole**

unterscheiden (Abb. 2.18). Die Grenzen sind fließend. Ein Staudamm etwa wird auf einer Übersichtskarte als Punktsymbol auftreten, während er in einem Detailplan sehr wohl ein-, zwei- und sogar dreidimensionale Charakteristika aufweist.

Diese (langfristig bedeutsame) Thematik muß im konkreten Anwendungsfall in eine - den jeweiligen Erfordernissen des Anwenders, des zu erreichenden Zieles und der vorhandenen Umgebung angepaßte - Darstellung münden. In der ingenieurtechnischen Vermessung wird man andere Symbole

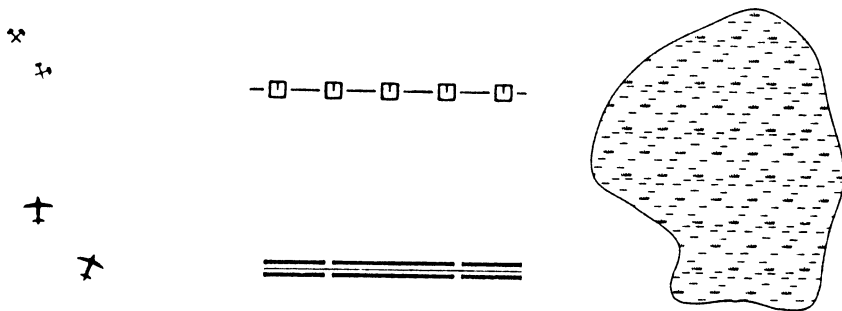


Abb. 2.18. Punkt-, Linien- und Flächensymbole

verwenden als bei der Erstellung von Straßenkarten. Aber auch das Anwenderprofil ist zu beachten; die Art der Darstellung wird sicherlich davon abhängen, ob wir eine in einem bestimmten Bereich spezialisierte Zielgruppe ansprechen wollen - als Beispiel sei ein Werkplan erwähnt -, oder ob wir uns an ein breites Publikum richten - wie etwa bei der Erstellung einer Wanderkarte. Schließlich sind auch noch Kriterien temporärer Bedeutung, wie Genauigkeitsanforderungen, Zeitverhalten oder Hardwareumgebung, ausschlaggebend.

Die Thematik des Symbols muß (im Falle eines Vektormodells) in eine **Figur** übersetzt werden, d.h. eine Substruktur von Punkten, Linien und Flächen, bzw. (im Fall eines Rastermodells) in ein Rasterzellenmuster. Wir haben den Begriff "Figur" geprägt, um eine Abhebung von dem mit der Thematik behafteten Begriff "Symbol" zu erreichen.

Die Figur ist das (topologische) Pendant zum (thematischen) Symbol. Bezieht sie sich auf einen Punkt (Beispiel: Symbol für einen trigonometrischen Punkt), so ist sie zentriert einzusetzen; andere in diesem Punkt einmündende Kanten sind entsprechend abzuschneiden (die damit zusammenhängende Problematik wird als **Symbolfreistellung** bezeichnet). Bei Liniensymbolen (Beispiel: Autobahnsignatur) sind **Verdrängungsprobleme** zu lösen: Signaturen dürfen einander nicht überdecken, daher müssen wir ihre Lage verfälschen. Bei Flächensymbolen wieder ist darauf zu achten, daß die entsprechenden Figuren auch tatsächlich innerhalb der angesprochenen Fläche liegen; gegebenenfalls müssen wir die Figur am Rand der Fläche schneiden. Wir werden auf die Problematik der Symbolfreistellung in Kapitel 6 zu sprechen kommen.

Alle eben angesprochenen Probleme sind topologischer Natur, und aus diesem Grund können wir die Figur in die Reihe der bis jetzt besprochenen topologischen Elemente Knoten, Kante, Fläche, Netz einreihen. Wir müssen andererseits aber auch

den grundlegenden Unterschied zu diesen Elementen herausstreichen. Es handelt sich hier durchwegs um darstellungsspezifische (also nur kurzfristig bedeutsame) geometrisch-topologische Modelle, während Knoten, Kante, Fläche und Netz langfristig relevante Zusammenhänge darstellen.

2.3.8. Die Geometrie und Topologie von Texten

Die Problematik von Texten im Bereich der Modellierung raumbezogener Informationen stellt sich in mancher Hinsicht ähnlich wie bei den Symbolen dar. Texte beschreiben eine Thematik (z.B. eine Punktnummer, eine Hausnummer, die Beschriftung einer Straße, der Name eines Gebirgszuges usw.). Diese Thematik wird in Form von Attributen modelliert und hat eine langfristige Bedeutung. Im Rahmen eines konkreten Anwendungsfalles wird die Thematik in einen darzustellenden Text umgesetzt, der einem bestimmten Knoten als Punktnummer, einer bestimmten Kante als Straßennamen, einer bestimmten Fläche als Flurname zugeordnet wird.

Das (thematische) Attribut verhält sich zum (darstellungsspezifischen) Text ebenso wie vorhin das Symbol zur Figur. Dementsprechend gibt es

- **punktbezogene Texte,**
- **Beschriftungen von Linien und**
- **Beschriftungen von Flächen (Abb. 2.19).**



Abb. 2.19. Beschriftungen für Punkte, Linien und Flächen

Die Platzierung von Texten ist wieder ein topologisches Problem. Punktbezogene Texte sollen "in der Nähe" des jeweiligen Knotens liegen, Beschriftungen von Linien sollen sich dem Linienverlauf anschmiegen, Beschriftungen von Flächen sollen nicht außerhalb der Flächen liegen.

Zusätzlich müssen wir fordern, daß Texte einander nicht überdecken, und daß "möglichst wenige" geometrisch-topologische Informationen unter der Plazierung des Textes leiden; diese müssen dann nämlich ausgeblendet (d.h. abgeschnitten) werden. Die damit zusammenhängende Problematik wird als **Textfreistellung** bezeichnet. Sie ist um einiges schwieriger zu lösen als die Symbolfreistellung, da Texte im allgemeinen mehr Platz benötigen als Symbole. Wir werden auf die Problematik der Textfreistellung in Kapitel 6 zu sprechen kommen.

Ähnlich wie die Figuren lassen sich auch Texte in die Reihe der bis jetzt besprochenen topologischen Elemente Knoten, Kante, Fläche, Netz einreihen. Und auch hier müssen wir den Vorbehalt anmelden, daß es sich durchweg um darstellungsspezifische (also nur kurzfristig bedeutsame) geometrisch-topologische Modelle handelt, während Knoten, Kante, Fläche und Netz langfristig relevante Zusammenhänge darstellen.

2.3.9. Zusammengesetzte geometrisch-topologische Strukturen

Für ein umfassendes Modell der Geometrie und Topologie eines raumbezogenen Informationssystems fehlt uns noch die Möglichkeit, einzelne Teile der Geometrie zu **Blöcken** zusammenzufassen. Wir wollen diese Blöcke als Ganzes ansprechen können. Dies ist im Bereich des CAD eine häufig gebrauchte Arbeitsweise; man beginnt etwa bei den elementaren Bestandteilen einer Konstruktionsaufgabe. Nachdem man diese gelöst hat, kann man daraus Werkstücke aufbauen, die ihrerseits wieder in größere Blöcke zusammengefaßt werden können. Damit hat man die Möglichkeit, im Sinne einer **Makrotechnik** sehr rasch und sehr effizient hierarchisch angeordnete Strukturen zu schaffen.

Diese Arbeitsweise ist in unserem Bereich nur bedingt einsetzbar. Zum einen beschreibt CAD "das, was sein soll", während uns eine Beschreibung des Ist-Zustandes am Herzen liegt. CAD macht also im wesentlichen eine Synthese, während wir eine Analyse der Wirklichkeit machen und versuchen, diese dann in unserem Modell zu rekonstruieren; und die Wirklichkeit eignet sich wenig für regelmäßige Duplikate: bei einem CAD-Werkstück gibt es viele Schrauben, die einander ähneln; Häuser, Leitungen, Grenzen, Flüsse sind alle voneinander verschieden.

Das Arbeiten mit Blöcken ist jedoch in gewissen (eingeschränkten) Bereichen durchaus sinnvoll; so wieder bei der Plangestaltung, d.h. bei der aktuellen Ausprägung bzw. Umsetzung der Themen in unterschiedliche Darstellungsaspekte. Hier kann man sich vorstellen, daß beispielsweise

im Bereich der **Bemaßung** eine Effizienzsteigerung erreicht werden kann, wenn wir Blöcke verwenden (Abb. 2.20). So können wir etwa die Beschriftungen und Hilfslinien, die bei der Bemaßung eines Leitungsstranges auftreten, als Block auffassen. Dessen Platzierung ist wieder ein topologisches Problem. Er kann als Ganzes verschoben werden, so lange, bis den topologischen Kriterien Genüge getan worden ist.

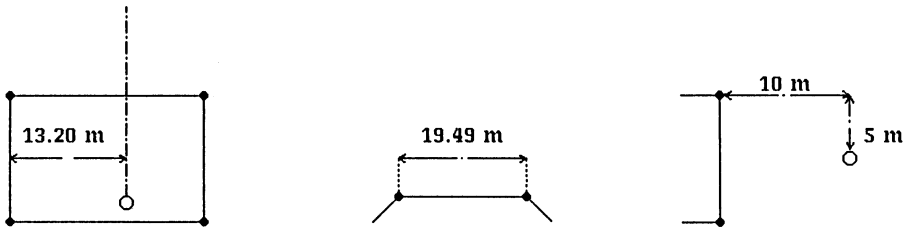


Abb. 2.20. Bemaßungsblöcke

2.3.10. Geometrisch-topologische Bearbeitungsauszüge

Beim graphisch-interaktiven Arbeiten in raumbezogenen Informationssystemen werden sich bestimmte Konfigurationen bezüglich der Geometrie und Topologie herauskristallisieren, die sehr häufig angesprochen werden. Viele Anwender werden ihre Abfragen immer innerhalb eines bestimmten Fensters tätigen. Wir können die Effizienz des Systems daher steigern, wenn wir erlauben, daß jeder Anwender ein solches Fenster - oder eine beschränkte Zahl dieser Fenster - definiert, die für ihn bedeutungsvoll sind. Diese *benützerspezifische Sicht* der Daten wird auch als View oder Bearbeitungsauszug bezeichnet.

Dabei werden die Daten nicht physisch umgruppiert oder kopiert, es wird lediglich ein logischer Zusammenhang definiert. In den Kapiteln 4 und 7 gehen wir näher auf Views und ihre Bedeutung für die Datenunabhängigkeit ein. Es sei auch erwähnt, daß wir für thematische Zusammenhänge ebenfalls Bearbeitungsauszüge definieren können.

Wurde ein Bearbeitungsauszug einmal definiert, so kann er rasch reaktiviert werden. Mehrere solcher Views können auch hintereinander gruppiert werden und dann in Form einer Präsentation von Diapositiven (Slides) abgerufen werden.

2.4. Die Geometrie und Topologie in Rastermodellen

Geometrische und topologische Sachverhalte sind in Rastermodellen um vieles einfacher darzustellen als in Vektormodellen. Es gibt nur ein geometrisches Grundelement im Rastermodell, die Rasterzelle (manchmal in Anlehnung an die Anwendungen in der graphischen Datenverarbeitung und der Bildverarbeitung als *Pixel* oder *Voxel* - je nach der Dimension des Modells - bezeichnet).

Die Rasterzelle ist rechteckig und überdeckt ein Gebiet mit homogener Bedeutung. Alle Rasterzellen sind gleich groß und unterliegen einer regelmäßigen Anordnung. In einem Satellitenbild etwa kann man die Rasterzellen zur Darstellung homogener Grau- oder Farbwerte heranziehen. Wir haben aber bereits in den einleitenden Abschnitten erkannt, daß es auch allgemeinere Bedeutungen geben kann, denen man eine Rasterung zugrunde legt, wie zum Beispiel diverse Statistiken, die relativ grob für rechteckige Bereiche gegeben sind (Waldschäden, Bevölkerungsstatistik, Infrastruktur).

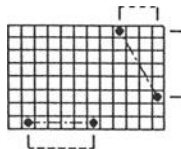


Abb. 2.21. Geometrie in Rasterzellen

Sind die Abmessungen einer Rasterzelle einmal festgelegt, so sind damit auch viele geometrische Probleme aus der Welt geschafft. Positionen, Distanzen und Winkel sind leicht in ganzzahlige Vielfache der elementaren Maschenweite umzurechnen; die Zeilen- und Spaltenposition der jeweiligen Zelle ist daraus ableitbar (Abb. 2.21). Außerdem ist eine Rasterzelle eine flächige Struktur; daraus folgt, daß alle geometrischen Abfragen, die Flächen betreffen (Schnitte, Auswertungen usw.) leicht durch Summation bzw. Differenz von Elementarflächen befriedigt werden können.

Der entscheidende Engpaß eines Rastermodells ist die erreichbare Genauigkeit. So gilt als Faustregel, daß sich die räumliche Auflösung (*Sampling*) bei einer Rasterung nach dem kleinsten Element richtet, das noch dargestellt werden soll. Eine elementare Zelle soll halb so groß sein wie dieses Element. Der damit verbundene enorme Speicherplatzbedarf läßt sich zwar durch Verwendung geeigneter Kompressionstechniken reduzieren, stellt aber immer noch das größte Hindernis für eine globale Anwendung des Rastermodells dar.

Während viele geometrische Operationen einfach durchzuführen sind, gibt es aufgrund der starren Nord-Süd- bzw. West-Ost-Ausrichtung der Rasterzellen Probleme bei Verdrehungen (siehe Abschnitt 6.4).

Für den topologischen Zusammenhang eines Rastermodells gibt es - so wie beim Vektormodell - eine Reihe von Grundbausteinen, auf welche wir in der Folge näher eingehen. Sie sind nicht so ausgeprägt wie beim Vektormodell, was sowohl als Vorteil (größere Einfachheit der Algorithmen) als auch als Nachteil (Einschränkung der Allgemeinheit) empfunden werden kann.

Linienhafte Strukturen sind in einem Rastermodell nur als zusammenhängende Folgen von Rasterzellen darstellbar. Ausgehend von einer Anfangszelle wird die Adresse (also Zeilen- und Spaltenposition) der nächstfolgenden Zelle vermerkt. Dabei kann man verschiedene Metriken zugrunde legen. So kennt man etwa in der Häuserblockmetrik ("city block metric", in Anlehnung an das Bild amerikanischer Großstädte) nur vier Nachbarn (Nord, Ost, Süd, West), während in der Schachbrettmetrikt ("chessboard metric") auch die Himmelsrichtungen Nordost, Südost, Südwest und Nordwest möglich sind (Abb. 2.22).

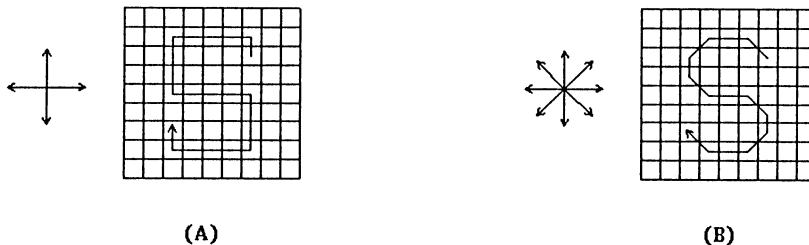


Abb. 2.22. Beispiele für Rastermetriken: City-Block (A) und Schachbrett (B)

Ordnen wir nun jeder Himmelsrichtung eine Zahl zu, so können wir linienhafte Strukturen durch eine Anfangsposition und durch eine darauffolgende Zahlenkette modellieren. Diese Strategie wird als **chain coding** (Kettencode) bezeichnet (Abb. 2.23).

Sind die Strukturen streng in horizontaler oder vertikaler Richtung angeordnet, so wird oft auch ein **Run length encoding** (Lauf­längen­kodierung) durchgeführt. Dabei nutzt man den Umstand aus, daß aneinanderstoßende Zellen sehr oft auch die gleiche Bedeutung besitzen. Jedesmal wenn eine Bedeutung wechselt, genügt es daher, die neue Bedeutung und die Anzahl der darauffolgenden Zellen mit dieser Bedeutung zu speichern (Abb. 2.24).

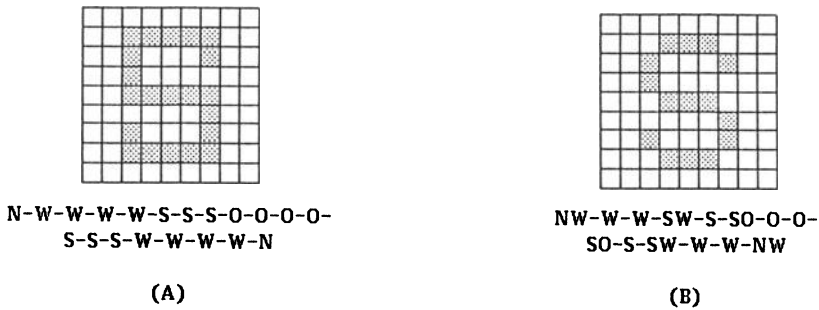


Abb. 2.23. Chain code für Cityblockmetrik (A) und Schachbrettmantik (B)

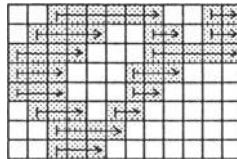


Abb. 2.24. Run length encoding von Rasterzellen

Flächige Sammlungen von Rasterzellen werden effizient durch eine **Baumstruktur** modelliert (Abb. 2.25). Dabei werden dieselben Konzepte angewandt, wie sie in Kapitel 5 erläutert werden. Allerdings gibt es hier jeweils nur zwei Alternativen für die Belegung einer Zelle ("ja", d.h. sie gehört zur Struktur, die modelliert werden soll, oder "nein", d.h. sie gehört zum Hintergrund).

Natürlich können wir Flächen auch - ähnlich wie in Vektormodellen - durch ihre linienhafte Umgebung darstellen. Diese Umgebung kann ihrerseits etwa durch chain codes modelliert werden. Ebenso wäre die Einbeziehung von Inseln durch die Einführung innerer Ringe möglich. Diese Art der Darstellung nützt jedoch die Möglichkeiten des Rastermodells zu wenig aus und wird weniger häufig angewandt.

Ganz anders jedoch das **Skelett** (Abb. 2.26). Wir haben das Skelett bei der Besprechung des Vektormodells als eine interessante - wenn auch seltener gebrauchte - Alternative kennengelernt. Im Rastermodell ist das Skelett eine effiziente "Kurzdarstellung" einer Fläche, ein steno-graphisches Kürzel, aus dem sich die ursprüngliche Fläche mittels einfacher und automatisierbarer Operationen rekonstruieren läßt. Wir werden auf diese Operationen in

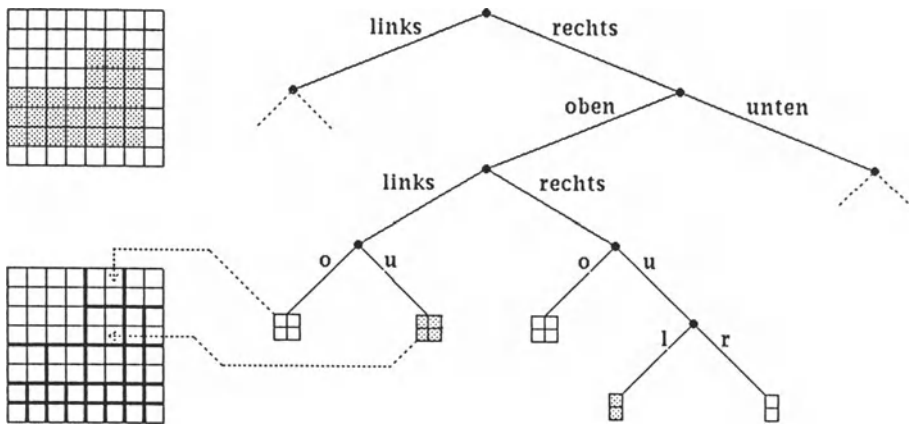


Abb. 2.25. Baumstruktur zur Modellierung einer flächigen Vorlage

Abschnitt 6.4 näher eingehen. Das Skelett kann seinerseits wieder durch eine Baumstruktur bzw. einen verallgemeinerten chain code dargestellt werden.

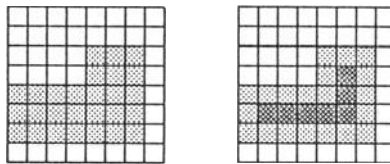


Abb. 2.26. Skelett einer Fläche im Rastermodell

2.5. Hybride Modelle

Wir haben in den einleitenden Abschnitten dieses Kapitels Vektor- und Rastermodelle einander gegenübergestellt und die Vor- und Nachteile abgewogen. Wir haben festgestellt, daß die Stärke von Rastermodellen in ihrem flächenhaften Aspekt und in der Einfachheit ihrer Geometrie liegt. Vektormodelle hingegen zeichnen sich durch die hohe Genauigkeit aus, die sie ermöglichen, sowie durch eine beliebig steigerungsfähige Komplexität in den Beziehungen der einzelnen Elemente. Bei den Operationen, die wir auf diese Modelle anwenden, gibt es auch wieder solche, die besser mit dem Rastermodell harmonieren (wie etwa Flächenverscheidungen) und andere, die eher dem Vektormodell angepaßt werden können (wie

beispielsweise Koordinatentransformationen). In Kapitel 6 gehen wir näher auf solche Operationen ein.

Die Vorteile beider Modelle können in einem *hybriden Modell* kombiniert werden. Ein solches Modell verfolgt im Groben eine Rasterstrategie, im Detail hingegen hat es Vektoraspekte. Teilen wir etwa die Topologie eines Interessensgebietes in gleich große Rasterquadrate, und schneiden wir alle Elemente entlang der Rastermaschen auf, so entsteht ein solches hybrides Modell (Abb. 2.27).

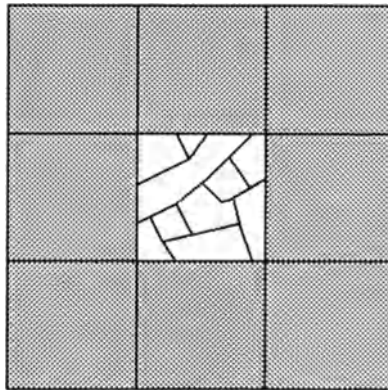


Abb. 2.27. Beispiel für ein hybrides Raster-Vektor-Modell

Dieses Modell hat den Vorteil, daß wir *lokal* arbeiten können, d.h. daß sich alle Operationen in jeweils einer Rastermasche abwickeln lassen. Dies ermöglicht eine effiziente Ausnutzung der Rechenkapazität und reduziert die Anzahl der Massenspeicherzugriffe. Auch der *raumbezogene Zugriff* wird begünstigt. Schwierigkeiten ergeben sich hingegen an den Rändern der Maschen: wenn wir die Elemente in die Nachbarmaschen fortsetzen, wenn wir Übergangsbedingungen berücksichtigen müssen, wenn ein Randausgleich notwendig wird, usw. (wir bezeichnen die damit zusammenhängenden Fragen als **Probleme des Blattschnittes**). Wir müssen daher diese grobe Einteilung in gleichförmige Maschen entsprechend modifizieren. In Kapitel 5 befassen wir uns näher mit diesen Fragen.

Hier erwähnen wir noch, daß der Gedanke eines hybriden Modells auch implizit in vielen Algorithmen vorhanden ist. So berücksichtigen beispielsweise viele Schnittalgorithmen in einer ersten Stufe nur die umschreibenden Rechtecke von Figuren. Nur jene Teile, die im gemeinsamen Durchschnitt der umschreibenden Rechtecke liegen, müssen weiter behandelt werden. Für diese kann man dann die (genaueren) Schnittberechnungen (Schnitt zweier Geraden, Schnitt Gerade

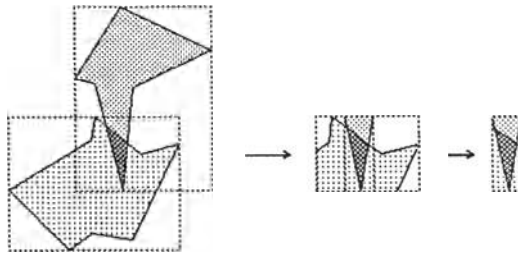


Abb. 2.28. Beispiel für eine hybride Strategie. Schnitt zweier Flächen als Iterationsprozeß

- Kreisbogen usw.) durchführen; man kann aber auch die im gemeinsamen Durchschnitt verbleibenden Reste der beiden Figuren nochmals mit umschreibenden Rechtecken versehen, diese zum Schnitt bringen, und in einem iterativen Prozeß das Problem auf sehr wenige Schnittaufgaben reduzieren (Abb. 2.28). Dies entspricht genau unserem hybriden Modellgedanken. Eine ähnliche Vorgangsweise findet man bei der Darstellung dreidimensionaler Objekte, deren gegenseitige Überdeckung berücksichtigt werden soll (Problematik der *hidden lines* und *hidden surfaces*). Schneiden sich nämlich die umschreibenden Rechtecke der in die Ebene projizierten Objekte nicht, so können sich auch die Objekte nicht überdecken.

2.6. Fraktale Modelle

Vektor- und Rastermodelle waren der Gegenstand unserer Betrachtungen in den vorangegangenen Abschnitten. In beiden Fällen geht man von einer kleinsten erreichbaren Auflösung aus: der Punktgenauigkeit in Vektormodellen und der Maschengröße in Rastermodellen. Verkleinern wir den Bildausschnitt so stark, daß wir in die Nähe dieser "kritischen" Schranke kommen, so verlieren viele geometrisch-topologische Aspekte ihre Aussagekraft; es werden beispielsweise Teile von Schichtenlinien zu Geraden; Der Abstand zwischen einzelnen Schichtenlinien wird immer größer; die Interpolation zwischen benachbarten Schichtenlinien liefert unrealistische Werte; im ungünstigsten Fall terminiert das Berechnungsprogramm.

Nun ist es natürlich so, daß das zugrunde liegende Gelände (auch in diesem extremen Maßstabsbereich) nicht entartet; - vielmehr ist es unser Modell, das versagt. Denn auch in diesem kleinen Bereich gibt es Steigungen, Falllinien, Isolinien; Ein Stück eines Geländes, durch die Lupe betrachtet, weist ähnliche Charakteristika auf wie das

globale Bild. Ähnliches läßt sich auch beispielsweise für die Küstenlinie einer Insel sagen. Große Halbinseln entsprechen auf lokaler Ebene kleinen Landzungen, und dieses Gedankenspiel läßt sich fortsetzen bis zum Sandkorn, das weiter ins Wasser ragt als ein anderes.

Ein Modell, welches seine (globalen) topologischen Eigenschaften in einem lokalen kleineren Bereich **reproduziert**, ist also durchaus realitätsnah. Wir müssen aber nicht immer vom Großen ins Kleine gehen, um auf diese Forderung nach einer hierarchisch sich fortpflanzenden **Selbst-Ähnlichkeit** der Topologie zu stoßen. Für viele Naturphänomene (von meteorologischen Verhältnissen bis zum Erdschwerefeld) können wir nur äußerst grobe Modelle bilden, weil die Anzahl der Parameter, die einen Einfluß ausüben, unüberschaubar groß ist. Eine Verfeinerung eines solchen Modells kann nur mit einer Vergrößerung der Rechner- und Speicherkapazität einhergehen, und dies kann nur in kleinen Schritten geschehen. Andererseits bemerken wir auch in diesen komplexen Phänomenen die Eigenschaft der Selbst-Ähnlichkeit. Globale Luftströmungen spiegeln sich in kleinen Wetterfronten wider, und auch hier läßt sich die Verfeinerung unbeschränkt vorantreiben. Wir wissen auch, daß die derzeit verwendeten Modelle äußerst labil sind, daß also kleine Einflüsse sehr wohl große Wirkungen haben können, und wir daher nicht alles, was klein ist, als "Störung" abtun können.

Es liegt also der Gedanke nahe, daß man, anstatt das Modell durch zusätzliche Parameter zu verfeinern, ein Modell einer gänzlich anderen Bauart verwendet, das auf der Reproduzierbarkeit im Kleinen, also auf der Selbst-Ähnlichkeit beruht. Ein solches Modell beschreibt die **fraktale Geometrie**, die wir im folgenden kurz erläutern wollen. Eine ausführliche Darstellung des gesamten Spektrums der fraktalen Geometrie kann dem Buch von Mandelbrot (1983) entnommen werden. Abgesehen von den darin enthaltenen und für uns verwertbaren Erkenntnissen über Geometrie und Topologie bietet dieses Buch einen höchst interessant und intellektuell unterhaltend geschriebenen Einblick in ein Weltbild, das uns - vorerst - ungewohnt erscheint.

Die in diesem Abschnitt vorgestellten fraktalen Modelle werden derzeit eher im Bereich der graphischen Datenverarbeitung angewendet, und zwar dort, wo es um die Erzeugung realistischer und doch einfach zu modellierender Geländedarstellungen geht (Animationsgraphik, Flugsimulationen etc.). Sie können aber auch im Bereich raumbezogener Informationssysteme wertvolle Dienste leisten; dies wollen wir im folgenden Abschnitt untermauern.

Zur Erklärung der fraktalen Strategie wird sehr häufig die Schneeflocke (Abb. 2.29) herangezogen. Wir wissen, daß sie jene eben geforderte Eigenschaft der Selbst-Ähnlichkeit besitzt. Jeder noch so kleine Teil von ihr ist dem Gesamtbild ähnlich. Wir können ihre Konstruktion als einen rekursiven Prozeß betrachten, der von einem gleichseitigen Dreieck ausgeht, dann von jeder Seite das mittlere Drittel herauschneidet und durch eine Zacke ersetzt; dadurch entstehen neue - kürzere - Seiten, die in einem zweiten Schritt ganz genau so behandelt werden. Der Prozeß kann beliebig fortgesetzt werden. Wir haben hier einen Anfangszustand, den wir **Initiator** nennen. Auf diesen Anfangszustand wenden wir einen **Generator** an. Das Resultat wird neuerlich dem Generator unterworfen usw.

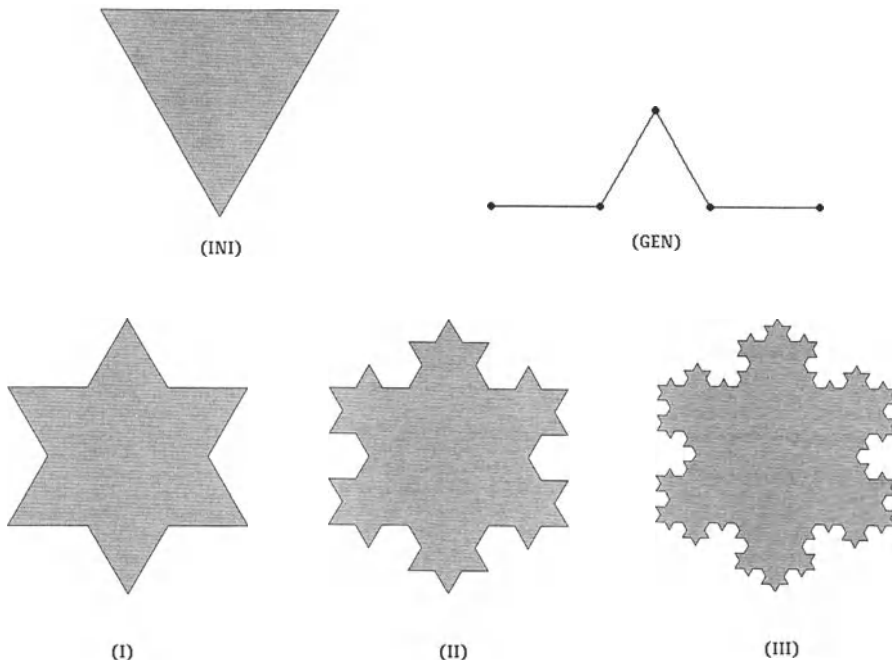


Abb. 2.29. Initiator (INI), Generator (GEN) und einzelne Zwischenstufen (I-III) einer fraktalen Kurve (Schneeflocke)

Da wir diese Rekursionsvorschrift theoretisch unbegrenzt anwenden können, ergeben sich Polygone mit immer mehr Zwischenpunkten. Der Grenzfall ist ein "Polygon" mit unendlich vielen Punkten, ein sogenanntes **Teragon**. Wenn wir den Initiator oder den Generator (oder beide) abändern, so ergibt sich ein anderes Teragon (Abb. 2.30).

Jedes Teragon hat die Eigenschaft, daß ein beliebig kleiner Teil dem Gesamtbild ähnelt. Wir haben also eine höchst

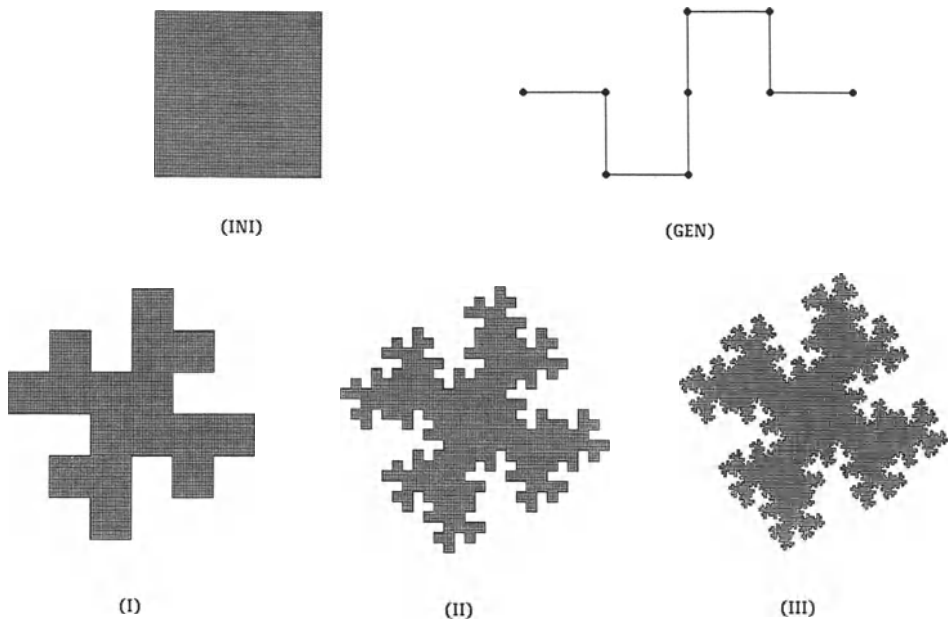


Abb. 2.30. Initiator (INI), Generator (GEN) und einzelne Zwischenstufen (I-III) eines weiteren Teragones

einfache Methode gefunden, um solche linienförmige Strukturen zu bilden, die diese gewünschte Eigenschaft haben.

Es ist einleuchtend, daß das Teragon aus Abb. 2.29 eine "einfachere" Figur darstellt als jenes aus Abb. 2.30. Ein Maß für den Grad der "Einfachheit" eines Teragones ist die sogenannte **fraktale Dimension**. Diese wird durch folgende Beziehung berechnet:

$$D = \log N / \log b \quad (2-6-1)$$

wobei D die errechnete fraktale Dimension ist, N den Vermehrungsfaktor der Seiten bei jedem Rekursionsschritt angibt, und b das Verhältnis zwischen einer alten und einer neuen Seitenlänge ist. Der Logarithmus in dieser Formel ist zur Basis 2 zu nehmen. Für das Beispiel aus Abb. 2.29 ist $N = 4$, $b = 3$, und somit $D = \log 4 / \log 3 = 1.26$, während wir im Fall von Abb. 2.30 $N = 8$ und $b = 4$ rechnen, so daß D den Wert 1.5 erhält. Wir sehen also, daß die fraktale Dimension im zweiten Fall höher ist. Als Spezialfall eines Teragones ergibt sich die gerade Linie selbst; sie hat den Wert $D = 1$. In diesem Fall stimmt die fraktale Dimension mit der uns vertrauten topologischen Dimension überein.

Ein weiterer Extremfall ist die Peano-Kurve (Abb. 2.31). Sie hat den Wert $D = \log 9 / \log 3 = 2$. Die fraktale Dimension ist doppelt so hoch wie die topologische Dimension.

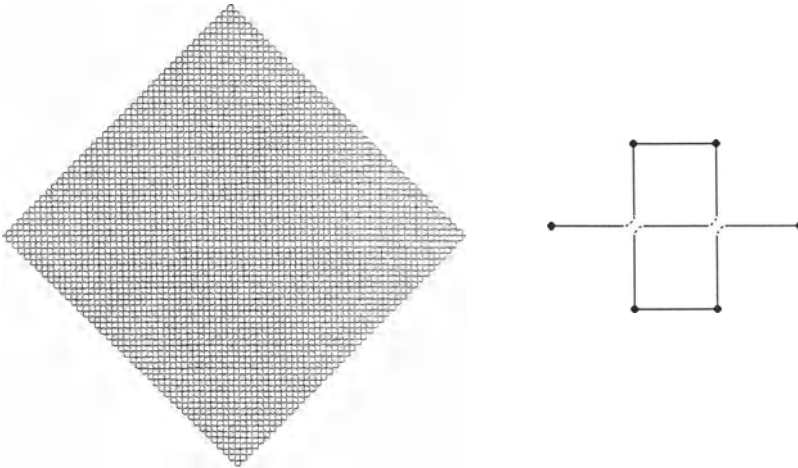


Abb. 2.31. Peano-Kurve und ihr Generator

In diesem Zusammenhang ist die Anmerkung interessant, daß man mit einer sukzessiv verfeinerten Peano-Kurve die gesamte Ebene überdecken kann. Wir haben es hier mit einer Kurve zu tun, deren (topologische) Dimension 1 ist, während sie andererseits auch Aspekte einer Fläche hat, was in dem Wert 2 ihrer fraktalen Dimension zum Ausdruck kommt. Sie ist - zum Unterschied zu den anderen eben gebrachten Beispielen - selbst-überschneidend, so daß sie für unsere Zwecke nicht brauchbar ist (obwohl sie als theoretischer Extremfall interessant ist). Versucht man, ein Teragon mit $D > 2$ zu erzeugen, so ist dies ebenfalls immer selbst-schneidend.

Die für uns relevanten Teragone nehmen also Werte für D an, die zwischen 1 und 2 liegen. Je näher D bei 2 liegt, desto komplexer wird das Teragon, desto mehr ähnelt es der flächendeckenden Peano-Kurve, und immer größer wird der fraktale Anteil. Wenn wir an die Karte Europas denken, so wird die fraktale Dimension der Küste Schottlands oder Norwegens sicher näher bei 2 liegen als jene der Küste Belgiens.

Es ist klar, daß die Länge eines Polygons von seinem Komplexitätsgrad abhängt. So hat etwa Richardson (1961) empirisch die Abhängigkeit der Gesamtlänge von Staatsgrenzen von der jeweils verwendeten kleinsten Polygonseite untersucht. Für die Länge der Grenze zwischen den Niederlanden und Belgien findet er in verschiedenen Quellen

auch zwei verschiedene Werte 380 km und 449 km; für die Grenze zwischen Spanien und Portugal variieren die angegebenen Werte von 987 km bis 1214 km. Richardson findet empirisch folgenden Zusammenhang: Wenn ds die Länge einer elementaren Polygonseite ist, so hängt die Anzahl Z der nötigen Polygonseiten von ds in folgender Weise ab:

$$Z = F * ds^{-D} \quad (2-6-2)$$

F und D sind Konstante. D ist genau unsere oben erwähnte fraktale Dimension. Je komplizierter also das Teragon ist, desto größer wird Z bei gegebenem ds sein. Natürlich ist dann die Gesamtlänge L des Polygons gegeben durch

$$L = ds * Z = F * ds^{1-D} \quad (2-6-3)$$

Betrachten wir nun nochmals die Diskrepanzen zwischen den angegebenen Längen der Staatsgrenzen, die sich im Bereich von 20% bewegen, so ergibt sich, daß diese Diskrepanz dann auftritt, wenn man im einen Fall die Elementarseite ds halb so groß wählt als im anderen Fall - eine durchaus realistische Annahme.

Wir sehen also, daß das Modell der Fraktale in diesem Fall sehr wohl die Realität besser widerspiegeln kann als irgendeine andere Kurve bzw. ein Polygon. Dasselbe gilt für Küstenlinien, Begrenzungen von Waldflächen, Schichtenlinien usw. Mandelbrot (1983) treibt das Gedankenexperiment übrigens noch weiter: wird ds immer kleiner, so wächst die Gesamtlänge immer mehr. Er kommt zu dem Schluß, daß die Länge der Küstenlinie Großbritanniens über alle Maßen wächst; dies, obwohl sie eine endliche Fläche einhüllt.

Schließlich wollen wir noch auf die Bedeutung von F in obiger Formel eingehen. F gibt ein Maß für das Teragon an, ähnlich wie die Länge einer Linie oder der Inhalt einer Fläche ein Maß ist. Berechnen wir etwa die Länge einer Linie (z.B. durch numerische Integration), so zerlegen wir sie in Z Inkremente und multiplizieren Z mit der (zur ersten Potenz erhobenen) Elementarstrecke.

$$L = Z * ds^1 \quad (2-6-4)$$

Der Inhalt einer Fläche wird gleichermaßen so berechnet, daß wir Z elementare Flächenstücke bilden (z.B. Quadrate); die Zahl Z wird dann mit dem zur zweiten Potenz erhobenen Elementarinkrement multipliziert.

$$F = Z * ds^2 \quad (2-6-5)$$

Machen wir dies auch mit unserem Teragon, so gibt D die Potenz an, zu der wir unser Elementarinkrement ds erheben müssen:

$$F = Z * ds^D \quad (2-6-6)$$

Setzen wir hier für Z aus der obigen Formel ein, so ergibt sich tatsächlich

$$F = F * ds^{-D} * ds^D \quad (2-6-7)$$

Somit ist F von der Länge eines elementaren Inkrementes ds unabhängig. Wir können den Grenzübergang für ds gegen Null machen und erhalten in F ein (endliches) Maß für das Teragon.

Um das fraktale Modell in der Praxis einsetzen zu können, bedarf es allerdings einiger Modifikationen. Wir können nicht erwarten, daß sich ein Grenzverlauf oder eine Küstenlinie im Kleinen tatsächlich in einer solch regelmäßigen Anordnung wie die Umrandung der Schneeflocke reproduziert. Vielmehr ist die Selbst-Ähnlichkeit nur eine *durchschnittliche*, das heißt, daß wir es hier mit einer stochastischen Größe zu tun haben. Die rekursive Generierung des Teragones kann als *stochastischer Prozeß* aufgefaßt werden. Mit fortschreitender Zeit ergeben sich jeweils neue Verfeinerungen des Ausgangspolygones, die sich "im Mittel" einem Teragon nähern, während die Ausprägung im Detail dem Zufall überlassen bleibt. Eine umfassende Einführung in die Theorie der stochastischen Prozesse bietet Papoulis (1965).

Betrachten wir beispielsweise den Generator der Schneeflocke. In jedem Rekursionsschritt können wir ihn entweder so anwenden, wie wir ihn weiter oben beschrieben haben, oder mit jeweils umgekehrten Vorzeichen (statt "Buchten" entstehen "Halbinseln"). Überlassen wir das Vorzeichen im k-ten Schritt - also zum Zeitpunkt t (k) - dem Zufall, so haben wir einen stochastischen Prozeß vor uns. In ähnlicher Weise können wir auch die Längen bzw. Richtungswinkel der Teilstrecken des Generators stören. Die *Brown'sche Molekularbewegung* ist ein klassisches Beispiel für einen stochastischen Prozeß. Sie wird aber auch (im stochastischen Sinn) durch eine fraktale Kurve der Dimension $D = 2$ beschrieben; ihre Dimension ist jener der Peano-Kurve äquivalent; dies bedeutet, daß ein Elementarteilchen mit fortschreitender Zeit und "im Mittel" jeden Punkt der Fläche überstreicht.

Für eine realitätsnahe Anwendung des fraktalen Konzeptes ist es oft auch notwendig, aus der Kette der Rekursionsschritte erst den k-ten Schritt und die nachfolgenden n Schritte herauszulösen, d.h. wir unterdrücken sowohl die (makroskopischen) Anfangszustände wie auch die (mikroskopischen) Folgezustände des Prozesses. Es ist auch denkbar, daß wir von einem koordinatenmäßig gegebenen (deterministischen) Zustand ausgehen; diesen verwenden wir dann als Initiator eines (stochastischen) fraktalen Prozesses.

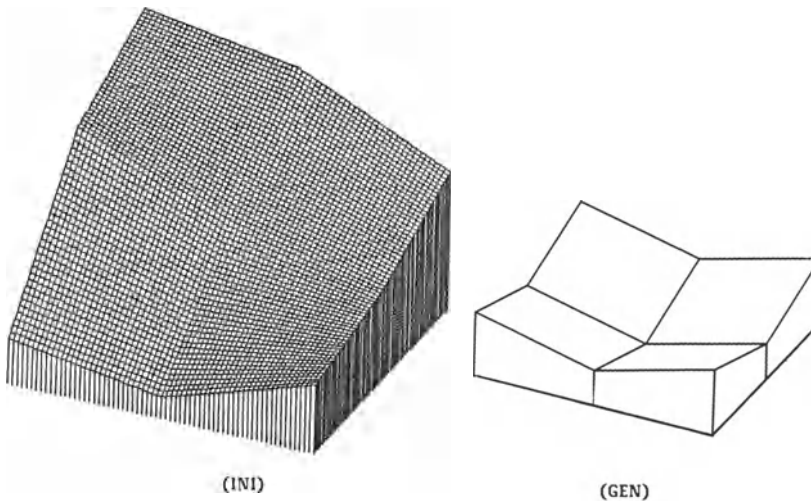


Abb. 2.32. Initiator (INI) und Generator (GEN) eines fraktalen Höhenmodells

Schließlich wollen wir auch noch erwähnen, daß die fraktale Geometrie nicht nur auf Kurven anwendbar ist. Vielmehr lassen alle in diesem Kapitel besprochenen topologischen Modelle eine fraktale Variante zu. Ein konventionelles (grobmaschiges) Höhenmodell (siehe Abschnitt 6.6) kann die Rolle des Initiators für ein fraktales Höhenmodell übernehmen. Der Generator ersetzt dabei eine Masche durch ein Geflecht von kleineren Maschen, wobei das Verhältnis des Flächeninhaltes der ursprünglichen Masche zum Flächeninhalt des Geflechtes für die Berechnung der fraktalen Dimension herangezogen wird (Abb. 2.32). Diese Dimension liegt - je nach dem Ausmaß der "Störung" - zwischen 2 und 3 (Abb. 2.33). Die Dimension eines Modells aus dem Bereich der Kalkalpen wird näher bei 3 liegen als jenes der Kärntner Nockberge.

Aber auch Flächen lassen sich durch fraktale Methoden erzeugen: in Abb. 2.34 sind zwei Flächen unterschiedlicher fraktaler Dimension dargestellt. Die Inseln der Ägäis oder die Seen der finnischen Seenplatte ergeben eine höhere fraktale Dimension als die Seen im österreichisch - bayerischen Alpenvorland.

Zusammenfassend können wir sagen, daß die Anwendungen des fraktalen Modells im Bereich raumbezogener Informationssysteme überall dort liegen werden, wo die Topologie stärker als die Geometrie gewichtet wird. Sie leistet also gute Dienste für eine realistische Modellierung der Verteilung der Höhen in einem Gebiet, wo es weniger auf eine

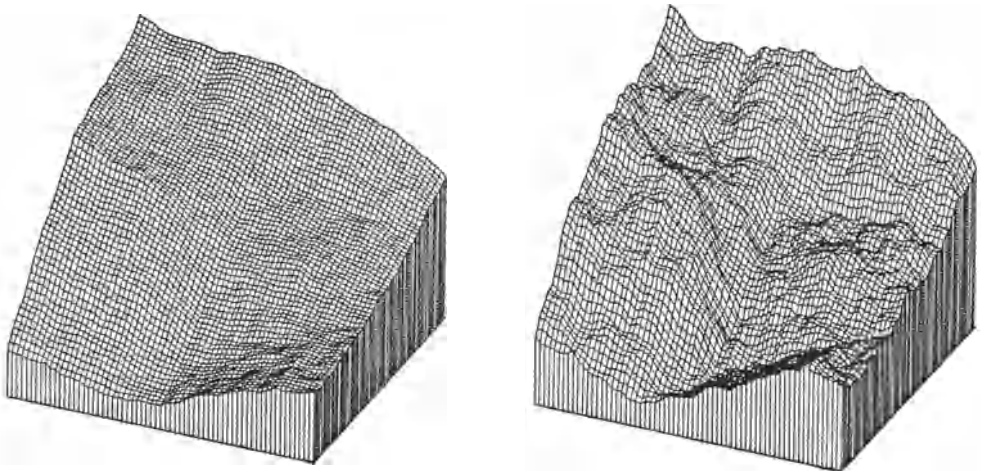


Abb. 2.33. Fraktale Höhenmodelle, die durch Anwendung unterschiedlich starker Generatoren auf den Initiator aus Abb. 2.32 entstehen

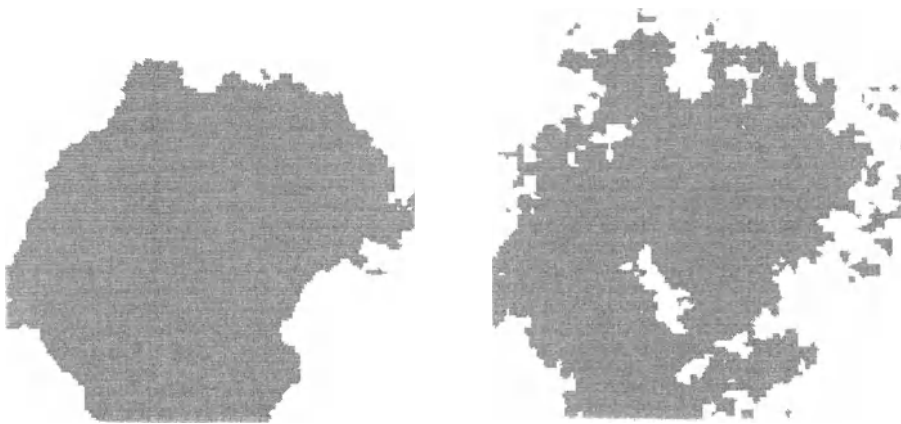


Abb. 2.34. Fraktale Flächenmodelle

koordinatenmäßig präzise Angabe ankommt (dies ist Geometrie) als vielmehr auf eine Veranschaulichung der Schroffheit oder Lieblichkeit eines Geländes (dies wird durch die Topologie der Schichtenlinien zum Ausdruck gebracht). Ähnliches läßt sich zum Beispiel von Küstenlinien sagen: eine realistische Darstellung der norwegischen Küste wird Teile der Geometrie opfern und Fjorde unter Umständen drastischer betonen als dies von den Koordinaten her notwendig ist.

Ein wichtiges Anwendungsgebiet fraktaler Modelle sind aber auch **Simulationsstudien**. Es geht dabei um die Frage "was wäre, wenn?". Modelle für die Luftströmungen und die Schadstoffkonzentration in einem Ballungsraum, den Wasserhaushalt in einer Kleinregion, die Hochwassergefährdung in einem Gebirgstal sind typische Beispiele. Ihre Modellierung mit herkömmlichen Mitteln erfordert eine Unzahl von Parametern. Fraktale Modelle hingegen können auf sehr viel einfachere Weise realistische Datengrundlagen bieten und erlauben es, sowohl Aussagen globaler Natur als auch Fallstudien im Detail durchzuführen.



KAPITEL 3
THEMATIK

3.1. Begriffsbestimmungen und Gliederungen

Die Daten raumbezogener Informationssysteme stützen sich auf zwei Säulen: eine dieser beiden Säulen wird von der *Geometrie* und *Topologie* der Daten gebildet; die andere Säule ist die *Thematik*. Während die Geometrie die örtliche Zuordnung gewährleistet, und die Topologie das räumliche Zueinander der Informationen widerspiegelt, gibt die Thematik Aufschluß über die *inhaltliche Information*. Bei allen Informationen, die wir in solch einem System speichern wollen, sind beide Aspekte - wenn auch mehr oder weniger stark ausgeprägt - vertreten. Die Thematik bildet also einen Kontrast zur Geometrie und Topologie - ebenso gut aber können wir sagen, daß die Aspekte einander ergänzen und vervollkommen, so wie die Form und der Bildinhalt von Puzzleteilen zu einem harmonischen Ganzen zusammenwachsen.

Wenn wir als Beispiel einen Punkt eines Katasterplanes betrachten, so hat er Koordinaten, er ist in Linien eingebunden, er liegt auf der Begrenzung gewisser Flächen, er hat einen bestimmten Abstand zu anderen Punkten / Linien / Flächen. All dies sind seine geometrisch-topologischen Aspekte. Andererseits wissen wir etwa, daß ein Parzellengrenzpunkt ist, und daß er durch einen Stein vermarktet wurde; dies sehen wir als die Thematik dieses Punktes an. Ähnliches läßt sich von den Linien auf einer topographischen Übersichtskarte sagen. Sie werden aus Stützpunkten generiert, sie schneiden einander, sie haben jeweils eine bestimmte Länge; und andererseits tragen sie die Bedeutung "Landesstraße", "Gemeindestraße", "Fahrweg", "Saumpfad" usw.

Nicht immer ist die Unterscheidung so leicht. Manchmal verschwimmen die Grenzen; die Punkthöhe etwa kann durchaus auch als thematischer Aspekt angesehen werden. Es ist dementsprechend schwierig, eine allgemein akzeptierte Umschreibung des Ausdruckes "thematische Kartographie" zu finden. In Witt (1970) finden wir eine Zusammenfassung der wichtigsten Begriffsbildungen und Gliederungen im Rahmen der thematischen Kartographie. Dieses Buch bietet auch eine erstklassige Einführung in dieses Gebiet, sowie eine erschöpfende Literaturliste. Obzwar nur stellenweise ein "Vorausahnen" der Möglichkeiten anklingt, die wir heute aufgrund der digitalen Methoden in der Kartographie haben, ist es doch eine solide Grundlage für Überlegungen hinsichtlich der Ausprägung der thematischen Komponente in einem raumbezogenen Informationssystem: denn die Zielsetzungen der thematischen Kartographie sind die gleichen geblieben, nur die Mittel haben sich geändert.

Wenn wir den Begriff der *Karte* näher untersuchen, so gilt diese gemeinhin als ein *orientiertes, verkleinertes, verebnetes Grundrißbild eines Teiles der Erdoberfläche, das*

die Gesamtheit der für diesen Teil bedeutungsvollen Erscheinungen oder eine beschränkte Auswahl daraus wiedergibt. Der Begriff "Grundriß" paßt in unsere Definition der Geometrie und Topologie, während es uns der zweite Teil der Definition überläßt, eine Auswahl der jeweils relevanten "Bedeutung" bzw. Thematik zu treffen.

Andere Autoren wiederum teilen die Karten in solche ein, in denen die Erscheinungen des Geländes Hauptgegenstand sind, und solche, in denen die Erscheinungen und Sachverhalte zur Erkenntnis ihrer selbst dargestellt sind, während der Kartengrund nur der Angabe ihrer geographischen Lage dient.

Diese Definitionen treffen auf topographische Karten zu. Wir stecken unsere Ziele aber etwas allgemeiner. Verwaltungs- oder Eigentumsgrenzen etwa sind keine "sichtbaren" Geländeformen; sie fassen Teile der Erdoberfläche eher inhaltlich (also thematisch) zu Ländern, Bezirken, Gemeinden, Parzellen zusammen. Auch unterirdisch verlegte Leitungsstränge sind nicht sichtbar. Trotzdem ist ihre geometrisch-topologische Komponente offensichtlich. Sie beziehen sich auf vermessene Punkte, die sehr wohl physisch vorhanden sind. Diese Punkte werden dann aufgrund einer topologischen Vorschrift miteinander verbunden (z.B. durch Gerade).

Aber sogar dieses Kriterium (die physische Existenz von Stützpunkten einer Linie bzw. von Begrenzungspunkten einer Fläche) müssen wir fallenlassen, wenn wir Waldschadenszonen betrachten, infrastrukturelle Gegebenheiten, aber auch Landschaften und Gebirgsregionen. (An welchem Punkt beginnen die Alpen und wo hören sie auf?)

In dieser Situation kommt uns die Einteilung zu Hilfe, die Schmidt-Falkenberg (1964) für die Erscheinungsformen einer Karte getroffen hat; es sind dies

- **geodätische Karten**, die vorwiegend Beobachtungs- und Forschungsergebnisse der Geodäsie enthalten (Topographie, Katasteraufnahme, Leitungskataster usw.),
- **geographische Karten**, die vorwiegend Beobachtungs- und Forschungsergebnisse der Geographie enthalten (z.B. Geomorphologie),
- **astronomische Karten**
- und **thematische Karten**, die auf der Grundlage von geodätischen, geographischen und astronomischen Karten entworfen werden und einen oder mehrere Darstellungsgegenstände im Vergleich zu den anderen vorrangig behandeln oder graphisch besonders hervorheben.

Diese Einteilung kommt unserer Sicht der Dinge schon näher.

Es gibt also einen "gemeinsamen Nenner" aller dieser Karten; der geometrisch-topologische Hintergrund, der hauptsächlich zur **Orientierung** des Kartenbenützers dient. Dazu gibt es dann im allgemeinen verschiedene thematische Ausprägungen, die der **Information** des Kartenbenützers über bestimmte inhaltliche Zusammenhänge dienen. Es wird also bewußt jeweils ein Thema (oder eine Auswahl weniger Themen) in den Vordergrund gestellt, z.B. das Klima, die Bevölkerungsstruktur, die Bodennutzung, der Verkehr insgesamt, bzw. nur der Straßenverkehr, oder davon nur der Teil, der auf Privatfahrzeuge entfällt usw. (Abb. 3.1).

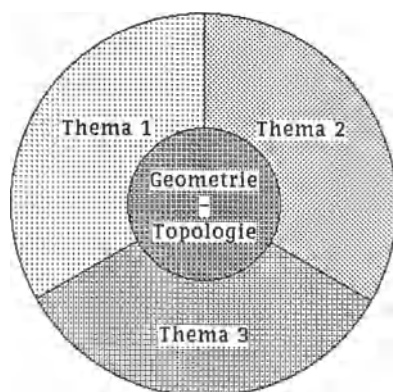


Abb. 3.1. Geometrie und Topologie in Verbindung mit verschiedenen Themen

Andererseits ist die Grenze zwischen den Bereichen nicht immer klar ersichtlich. Betrachten wir als Beispiel die Grundstücke einer Gemeinde. Wenn es uns genügt, für jedes Grundstück nur die Art seiner Nutzung festzuhalten, so können wir diese (an und für sich thematische) Information beispielsweise durch Hinzufügen einer weiteren Spalte zur Tabelle der Grundstücke berücksichtigen. Der dort abgelegte Zahlenwert spielt in diesem Fall eine ähnliche Rolle wie der (topologische) Formparameter. Er kann dann beim Zeichnen des Planes beispielsweise in eine bestimmte Farbe oder Schraffurart umgesetzt werden.

Wenn wir aber zusätzlich noch die Bebauungsart, die Eigentumsverhältnisse, die Bodenbeschaffenheit, den Verkehrswert, die Schuldenlasten oder die Aufschließung betrachten, so sind dies eindeutig thematische Komponenten, für die wir gesonderte Strategien bereitstellen müssen; es genügt nicht mehr, im topologischen Bereich eine thematische "Nische" bereitzustellen, sondern die Thematik beansprucht die ihr zustehende Sonderbehandlung, während die Grundstücksflächen den geometrisch-topologischen Kern bilden.

Andererseits könnten wir aber auch behaupten, daß der Umstand, daß wir Flächen bilden, die sich selbst nicht überschneiden dürfen, die auch ihre Nachbarflächen nicht schneiden, und die keine undefinierten "Löcher" zulassen, einer thematischen Begründung bedarf (denn in anderen thematischen Bereichen wären Überschneidungen sehr wohl zulässig, so etwa Schadstoff-Emissionszonen). In dieser strikten Sicht wären die Geometrie und Topologie auf eine Ansammlung von Punkten und Kanten reduziert, und erst die Thematik bringt Ordnung in diese Ansammlung.

Extrembeispiele sind natürlich dazu da, um die "goldene Mitte" herauszustreichen. Es ist sicher sinnvoll, bestimmte "einfache" Teile der Thematik in den zentralen Block zu ziehen. Dies betrifft beispielsweise jenen Teil der Grundstücksthematik, der für die flächige und nicht-überschneidende Aufteilung zuständig ist (Abb. 3.2). Für diese Vorgangsweise spricht der Umstand, daß die Topologie gar nicht gänzlich unabhängig von der jeweils gewählten Thematik (den jeweils gewählten Themen) sein kann. Viele topologische Fragestellungen sind nur im Verbund mit diesen Themen sinnvoll. Auch wenn wir oben gefordert haben, daß die Geometrie und Topologie den Hintergrund abgibt, vor dem sich der Kartenbenutzer orientiert, so ist es dennoch klar, daß jede Art der Darstellung schon eine bestimmte Thematik suggeriert: die Farbe, der Linientyp, die Schraffur usw. sind graphische Umsetzungen der Thematik. (In Kapitel 7 beschäftigen wir uns mit dieser graphischen Umsetzung eines Themas).

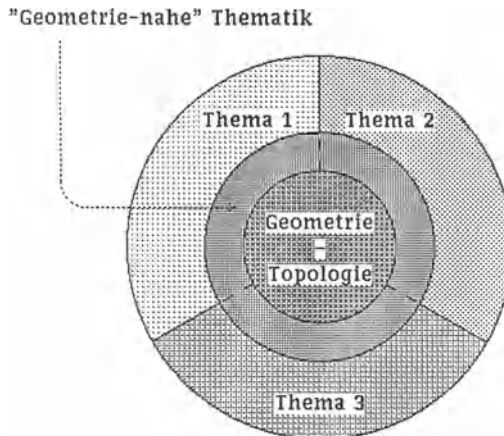


Abb. 3.2. Enge und lose Beziehungen zwischen der Thematik und der Geometrie und Topologie

In einem raumbezogenen Informationssystem sind im allgemeinen mehrere Themen ("thematische Ebenen") gespeichert,

die wahlweise einzeln oder in bestimmten Kombinationen betrachtet werden. Es ist nützlich, wenn wir uns jede thematische Ebene als Folie vorstellen. Die für eine Auswertung jeweils relevanten Folien können dem Gesamtstapel entnommen und übereinandergelegt werden (Abb. 3.3). Natürlich können zu einem späteren Zeitpunkt neue Folien auf den Stapel gelegt werden (es wird eine neue Thematik aufgenommen). Es kann aber auch sinnvoll sein, zwei Folien, die oft gemeinsam auftreten, zusammenzulegen (**Verschneidung zweier thematischer Ebenen**): so entsteht beispielsweise aus einer Folie "Grundstücke" und einer Folie "Bebauung" eine dritte Folie "bebaute und unbebaute Grundstücksanteile".

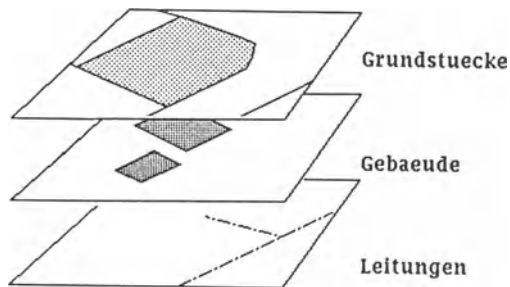


Abb. 3.3. Auffächerung der Thematik in Ebenen

Wo hat nun die Topologie ihren Platz in diesem Folienkonzept? Sie ist natürlich implizit und im allgemeinen bruchstückhaft in jeder einzelnen Folie vorhanden. Verschneiden wir alle thematischen Ebenen, die in unserem Informationssystem auftreten, so ergibt sich als Endprodukt genau die Topologie. Dies lässt sich gut am obigen Beispiel der bebauten und unbebauten Grundstücksanteile veranschaulichen. Diese Teilflächen und Teillinien sind dann der topologische "Untergrund". Je mehr thematische Ebenen vorhanden sind, desto mehr wird die Topologie zerstückelt (Abb. 3.4).

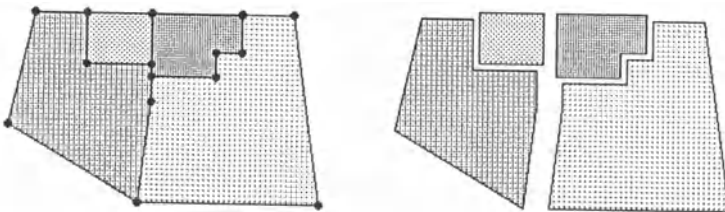


Abb. 3.4. Topologie als Endprodukt von Verschneidungen

Wir gehen nun näher auf die verschiedenen Variationen ein, die es für thematische Karten gibt. So können wir grundsätzlich zwischen **qualitativen Karten** und **quantitativen Karten** unterscheiden. Während qualitative Karten mehr die Art des Themas in den Vordergrund stellen, geht es bei quantitativen Karten um Größen, Mengen und Werte. Ein Beispiel für eine qualitative Karte ist ein Straßenatlas (Abb. 3.5). Rote Doppellinien sind Autobahnen, einfache rote Linien sind Bundesstraßen, gelbe Linien sind Landesstraßen usw.

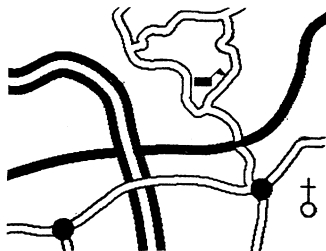


Abb. 3.5. Straßenatlas als Beispiel einer qualitativen Karte

Als Beispiel für eine quantitative Karte sei die Temperaturverteilung in der Wetterprognose des Fernsehens genannt (Abb. 3.6). Diese Temperaturverteilung kann etwa durch numerische Werte angegeben werden, die an bestimmten diskreten Punkten prognostiziert werden und der Karte überlagert werden. Man kann aber auch unterschiedliche Temperaturen durch Balken verschiedener Höhe angeben. Aber auch Linien gleicher Temperatur (oder ähnliche Isolinien) bzw. die dazwischenliegenden Flächen können dargestellt werden. Sowohl absolute als auch relative Angaben können der quantitativen Methode zugrunde liegen.

Im Moment sind uns Darstellungsvarianten weniger wichtig als das **Modell** in seiner Eigenschaft als abstraktes Abbild der Wirklichkeit (bzw. jenes Teiles der Wirklichkeit, der uns interessiert). Dieses Modell hat natürlich eine größere (weil langfristige) Bedeutung als eine bestimmte Art der Darstellung. Es beeinflusst die Speicherungsstrategie und das Spektrum der möglichen Anwendungen. Als Modell einer qualitativen Thematik genügt etwa ein Index, der einem topologischen Element - einem Punkt, einer Linie, einer Fläche - zugeordnet wird: im obigen Beispiel der Straßenkarte würde man etwa Autobahn = 1, Bundesstraße = 2 wählen usw. Diese Indizes 1,2,... würde man den Linien zuordnen. Soll ein Auszug aus den Daten dargestellt werden, so wird eine "Übersetzungstabelle" (**Look-up table**) dazwischengeschaltet, die jedem Index eine entsprechende graphische Erscheinungsform zuteilt.

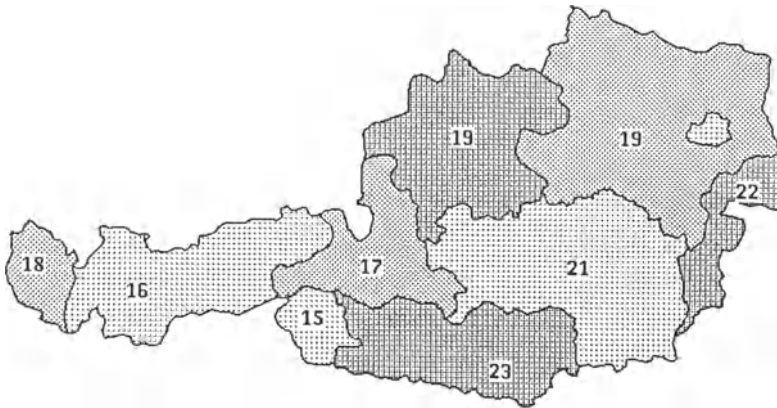


Abb. 3.6. Temperaturverteilung als Beispiel einer quantitativen Karte

Das Modell einer quantitativen Abhandlung der Thematik würde hingegen jeder topologischen Einheit die erforderlichen numerischen Felder zuordnen. Eine graphische Umsetzung ist (in einfachen Fällen, wie im obigen Beispiel der Temperaturverteilung) möglich, ansonsten müssen die Werte numerisch eingeblendet werden.

Die Einteilung in qualitative und quantitative Aspekte zwingt uns, noch einmal auf das Problem der Unterscheidung zwischen Geometrie und Topologie einerseits und Thematik andererseits zurückzukommen: qualitative Aspekte können der Topologie natürlich leichter im Sinne von zusätzlichen, beschreibenden Merkmalen zugeordnet werden als quantitative Aspekte, die eher ein "Eigenleben" führen.

3.2. Die Thematik in Rastermodellen

In Kapitel 2 haben wir uns eingehend mit der Bildung geometrisch-topologischer Modelle beschäftigt und die beiden Hauptvarianten Rastermodell und Vektormodell herausgearbeitet. In diesem Abschnitt wollen wir untersuchen, wie wir auch eine Thematik in das Rastermodell hineinbringen können. Im nächsten Abschnitt werden wir uns mit der Thematik in Vektormodellen auseinandersetzen. (Wir können bereits jetzt vorausschicken, daß es auch hier hybride Formen gibt.)

Im Rastermodell gibt es nur ein geometrisches Grundelement, die Rasterzelle. Alle Rasterzellen sind gleich groß und spannen in einer regelmäßigen zeilen- und spaltenweisen Anordnung die gesamte Ebene (im dreidimensionalen Fall den

gesamten Raum) auf. Fernerkundungsdaten haben grundsätzlich Rastercharakter; wir haben aber bereits darauf hingewiesen, daß man auch Bevölkerungsdaten, Infrastrukturdaten und ähnliches in Rasterform ablegen kann (Abb. 2.4). In diesem Fall haben wir der Topologie (also jeder Rasterzelle) bereits eine Thematik zugeordnet. Wenn wir über jeder Rasterzelle eine Säule errichten, deren Höhe der jeweiligen thematischen Information innerhalb der Zelle entspricht, so wird es augenscheinlich, daß die natürlichste Art, eine Thematik in ein Rastermodell einzubringen, das Hinzufügen einer weiteren **thematischen Dimension** ist (Abb. 3.7). Längs dieser Dimension nimmt die Thematik diskrete oder auch kontinuierliche Werte an. Diskrete Werte etwa, falls es sich um die Anzahl der Industriebetriebe innerhalb der Rasterzelle handelt, kontinuierliche Werte, wenn es um die jährliche Niederschlagsmenge geht.

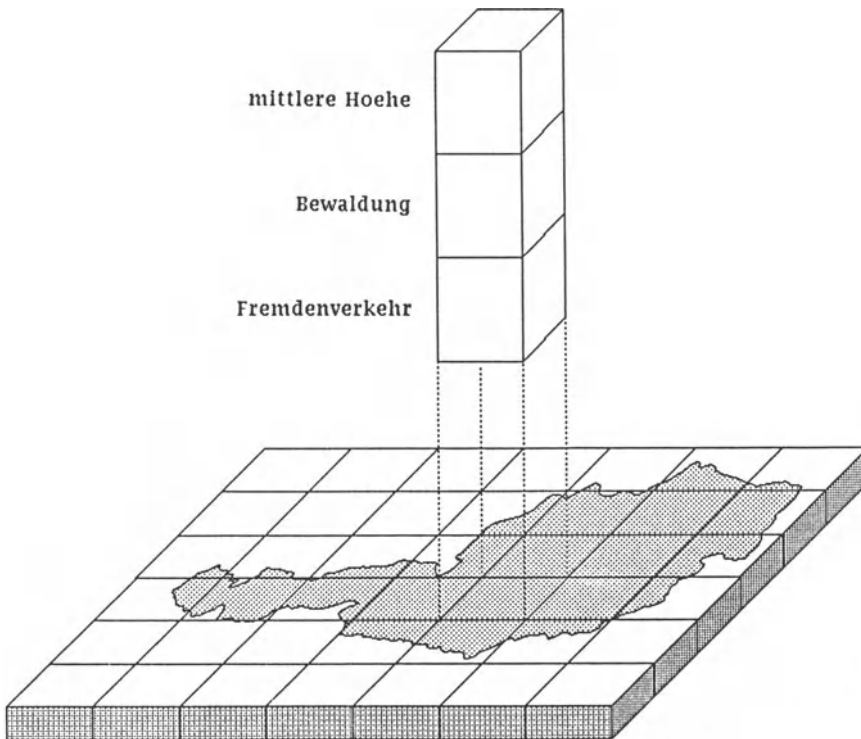


Abb. 3.7. Thematik als zusätzliche Dimension im Rastermodell

Wir bilden also einen **Produktraum**, der sich aus dem topologischen und dem thematischen Raum ergibt. Für den Fall, daß der Topologie genau ein Thema überlagert wird, ist die Dimension des Produktraumes um 1 höher als jene des topologischen (Teil-)Raumes; jedoch können wir dieses

Konzept auch auf mehrere thematische Dimensionen verallgemeinern. Im folgenden beschränken wir uns aber immer auf das Produkt eines zweidimensionalen topologischen Grundraumes mit genau einer thematischen Dimension.

Betrachten wir als Beispiel ein Fernerkundungsbild und als Thema die jeweilige Beschaffenheit des Bodens. Nehmen wir ferner an, daß dieses Thema vier verschiedene Werte annehmen kann:

0 = Ödland bzw. nicht klassifiziert
 1 = Wasser
 2 = Grünland
 3 = Ackerland

Wir können nun jeder Rasterzelle unseres Modells einen der Werte 0, 1, 2, 3 zuordnen. Dieselben Möglichkeiten, wie wir sie im Abschnitt 2.4 für den **topologischen Zusammenhalt** von Rasterelementen kennengelernt haben (chain codes, run length encoding, Skelettbildung, Baumstrukturen) können wir auch für den **thematischen Zusammenhalt** definieren, weil ja im Rastermodell der Unterschied zwischen Topologie und Thematik nicht so deutlich zum Ausdruck gebracht werden kann wie im Vektormodell (es ist schwierig, oft unmöglich, einen thematischen Zusammenhang zwischen Rasterzellen zu definieren, die nicht benachbart sind). Der thematische Wert einer Rasterzelle kann wie der **Grauwert** eines Pixels behandelt werden. Er wird deshalb oft auch als Grauwert bezeichnet, obzwar er im allgemeinen mit der Darstellung nichts zu tun hat; stellt man diese thematischen Werte dann doch dar, so erhält man ein **Falschfarbenbild**. In unserem Beispiel könnten wir den Grauwert dieses Falschfarbenbildes durch zwei Bits repräsentieren.

Wir können aber auch die oben gegebene Erklärung über den Produktraum wörtlich auslegen und tatsächlich statt der Matrix ein dreidimensionales Feld anlegen. In unserem Fall gibt es längs der dritten Dimension vier Ebenen. In jedem Element dieses Feldes kann der Wert 0 oder 1 stehen. Beim Durchschreiten des Feldes in vertikaler (also thematischer) Richtung treffen wir auf genau ein Element mit dem Wert 1, und zwar in der Höhe, die der jeweiligen Bodenbeschaffenheit entspricht. Natürlich können wir auch hier die Einsen und Nullen als Bits eines Grauwertes auffassen, womit wir wieder bei der früheren Anschauungsweise angelangt sind.

Eine interessante Variante ergibt sich, wenn wir mehrere Themen in eine thematische Dimension verpacken. Es ist ja denkbar, daß pro Rasterzelle mehrere thematische Informationen anfallen, wie etwa die Bevölkerungsdichte und die Pendlerbewegungen in einem Rasterquadrat.

Sehen wir etwa für die Bevölkerungsdichte die folgenden Möglichkeiten vor:

- 0 = weniger als 100 Einwohner pro Quadratkilometer
- 1 = 100 - 200 Einwohner
- 2 = 200 - 300 Einwohner
- 3 = mehr als 300 Einwohner

Bezüglich der Pendler treffen wir die folgende Einteilung:

- 0 = weniger als 10%
- 1 = 10% - 20%
- 2 = 20% - 30%
- 3 = mehr als 30%

Dann können wir aus dem kombinierten Grauwert 0111 einer Rasterzelle ablesen, daß die Dichte bei 100 - 200 Einwohnern liegt (die ersten beiden Bits entsprechen der dezimalen Zahl 1), während mehr als 30% der Einwohner pendeln (die letzten beiden Bits entsprechen der Zahl 3). Diese Vorgangsweise wird als **bit slicing** bezeichnet; sie hat ein Pendant in der graphischen Datenverarbeitung, wo man mehrere **bit planes** definiert, die mit einer Lookup-table gekoppelt sind. Jedes Pixel hat im allgemeinen Anteil an mehreren bit planes und wird in der jeweiligen Mischfarbe dargestellt. In unserem Fall geht es (vorerst) nicht um die Darstellung, und die Einwohner- und Pendlerstatistik übernimmt die Rolle der Lookup-table.

Natürlich ergibt das **bit slicing** (wegen des explodierenden Speicherplatzbedarfes) eine sehr beschränkte Anzahl von Variationsmöglichkeiten, was ein typisches Kennzeichen der Rasterstrategie ist, während die Vorteile dieser Strategie (Einfachheit in der Handhabung) ebenso offensichtlich sind.

Als besonderen Vorteil eines Rastermodells haben wir immer den Umstand hervorgehoben, daß raumbezogene Abfragen leicht befriedigt werden können; die Frage, welche Rasterzellen (zumindest teilweise) innerhalb eines bestimmten Bereiches liegen, ist schnell beantwortet: man schneidet ein Rechteck mit dem Rastergitter. Diese Eigenschaft überträgt sich nun in natürlicher Weise auf das um die Thematik erweiterte Modell. Wir müssen nur den gesamten Produktraum auf den topologischen Teilraum projizieren und können dann dort den Schnitt durchführen. Gleichbedeutend damit ist das Schneiden des gesamten Rastergeflechtes im Produktraum mit einer rechteckigen Säule, die über dem Suchrechteck errichtet wird (Abb. 3.8A).

Ganz analog dazu können wir aber auch rein thematische Anfragen behandeln. Wollen wir etwa alle jene Rasterzellen herausfinden, in denen die Bevölkerungsdichte zwischen 100 und 300 Einwohnern pro Quadratkilometer liegt, so schneiden wir das Rastergeflecht wieder mit einem Quader, der aber

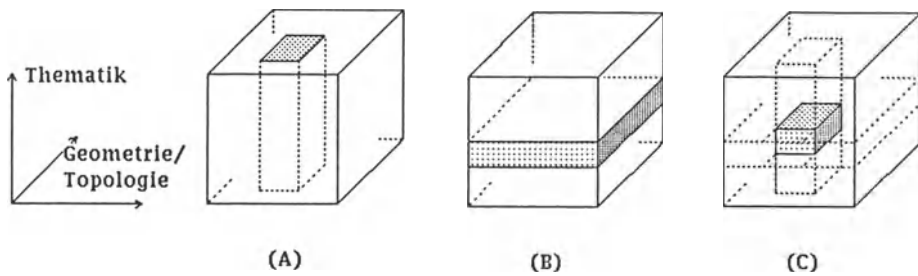


Abb. 3.8. Raumbegrenzte Abfrage (A), thematische Abfrage (B) und kombinierte Abfrage (C) im Produktraum

diesmal horizontal das gesamte Gebiet überdeckt, vertikal jedoch nur eine bestimmte Dicke besitzt (Abb. 3.8B). Die oben beschriebene Auffächerung der Thematik in Ebenen bzw. Folien entspricht übrigens genau dieser Anschauungsweise.

Am häufigsten in einem raumbezogenen Informationssystem sind aber kombinierte geometrisch-topologisch-thematische Abfragen. Der Suchquader ist in diesem Fall durch den mengentheoretischen Durchschnitt aus der topologischen Säule und der thematischen Schicht gegeben (Abb. 3.8C).

3.3. Die Thematik in Vektormodellen

Im Vektormodell läßt sich die Thematik nicht so einfach wie im Rastermodell einbringen. Dort geschah dies durch Hinzufügen einer oder mehrerer Dimensionen zum topologischen Modell, wie wir im vorigen Abschnitt sahen. Hier müssen wir einen anderen Weg einschlagen, der zwar nicht so geradlinig, dafür aber flexibler und (vom Speicherplatzbedarf her gesehen) ökonomischer ist; der Gegensatz zwischen Raster- und Vektormodellen tritt natürlich auch hier zutage, und die jeweiligen Vorzüge werden durch entsprechende Nachteile aufgewogen.

Wir wollen zunächst den Fall annehmen, daß es sich um ein "einfaches" Modell handelt, das mit wenigen Themen, vielleicht sogar nur mit einem Thema ausgestattet ist. Zu Beginn dieses Kapitels haben wir das Beispiel der Grundstücke einer Gemeinde gewählt, für die nur die Nutzung gespeichert wird. Es genügt, für jede mögliche Art der Nutzung einen Code (einen Zahlenwert oder eine bestimmte Zeichenkombination) zu vergeben:

F-100 = Ackerland
 F-200 = Wiese
 F-300 = Garten
 F-400 = Bebauung
 usw.

Jedes Grundstück erhält dann zusätzlich zu seiner geometrischen Beschreibung (Koordinaten der Eckpunkte) und zur topologischen Beschreibung (die Reihenfolge, in welcher die Eckpunkte verbunden werden) einen solchen Code zugeordnet. Wenn wir eine **thematische Karte** dieser Gemeinde mit dem Thema "Flächennutzung" zeichnen, würde die Nutzung durch entsprechende Farben bzw. Schraffuren gekennzeichnet werden, die dem geometrisch-topologischen Grundinhalt der Karte überlagert werden (Abb. 3.9).

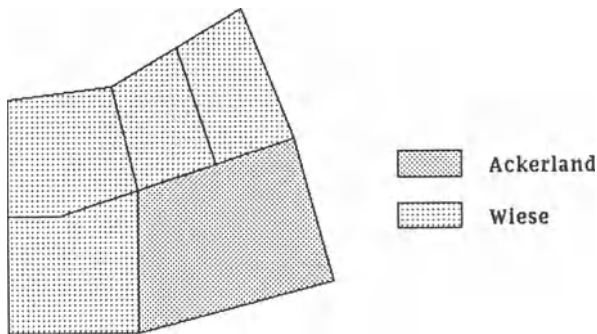


Abb. 3.9. Grundstücke und ihre Nutzung
 (Darstellung durch Flächen)

Aber nicht nur die (topologischen) Flächen lassen sich in dieser Weise klassifizieren; auch die Linien, die diese Flächen einrahmen, können wir dementsprechend einteilen. Wenn wir für die Flächen nur die Nutzung festhalten, ergibt sich natürlich für Linien nur die Thematik "Kulturgrenze"; diese Thematik wird dann für die Zeichnung wieder in einen bestimmten Linientyp, eine Linienbreite, eine Farbe usw. umgemünzt. Wir können aber auch für verschiedene Nutzungen verschiedene Liniensignaturen einführen, um auch ein linienhaftes Bild der Thematik erzeugen zu können (die Flächen bleiben dann leer); ein Beweggrund dafür mag sein, daß wir in die Flächen einen Text schreiben wollen, der nicht von der Farbe verdeckt werden soll, und daß Freistellungs- und Schraffurprogramme oft aufwendig sind. In diesem Fall ist es notwendig, für eine Linie, die zwei unterschiedlich genutzte Flächen trennt, die Priorität festzulegen. Als Beispiel: eine Linie, die "Wiese" und "Acker" trennt, wird immer wie "Wiese" dargestellt; dann ergeben sich allerdings Schwierigkeiten, wenn wir nur die Ackerflächen der Gemeinde darstellen wollen, denn die

Umrandungen der entstehenden Flächen werden unterschiedlich dargestellt. Flexibler sind wir, wenn wir jeder Linie einen Kombinationscode aus beiden angrenzenden Flächen geben (Abb. 3.10).

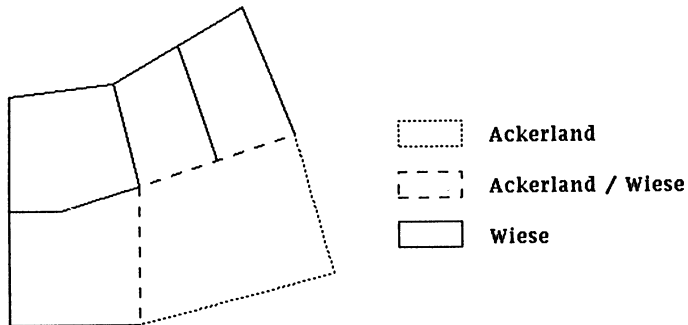


Abb. 3.10. Grundstücke und ihre Nutzung
(Darstellung durch Linien)

L-100/100 = Kulturgrenze Ackerland
 L-100/200 = Kulturgrenze Ackerland / Wiese
 L-200/200 = Kulturgrenze Wiese usw.

Dies gibt uns auch die Möglichkeit, allen jenen Linien, die zwei gleiche Nutzungen trennen, zwischenzeitlich die Darstellungsart "unsichtbar" zu geben; auf diese Weise können wir grundstücksübergreifende Nutzungen sichtbar machen.

Als Anmerkung sei hier gesagt, daß wir in diesem einfachen Modell die Information über die Thematik der Linien und Punkte grundsätzlich aus der *Topologie* des Modells ableiten können: zu jeder Linie findet man die linke und die rechte Nachbarfläche und damit auch deren Nutzung; das Problem der Prioritäten bei verschiedenen Nutzungen haben wir allerdings dabei immer noch.

Schließlich sei noch erwähnt, daß wir bei Punkten genauso wie bei Flächen und Linien vorgehen können; auch sie lassen sich thematisch klassifizieren. Das gleiche gilt für Texte, Symbole und zusammengesetzte topologische Strukturen, die wir in Kapitel 2 besprochen haben.

Natürlich ist die Einteilung in unserem Beispiel sehr grob, denn oft werden verschiedene Teile ein- und desselben Grundstückes auch verschieden genutzt. Wir müssen daher für jedes Grundstück mehrere Eintragungen vorsehen; wir können diese Datei aber auch aus *Grundstücksteilen mit homogener Nutzung* aufbauen; in diesem Fall würde aber wieder die Information verlorengehen, welche Grundstücksteile zusammen-

gehören. Wir sehen also, daß wir eine Hierarchie hineinbringen müssen: jedes Grundstück besteht aus n Anteilen; jeder Anteil gehört genau zu einem Grundstück.

Bei den Linien ergibt sich ebenfalls eine Hierarchie: Kulturgrenzen, die innerhalb eines Grundstückes verlaufen, und solche, die gleichzeitig Grundstücksgrenzen sind. Eigentlich sind wir schon bei einer zweischichtigen Thematik angelangt: wir haben *administrative Informationen* zu verwalten (Grundstücksflächen, Grundstücksgrenzen, Grundstücksgrenzpunkte), und daneben noch die Information über die *Nutzung* (Nutzungsflächen, Kulturgrenzen, Kulturgrenzpunkte). Beide Themen beziehen sich auf ein- und dieselbe Geometrie und Topologie. Nachdem wir bereits oben erwähnt haben, daß mehrere Nutzungen in einem Grundstück erlaubt sind, können wir nun gleich den allgemeinen Fall zulassen: die Nutzung kann grundstücksübergreifend definiert sein.

Es ergibt sich eine **M:N-Beziehung** ("many to many"; siehe Kapitel 4): Ein Grundstück hat (im allgemeinen) mehrere Nutzungen, eine Nutzung erstreckt sich (im allgemeinen) über mehrere Grundstücke. Eine solche M:N-Beziehung wird in klassischer Weise dahingehend aufgelöst, daß man eine zusätzliche Einheit "Nutzungsanteil eines Grundstückes" einführt und dann eine **1:M-Beziehung** ("one to many"; ein Grundstück hat mehrere solche Anteile, aber ein Anteil kann nur zu einem Grundstück gehören) und eine weitere **1:M-Beziehung** (eine Nutzungsfläche besteht aus mehreren solchen Anteilen, aber jeder Anteil kann nur zu einer Nutzungsfläche gehören) erhält.

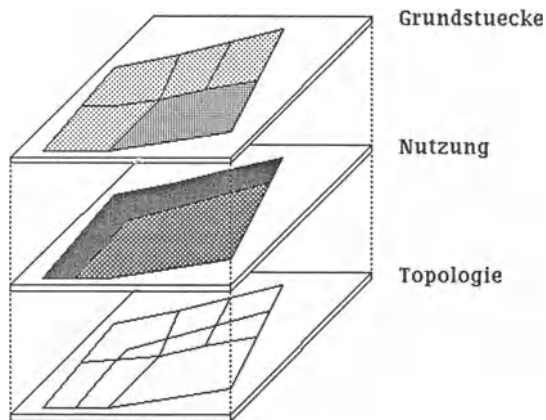


Abb. 3.11. Grundstücke, Nutzung und gemeinsame Topologie

Geometrisch gesprochen ergeben sich diese Nutzungsanteile durch eine **Verschneidung** der beiden Themen "Grundstück" und

"Nutzung" (Abb. 3.11). Das Resultat dieser Verschneidung ist die den beiden Themen zugrunde liegende *Topologie*. Dies gilt natürlich auch, wenn wir das Konzept von zwei auf mehrere Themen verallgemeinern. Die topologischen Bestandteile Punkt, Linie und Fläche werden für jedes Thema in unterschiedlicher Weise zu thematischen Einheiten zusammengefügt; wir können dies auch so formulieren, daß wir sowohl auf der topologischen wie auch auf der thematischen Seite null-, ein- und zweidimensionale Bestandteile verwenden. Um die Begriffe auseinanderzuhalten, können wir für topologische Einheiten die Bezeichnungen "Knoten", "Kante" und "Fläche" und für thematische Einheiten die Bezeichnungen "Punkt", "Linie", "Region" verwenden.

topologische Einheiten <---> thematische Einheiten

Knoten	<--->	Punkte
Kanten	<--->	Linien
Flächen	<--->	Regionen

Diese Art der Bezeichnung ist jedoch nicht allgemein anerkannt; wir werden sie daher nur dann gebrauchen, wenn wir explizit auf die Unterschiede zwischen Topologie und Thematik aufmerksam machen wollen. Ansonsten ist es aus dem Zusammenhang ersichtlich, ob es sich bei einer "Fläche" um deren Topologie im engeren Sinn handelt, oder ob sie als thematischer Zusammenhalt solcher topologischen Bausteine aufzufassen ist. Ein gutes Beispiel für den Unterschied zwischen der Topologie und der Thematik von Flächen bieten die Schweizer Kantone. Es gibt Kantone, die sich aus mehreren disjunkten Teilflächen zusammensetzen (Abb. 3.12). Diese Teilflächen sind als Topologie anzusehen, während die Kantonsbildung natürlich einen thematischen Hintergrund hat.

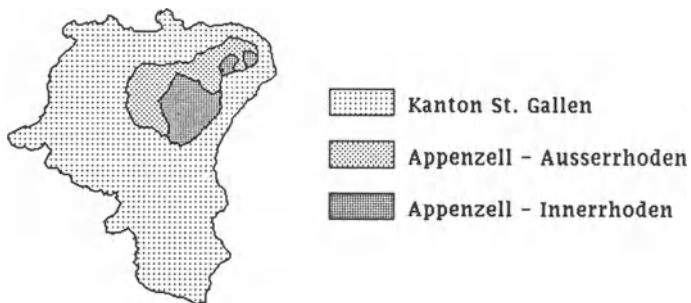


Abb. 3.12. Zusammenhang zwischen thematischen Regionen und topologischen Flächen

Topologische Inseln können ebenso in dieses Konzept eingebaut werden. Wenn wir der Landkarte der Schweiz

einerseits die administrative Gliederung in Kantone und andererseits die Einteilung in Sprachgebiete überlagern, so ergibt sich die Topologie durch die Verschneidung dieser beiden Themen. Während der Kanton Appenzell jedoch als Insel des Kantons St. Gallen auftritt, ist diese Insel in der thematischen Ebene der Sprachgebiete irrelevant; die Region "deutschsprachige Schweiz" hat Appenzell nicht zur Insel (dafür hat sie andere Inseln, die wiederum nicht unbedingt den Kantonsgrenzen entsprechen).

In unserem thematischen Modell lassen sich Inseln mühelos so einbauen, daß wir beim Aufbau der thematischen Region aus topologischen Flächen nicht nur **additive**, sondern auch **subtraktive Operationen** zulassen:

$$\text{Region R} = \text{Fläche F1} + \text{Fläche F2} - \text{Fläche F3}$$

Ähnliches läßt sich über den Zusammenhang zwischen Kanten und Linien sagen. So sind etwa die Routen der öffentlichen Verkehrsmittel thematische Linien; das Thema hat hier die Ausprägungen "Bus", "Straßenbahn", "U-Bahn" usw. Dort, wo sich diese Linien überkreuzen oder gar streckenweise einen gemeinsamen Verlauf nehmen, müssen sie in topologische Kanten aufgetrennt werden (Abb. 3.13).

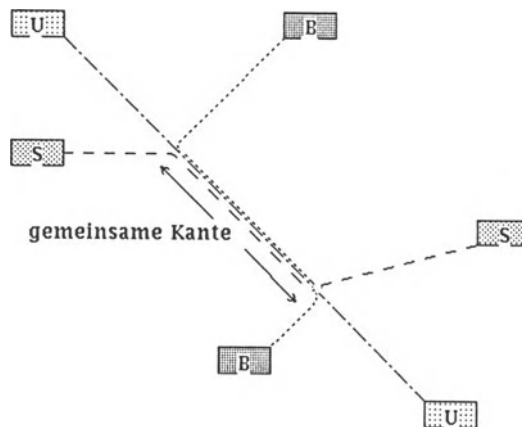


Abb. 3.13. Zusammenhang zwischen thematischen Linien und topologischen Kanten

Das Konzept der Inseln bzw. der disjunkten Teilflächen hat hier im eindimensionalen Bereich ein Gegenstück: Eine U-Bahnlinie, die erst in Teilbereichen fertiggestellt ist, läßt sich sowohl als thematische Aneinanderreihung mehrerer disjunkter Kanten auffassen als auch als "unterbrochene" Linie; die noch im Bau befindlichen Teile der Strecke sind "ein-dimensionale Inseln".

Ist der Zusammenhang zwischen Linien und Kanten bereits um einiges einfacher als zwischen Regionen und Flächen, so wird es bei Punkten und Knoten (beinahe) trivial, obzwar wir hier natürlich am häufigsten auf das Gegeneinander von Topologie und Thematik stoßen; die meisten Punkte in einem Kataster haben mehrere thematische Bedeutungen; ein Grundstücksgrenzpunkt ist oft auch der Eckpunkt eines Hauses, er ist ein Stützpunkt einer elektrischen Freileitung usw. Der Unterschied ist jedoch hier nur mehr konzeptioneller Natur, während es für die Implementierung meist genügt, dem Punkt eine Liste seiner Bedeutungen zuzuordnen.

3.4. Thematische Hierarchien

Bisher haben wir uns nur mit den thematischen Verfeinerungen der Topologie beschäftigt. Wir haben gesehen, daß Punkte, Linien und Flächen eine differenzierte thematische Sichtweise zulassen müssen. *Hierarchische Beziehungen* wurden in diesem stark vereinfachten Modell bisher nicht berücksichtigt, obwohl diese Sichtweise verschiedentlich (wie etwa im Beispiel der Grundstücksnutzungen) angeklungen ist.

Betrachten wir etwa die administrativen Einheiten Land - Bezirk - Gemeinde, so tritt die Hierarchie deutlich zutage. Alle diese Einheiten sind Flächen (in der Terminologie des vorigen Abschnittes sind es thematische Regionen, die ihrerseits aus topologischen Flächen aufgebaut sind). Wir können also drei thematische Ebenen - und zwar jeweils unabhängig voneinander - bilden. Nach dem Verschnitt dieser drei Ebenen erhalten wir als gemeinsamen Nenner die Topologie, die in diesem Beispiel den Flächen der niedrigsten Hierarchie, der Gemeinden also, entspricht. Jede Region der drei Ebenen baut auf diesen Flächen auf.

Wir können aber auch die Tatsache ausnützen, daß wir von der *Thematik her* wissen, daß die Regionen in einem streng hierarchischen Verhältnis zueinander stehen: jede Einheit einer höheren Stufe ist aus Einheiten einer niedrigeren Stufe aufgebaut, wobei von unten nach oben das assoziative Gesetz gilt: Wenn die Gemeinde G zum Bezirk B gehört, und der Bezirk B ein Teil des Landes L ist, so gehört die Gemeinde G zum Land L. Ein solches Schema wird Baum genannt. Jeder Vorgänger hat im allgemeinen mehrere Nachfolger, während jeder Nachfolger genau einen unmittelbaren Vorgänger hat. Zwischen jedem Vorgänger und seinen unmittelbaren Nachfolgern gibt es eine 1:M-Beziehung ("one to many"). Alle Fäden gehen von einer gemeinsamen Wurzel (in unserem Fall das Land) aus. Die Blätter des Baumes sind dort, wo es keine Nachfolger mehr gibt (in unserem Fall sind es die Gemeinden, es sei denn, wir

führen noch weitere Hierarchiestufen ein, zum Beispiel Katastralgemeinden). Dieser Hierarchiebaum wird entweder summarisch durch die Beziehungen zwischen den Datengruppen (Land - Bezirk - Gemeinde) oder detailliert durch die Beziehungen zwischen den jeweiligen Realisierungen (Steiermark - Graz) dargestellt (Abb. 3.14). Hier verweisen wir auf Kapitel 4, wo näher auf die Beziehungen zwischen Datengruppen und Realisierungen eingegangen wird.

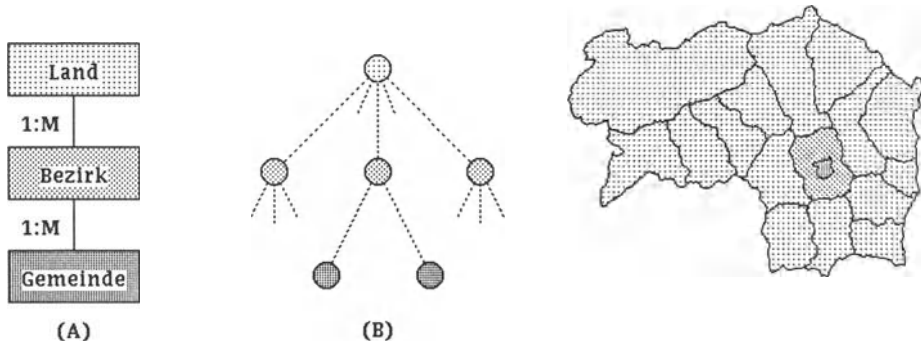


Abb. 3.14. Hierarchische Beziehung zwischen Datengruppen (A) und daraus resultierende Beziehungen zwischen Realisierungen (B)

Wir wollen den Unterschied zwischen den beiden eben erwähnten Vorgangsweisen noch einmal zusammenfassen: während in der ersten Variante eine Beziehung zwischen verschiedenen thematischen Ebenen nur über die zugrunde liegende Topologie zustande kommt, wird in der zweiten Variante eine thematische Hierarchie aufgebaut, deren unterste Stufe erst die Verbindung zur Topologie herstellt. In dieser zweiten Variante werden **Objekte** gebildet. Dies sind komplexe thematische Einheiten, die aus den elementaren thematischen Bestandteilen Region, Linie und Punkt aufgebaut sind. Hier haben wir natürlich wieder den einfachsten Fall vorliegen: jedes Objekt besteht nur aus einer Kategorie von Elementen, nämlich aus Regionen. So gibt es ein Objekt des Typus "Bezirk", das aus Elementen des Typus "Regionen" besteht. Bezirksobjekte werden ihrerseits zu Objekten des Typus "Land" zusammengefügt (Abb. 3.15).

Die Entscheidung für eine der beiden Strategien hat weitreichende Konsequenzen; so kann etwa die Gültigkeit des oben zitierten Sachverhaltes, daß eine Gemeinde nicht gleichzeitig zu zwei Bezirken gehören kann, in Variante 1 nur durch aufwendige Operationen (**Konsistenzüberprüfungen**) nachgeprüft werden, während dies in Variante 2 bereits durch die Objektstruktur vorgegeben ist. Will man andererseits den Flächeninhalt eines Bezirkes ausrechnen, so müssen bei Variante 2 die Flächeninhalte sämtlicher dazwischen

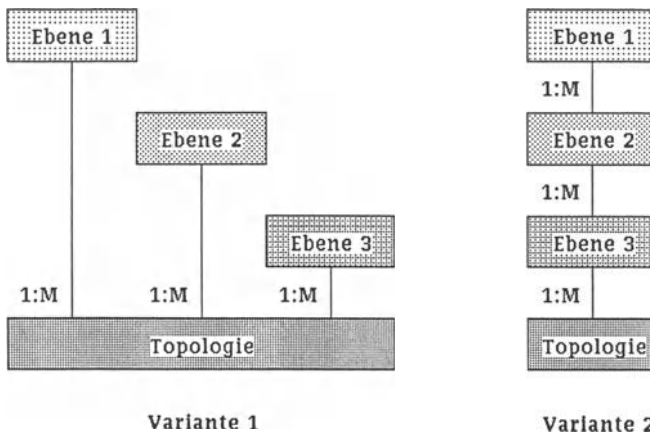


Abb. 3.15. Formen der Flächenhierarchie: implizit über Topologie oder explizit über Objektbildung

liegender Hierarchiestufen ausgerechnet werden, während es in Variante 1 genügt, auf der topologischen Ebene zu operieren. Außerdem kann man schwerlich die Ebene "Bezirk" aus dem Stapel der thematischen Folien herausgreifen, ohne implizit auch die Ebene "Gemeinde" mitziehen zu müssen, was in der Variante 1 jedoch möglich ist, denn dort sind die Ebenen gleichberechtigt; im zweiten Fall sind sie voneinander abhängig.

Wir haben bewußt dieses Beispiel als erstes gewählt, weil es die Möglichkeit offen läßt, die thematische Hierarchie auch über die topologischen Eigenschaften der Elemente zu definieren, und dies die beste Überleitung von den im vorigen Abschnitt besprochenen Rasterstrukturen bietet, wo die Thematik noch enger mit der Topologie verwoben ist. Hier aber ist dies eher eine Ausnahme. Thematische Beziehungen sind wie unsichtbare Fäden, die einzelne Teile der Topologie zusammenhalten. Der Eigentümer mehrerer Grundstücke etwa, die über eine oder mehrere Gemeinden verteilt sind, wird Anlaß zur Bildung eines Objektes "Grundbesitz" geben, das aus mindestens einer Region, im allgemeinen aber aus mehreren Regionen bestehen kann. Die Bauten auf einem Grundstück (Wohnhaus, Garage, Schuppen) werden zu einem Objekt "bebaute Fläche eines Grundstückes" zusammengefaßt. Im Altstadtbereich geben alle diese Verbauungen innerhalb eines Straßenblockes Anlaß zu einem Objekt "Häuserblock". Die Einsatzzentralen einer städtischen Feuerwehr werden zu Objekten "Bereich Süd", "Bereich Ost" usw. zusammengefaßt. Die (punktförmige) Quelle eines Flusses, der (linienförmige) Verlauf und das (flächenförmige) Delta gehören zu einem Objekt "Fluß". Die Beispiele ließen sich beliebig fortsetzen.

Aber auch bei diesen allgemeineren Beispielen wird man im Sinne einer Überschaubarkeit danach trachten, die Baumstruktur in groben Zügen beizubehalten, also die hierarchische Zusammenfassung von (bereits klassifizierten) Details zu größeren Gebilden (dies ist die sogenannte **bottom-up-Methode**) bzw. die Zersplitterung bestehender Gebilde in kleinere Einheiten (**top-down**). Elemente, die (vorerst) nichts miteinander gemein haben, wie etwa Grundstücke einerseits und Versorgungsleitungen andererseits, wird man in gesonderte Ebenen legen; also auch die Ebenen sind hierarchisch angeordnet, so daß sich die Thematik des gesamten raumbezogenen Informationssystems als Baum darstellen läßt.

Die Vorteile einer solchen hierarchischen Strategie und der damit verbundenen baumartigen Anordnung der Realisierungen sind dadurch gegeben, daß der Baum in der Graphentheorie ein wohlbekanntes und gut erforschtes Modell einer Beziehung ist; es gibt auch dementsprechend viele Algorithmen, welche das Einfügen, das Löschen, das Suchen in Bäumen und ähnliche Operationen effizient unterstützen. Ein nicht zu unterschätzendes Kriterium ist das Bild, das sich der Benutzer von den Daten eines Informationssystems aufbaut (ein "externes Schema"; siehe Kapitel 4 und 7). Je größer die Freiheit in der Wahl der Beziehungen ist, desto unsicherer wird der durchschnittliche Anwender in der Handhabung dieser Beziehungen.

Freilich geht es hier mehr um die eingeschränkte Sicht eines speziellen Benützers und weniger um die Gesamtschau der Daten ("konzeptionelles Schema"), die natürlich komplizierter sein wird und auch komplizierter sein muß. Aus diesem Grund wird ein allgemein gehaltenes Modell eines Informationssystems den hierarchischen Aufbau der Thematik zwar als Richtschnur beibehalten, trotzdem aber "kontrollierte Querverweise" zulassen. Ein Beispiel ist ein Grundstück, das nicht nur in seine administrative Oberhoheit, die Gemeinde, eingebettet ist, sondern auch noch in die Schuldenlasten des Grundbuches. Ein weiteres Beispiel ist ein Fluß, der nicht nur im Thema "Gewässer", sondern auch im Thema "Verkehrswege" aufscheint. Daß die Verweise nicht zu unübersichtlich (unter Umständen sogar widersprüchlich) werden, dafür sorgt dann eine entsprechend aufgebaute Datenbank: wenn wir eine relationale Datenbank verwenden, so können wir konzeptionell auch im nachhinein Querverbindungen erlauben, und trotzdem die externe Sicht des Benützers "einfrieren" und ihm seine einfache hierarchische Sicht der Dinge lassen. Auf diese Überlegungen gehen wir in Kapitel 7 näher ein.

3.5. Thematische Attribute

Bisher haben wir unser thematisches Gedankengebäude eher nach einer *qualitativen* Strategie errichtet. Wir haben uns überlegt, welche Themen auftreten können, und wie die Themen miteinander korreliert sind. Nun wollen wir auch der *Quantität* in der thematischen Information den ihr gebührenden Stellenwert einräumen. Es genügt uns nicht, eine Region als Grundstück zu klassifizieren. Wir wollen diesem Grundstück noch beliebig viele andere Informationen zuordnen, die zu seiner näheren Beschreibung dienen, so etwa:

- die *Grundstücksnummer*,
- den *Flächeninhalt*,
- den aktuellen *Verkehrswert*,
- den *Namen des Eigentümers*,
- die *Adresse*,
- eine etwaige *Baubewilligung*,
- etc.

Wir verwenden für diese Informationen summarisch die Bezeichnung **thematische Attribute**, wohl wissend, daß der Ausdruck "Attribut", von verschiedenen Blickwinkeln aus betrachtet, unterschiedlich ausgelegt wird. Im Sinne der relationalen Datenbanken sind alle Spalten einer Tabelle, die eindeutige Eintragungen haben, **Schlüssel**, während die restlichen Spalten als Attribute gelten. In diesem Sinne wäre die Grundstücksnummer wohl ein Schlüssel zur Grundstückstabelle. Der Eigentümername wäre in der Grundstückstabelle ein Attribut, in der Eigentümertabelle hingegen ein Schlüssel. Wir sehen also, daß sich die Grenzen verwischen, vor allem dann, wenn Daten dynamisch verändert und kombiniert werden.

Wir werden daher alles das als Attribut bezeichnen, was eine bestimmte Gruppe thematischer Daten (z.B. die Gruppe der Grundstücke) näher beschreibt, als dies durch den thematischen Code (die thematische Bezeichnung der Gruppe) möglich ist. Attribute erzielen also eine Tiefenwirkung; sie erlauben es, beliebig detaillierte Beschreibungen anzufertigen.

Wenn wir für ein Grundstück die oben angeführten Attribute vorsehen, so bedeutet dies noch nicht, daß im konkreten Fall alle diese Felder tatsächlich besetzt sind. Es gibt nämlich neben **obligatorischen Attributen** (solche, die immer besetzt sein müssen, also hier etwa die Grundstücksnummer) auch **fakultative Attribute** (sie sind von Fall zu Fall besetzt; hier beispielsweise die Information, ob eine Baubewilligung vorliegt oder nicht).

Für jede Gruppe (z.B. Grundstück) gibt es also einen Rahmen (frame) von möglichen Attributen (slots). Für eine konkrete Realisierung aus dieser Gruppe kann es dann Werte (instances) für diese Attribute geben. Regionen, Linien und Punkte können in diesem Zusammenhang ebenso als Gruppen aufgefaßt werden wie komplexe Strukturen (Objekte) (Abb. 3.16 und Abb. 3.17).

Gruppe	Attribut 1	Attribut 2	Attribut n
GRUNDSTUECK	WERT	BAUBEWILLIGUNG	FLAECHE

Abb. 3.16. Rahmen von Attributen für thematische Datengruppen

Realisierung	Instanz 1	Instanz 2	Instanz n
171/3	340.000.-	ja	981 m2
285/2	425.000.-	unbekannt	1024 m2

Abb. 3.17. Instanzen für verschiedene Realisierungen

Attributwerte sind meist alphanumerisch; sie können aber auch numerisch sein. In diesem Zusammenhang sind **berechenbare Attribute** erwähnenswert; deren Werte werden "in Echtzeit" aus der aktuellen Geometrie bzw. Topologie errechnet. So könnte etwa der Flächeninhalt im obigen Beispiel ein solches berechenbares Attribut sein. Ändern sich die Koordinaten eines Eckpunktes, so ändert sich der Attributwert. Berechenbare Attribute erhöhen auch im Rahmen der *Bemaßung* den Anwenderkomfort beträchtlich.

Schließlich sei auch noch vermerkt, daß es eine **hierarchische Fortpflanzung von Attributen** gibt. Wir haben im vorigen Abschnitt die Hierarchie der Objektgruppen besprochen. Fassen wir etwa die Quelle, den Verlauf und das Mündungsdelta eines Flusses zu einem Objekt zusammen, so pflanzt sich das Attribut "Flußname" und damit auch dessen Instanz (z.B. "Donau") auf die Teile dieses Objektes fort. Andere Attribute, wie etwa die Wassergüte, sind zwar für den gesamten Verlauf definiert, ihre Instanzen haben aber in unterschiedlichen Streckenabschnitten auch unterschiedliche Werte. Andere Attribute wiederum (wie etwa die Fläche des

Mündungsdeltas) sind nur für einen Teil des Objektes relevant.

Bei der Definition von Attributen müssen wir auch den Rahmen abstecken, innerhalb dessen sich die Instanzen bewegen können. So ist es beispielsweise klar, daß der Verkehrswert eines Grundstückes größer als Null, und in der jeweiligen Landeswährung eingetragen sein muß; daß die Wassergüte eines Flusses eine der Zahlen 1,2,3,4,5 annehmen muß; daß eine durchschnittliche Temperaturverteilung Celsiusgrade beinhaltet, die in einem bestimmten Bereich bleiben; daß das Datum der letzten Transaktion, die an einem Grundstück durchgeführt wurde, im Format TT-MMM-JJ abgelegt sein muß, wobei die Tage numerisch und die Monate alphanumerisch durch ihre ersten drei Buchstaben angegeben werden, während beim Jahr nur die beiden letzten Ziffern zählen. All diese Bedingungen dienen der Einhaltung der Datenkonsistenz; sie helfen dabei, die Widerspruchsfreiheit der Daten auf lange Sicht zu garantieren.

3.6. Hybride Modelle der Thematik

Wir haben in den vorangegangenen Abschnitten die Thematik in Raster- und Vektormodellen kennengelernt, und die Vor- und Nachteile erwogen. In einem umfassenden raumbezogenen Informationssystem wird es nötig sein, beide Varianten zu berücksichtigen, so wie wir dies bereits im Kapitel 2 für die Topologie gefordert haben.

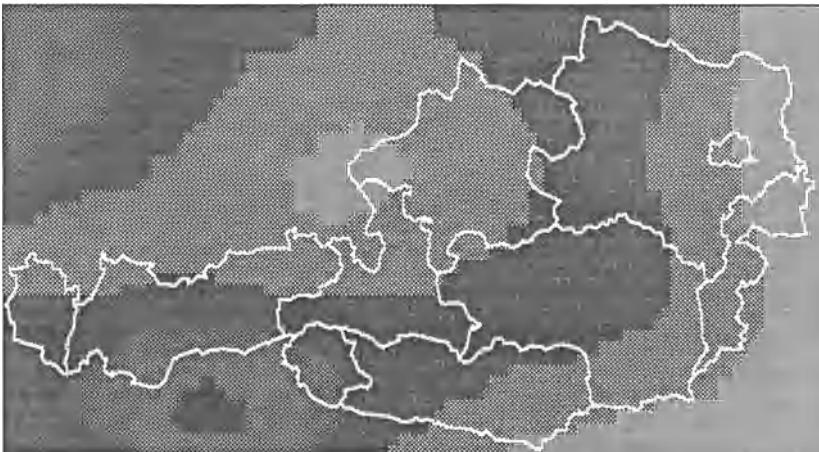


Abb. 3.18. Vektor- und Rasterthematik
(Landesgrenzen und Geoidhöhen)

Es wird also in unserem Stapel der thematischen Ebenen einige geben, die dem Vektormodell entsprechen (z.B. die Ebene des Grundstückskatasters oder die Ebene von Schichtenlinien); andere Ebenen wiederum werden auf einem Rastermodell aufbauen (z.B. ein Orthophoto). Durch das Übereinanderblenden zweier solcher Ebenen entstehen Bilder von hoher Aussagekraft (siehe auch Abb. 3.18). Gerade hier liegt eine der großen Stärken eines digitalen raumbezogenen Informationssystems.

Im allgemeinen wird einer Vektorthematik auch eine Vektortopologie zugrunde liegen, und dasselbe gilt für das Rastermodell. Trotzdem sind natürlich Mischformen denkbar. So müssen wir beispielsweise die Grauwerte eines Rastermodells nicht unbedingt als thematische Aussage auffassen; vielmehr können wir sie als Zeiger in eine Themadatei benutzen, wo dann ausführliche - dem Vektormodell nahestehende - thematische Beschreibungen der jeweiligen Rasterzelle gespeichert sind (Abb. 3.19). Auch Attribute kann man auf diese Weise in eine Rastertopologie einbetten.

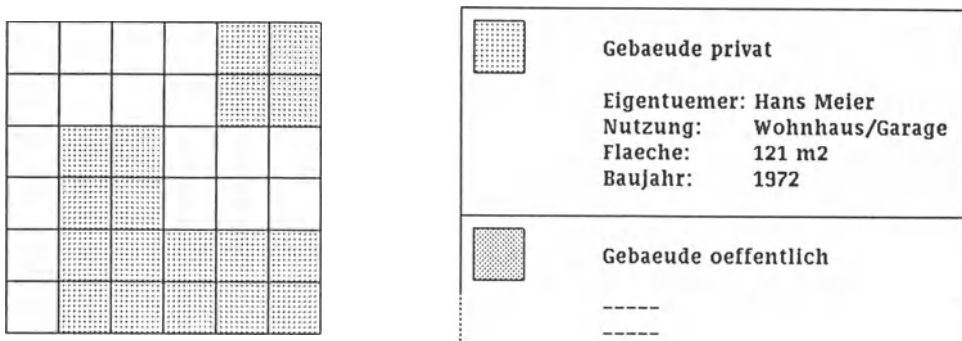


Abb. 3.19. Grauwerte als Zeiger in eine Themadatei

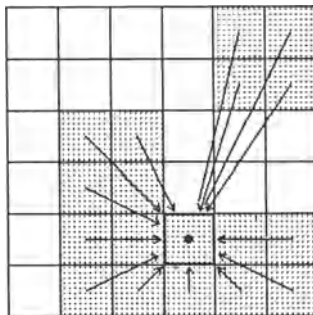


Abb. 3.20. Objektbildung und Zentroid im Rastermodell

Eine weitere Möglichkeit wäre eine Objektbildung in der Rastertopologie, die einen thematischen Zusammenhang zwischen disjunkten Rasterzellen herstellt. Jedes Objekt wird in diesem Fall mit einer Rasterzelle identifiziert, deren Lage für das jeweilige Objekt charakteristisch ist, also etwa im ungefähren Schwerpunkt des Objektes. Diese Rasterzelle wird als Zentroid bezeichnet (Abb. 3.20). Alle anderen Rasterzellen, die zu demselben Objekt gehören, erhalten einen Zeiger zu diesem Zentroid, der formal wieder wie ein Grauwert behandelt wird. Natürlich ist hier wieder der vorhandene Speicherplatz eine Schranke, so daß nur einfache Themen in dieser Weise abgehandelt werden können.

3.7. Thematik und topologische Konsistenzbedingungen

Die Datenkonsistenz ist eine der wichtigsten Forderungen, die wir an ein Informationssystem stellen. Über lange Zeit hin entwickelt es sich dynamisch, es kommen also ständig neue Daten hinzu, und vorhandene Daten werden modifiziert. Wenn wir dieser Entwicklung nicht rigorose Strategien zur Überprüfung der Konsistenz gegenüberstellen können, so wird das Gesamtsystem bald unüberschaubar und für den Anwender wertlos; die Kosten für die Wartung der Daten übersteigen bald jene der Datenerfassung und Auswertung. Das System kann seine Funktion als Auskunftsmedium und Entscheidungshilfe in Fragen der Planung, der Verteilung von Ressourcen, der Verbesserung der Infrastruktur und der Umweltgestaltung nur mehr bedingt wahrnehmen. Inkonsistente Daten verunsichern den Benutzer und verringern die Akzeptanz und somit die Wirtschaftlichkeit des Systems.

Das Problem der Inkonsistenz kommt durch die automatisierte Bearbeitung der Daten zustande. Wenn wir einen Plan manuell anfertigen, so vermeiden wir vieles intuitiv, das zu einer Inkonsistenz der Daten führt. Wir wissen, daß Grundstücke von geschlossenen Linienzügen umrahmt werden, die sich nicht selbst schneiden dürfen; zwischen den Grundstücken darf es kein "Niemandland" geben; Flächen, die zu schmal sind, beruhen höchstwahrscheinlich auf Ungenauigkeiten in der Punktdefinition. Wir wissen auch, daß Häuser einander nicht überlappen; daß Straßen nicht durch Häuser gehen; daß Gemeinden nicht zu zwei Ländern gleichzeitig gehören; daß sich Bäche nicht kreuzen, daß sie im allgemeinen nicht unterbrochen sind, und daß sie keine in sich geschlossenen Schleifen bilden dürfen. Ferner wissen wir, daß Leitungsnetze zusammenhängen müssen, daß sie also keine isolierten Teile haben dürfen. All dies sind topologische Bedingungen, die sich aufgrund der Thematik ergeben. Das jeweilige Thema wirkt sich aber auch auf die Geometrie aus. So wissen wir, daß in einer amerikanischen Reihenhaus-

siedlung alle Häuser rechtwinkelig sind, während ein rechter Winkel in einer europäischen Altstadt eher die Ausnahme ist. Dasselbe läßt sich von Straßenrändern sagen, die zueinander parallel sind, ferner von Gebäuden, deren Abstand von der Straße ein bestimmtes Mindestmaß nicht unterschreiten darf, usw. Diese Beispiele ließen sich beliebig fortsetzen.

Das menschliche Auge erkennt solche Ungereimtheiten sehr genau, denn es ist höchst empfindlich gegenüber geometrischen Abweichungen, und aufgrund der ganzheitlichen Erfassung eines Bildes werden topologische Fehler in Sekundenbruchteilen erfaßt. Oft widersprechen einzelne Bedingungen einander, und der Bearbeiter gewichtet diese und entscheidet dann aufgrund seiner Erfahrung. Das Problem bei der digitalen Erfassung besteht darin, daß sehr viele Daten ohne "menschliche Überwachung" erfaßt werden, und daß die Überprüfung durch herkömmliche Computerprogramme zeitaufwendig ist. Widersprüchliche Bedingungen stellen derzeit oft ein unüberwindliches Hindernis dar.

Ein Ansatz zur Lösung dieses Problems besteht im Einsatz von **Expertensystemen** zur Überprüfung der Datenkonsistenz; wir werden im Kapitel 8 näher darauf eingehen. Trotzdem wollen wir bereits hier untersuchen, wie wir jene Bedingungen, welche die Thematik der Topologie auferlegt, so formalisieren können, daß sie einer EDV-gerechten Behandlung zugeführt werden können. Einfache geometrische Bedingungen hingegen - wie die Eindeutigkeit von Knoten und Kanten - werden in Kapitel 2 behandelt. Komplexe geometrische Bedingungen - wie etwa die Rechtwinkeligkeit und Parallelität zusammengesetzter topologischer Strukturen - werden in Kapitel 6 erwähnt.

Die **Graphentheorie** bietet gute Hilfsmittel zur Formalisierung topologischer Eigenschaften. Die folgende Definition ist dem Buch von Wagner (1970) entnommen:

Ein **Graph** G ist definiert als ein Tripel V, E, z , worin V und E zwei elementfremde Mengen sind, und z für eine Vorschrift steht, durch die jedem Element e von E genau zwei Elemente v von V zugeordnet werden. V steht für die Menge der Knoten (vertices), und E für die Menge der Kanten (edges). Knoten werden also durch Kanten verbunden, und jede Kante hat einen Anfangs- und einen Endknoten. Zwei durch eine Kante verbundene Knoten sind **benachbart**. Hier ist besonders hervorzuheben, daß es vorerst nicht um das Längenmaß für diese Verbindung geht; wir haben also noch keine Metrik eingeführt.

Ein **Weg** in einem Graphen ist eine zusammenhängende Folge von Kanten, die von einem Knoten zu einem anderen Knoten führen. Ein Graph heißt **zusammenhängend**, wenn es zu je zwei beliebigen Knoten mindestens einen Weg gibt (Abb. 3.21).

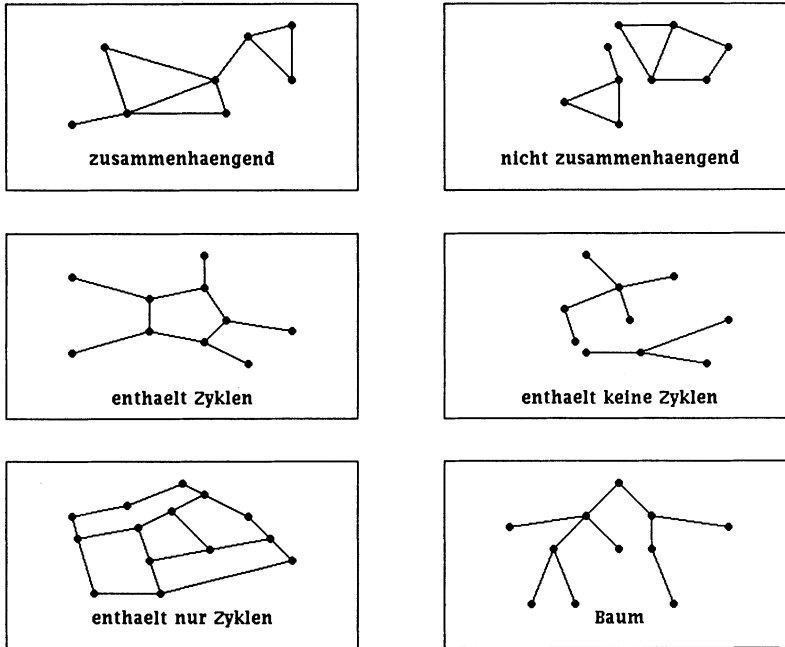


Abb. 3.21. Beispiele für Graphen

Gibt es mehrere Wege, so enthält der Graph einen **Zyklus**, d.h. es gibt einen Knoten, von dem ein Weg ausgeht, der wieder im Knoten endet, wobei die Ausgangskante und die Endkante voneinander verschieden sind. Ein wichtiger Spezialfall ergibt sich, wenn ein Graph **nur Zyklen** enthält, wenn die eben zitierte Eigenschaft also für jeden Knoten gilt. Ist ein Graph zusammenhängend und hat er keine Zyklen, so nennen wir ihn einen **Baum**.

Zwei Graphen sind zueinander **isomorph**, wenn zwischen ihnen eine umkehrbar eindeutige Abbildung existiert, die auch alle Nachbarschaften intakt läßt (Abb. 3.22). Wenn man zu einem Graphen eine isomorphe Entsprechung findet, die sich so in die Ebene abbilden läßt, daß sich keine Kanten schneiden, so ist dies ein **planarer Graph** (Abb. 3.23).

Es stellt sich heraus, daß topologische Konsistenzbedingungen hauptsächlich auf folgenden drei Eigenschaften aufbauen:

- 1) Graph hängt zusammen / hängt nicht zusammen,
- 2) Graph ist planar / nicht planar,
- 3) Graph hat Zyklen / keine Zyklen / nur Zyklen.

So muß etwa der Graph eines Grundstücksplanes zusammenhängen; außerdem muß er planar sein und darf nur Zyklen haben. Für ein Leitungsnetz werden wir hingegen nur den Zusammenhang fordern (Zyklen und Planarität sind hier irrelevant). Gewässerlinien dürfen nur eine baumartige Struktur aufweisen.

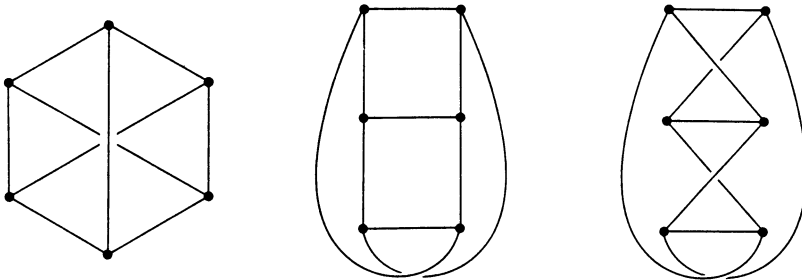


Abb. 3.22. Graphen, die zueinander isomorph sind

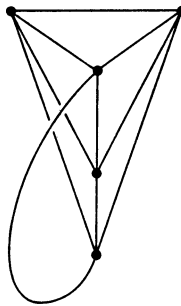


Abb. 3.23. Beispiel für einen nicht-planaren Graphen

An dieser Stelle definieren wir den Begriff der **Transaktion**. Es ist dies eine konsistenzhaltende topologische Operation, welche einen Graphen von einem (konsistenten) Zustand A in einen (konsistenten) Zustand B überführt. Je nach dem topologischen Grundmuster sind Transaktionen unterschiedlich komplex. So darf etwa in einem Leitungsnetz eine Kante hinzugefügt oder entfernt werden, solange der Zusammenhang nicht gefährdet ist. In einem Grundstücksplan jedoch muß die Transaktion die Hinzunahme oder Wegnahme eines gesamten Zyklus umfassen. So darf man ein Grundstück teilen (aus einem Zyklus entstehen zwei oder mehrere), man

darf zwei benachbarte Grundstücke zusammenlegen (zwei Zyklen werden zu einem verschmolzen); man darf aber nicht eine isolierte Kante wegnehmen bzw. hinzufügen. Natürlich ist eine Transaktion selbst wieder aus Elementaroperationen aufgebaut. Wir fordern nur, daß sie als Ganzes durchgeführt werden muß, d.h. wenn sie begonnen wird, muß der entsprechende Datenbereich gesperrt werden; daraufhin wird sie "versuchsweise" durchgeführt; treten dabei keine Widersprüche auf, so wird sie dann definitiv durchgeführt; in jedem Fall wird der Datenbereich daraufhin wieder "geöffnet".



KAPITEL 4
LANGZEITSPEICHERUNG

4.1. Datenbanken: Definitionen und Ziele

In den vorangegangenen Kapiteln haben wir festgestellt, daß raumbezogene Daten sowohl einen *geometrisch-topologischen*, als auch einen *thematischen* Aspekt haben, die einander ergänzen und beeinflussen:

Von der Thematik her gehört ein Grundstück in den Bereich 'Liegenschaftskataster'. Es hat einen (oder mehrere) Besitzer, es hat einen Verkehrswert, eine Nutzungsart, aber auch unter Umständen grundbücherlich festgelegte Belastungen, Nutzungsrechte u.a.m.

Vom geometrisch-topologischen Standpunkt aus betrachtet ist das Grundstück jedoch eine Fläche, die von einer geschlossenen und sich selbst nicht schneidenden Kantenfolge umrahmt ist. Diese Kanten haben im allgemeinen einen Formparameter "geradlinige Verbindung" und bestehen ihrerseits wieder aus einer Folge von Punkten, denen Koordinaten zugeordnet sind.

Würde nun der gesamte Datenbereich nur eine Thematik beschreiben, so könnte man die Thematik der Topologie unterordnen. Jeder topologischen Struktur würde genau eine thematische Bedeutung zugeordnet werden. Im Falle der Grundstücke würde sich der gesamte Datenbereich in eindeutiger Weise als Mosaik von Flächen erzeugen lassen, und jede dieser Flächen hätte die thematische Bedeutung "Grundstück". Man könnte in so einem Fall die thematischen Attribute in ähnlicher Art behandeln wie die geometrischen und topologischen Eigenschaften der Fläche.

Im allgemeinen ist dieser Zusammenhang jedoch nicht so einfach darzustellen. Für ein- und dieselbe Fläche sind im allgemeinen mehrere thematische Bedeutungen aktuell. Das Grundstück tritt auch gleichzeitig als Teil einer Nutzungsfläche in einem Flächenwidmungsplan auf. Es ist ein Teil einer Altstadterhaltungszone. Es besteht selbst wieder aus mehreren Teilflächen usw.

Hinzu kommt, daß vielfältige Zusammenhänge zwischen diesen einzelnen Themen bestehen, die ihrerseits wieder Auswirkungen auf die Topologie haben: So kann man es etwa verbieten, daß ein Grundstück Inseln (im topologischen Sinn; vgl. Kapitel 2) haben kann. Diese Zusammenhänge müssen auf lange Sicht aufrecht erhalten werden. Die Datenbestände müssen **konsistent** gehalten werden. So muß beispielsweise verhindert werden, daß sich Flächen ergeben, die nicht von einem geschlossenen Kantenzug umgeben werden, oder daß Straßen durch Häuser gehen.

Die Daten müssen auch mehreren Anwendersituationen in unterschiedlich ausgeprägtem Detailreichtum gerecht werden, und sie müssen diese Anforderung im Rahmen eines Netzes von gleichzeitig operierenden Arbeitsstationen erfüllen (**Multi-User-Betrieb**).

Schließlich muß auch noch die Problematik des Schutzes der Daten vor (unabsichtlicher oder absichtlicher) Verfälschung, Zerstörung oder mißbräuchlicher Verwendung beachtet werden, besonders wenn es sich um personenbezogene Daten handelt.

Alle diese Erfordernisse bringen als logische Konsequenz die Einbeziehung einer **Datenbank (DB)** mit sich. Die Datenbank und ihre Ziele werden von Frank (1983a) folgendermaßen charakterisiert:

Ein auf Dauer angelegter Datenbestand wird von einem Datenverwaltungssystem geschützt und verschiedenen Benutzern zugänglich gemacht. Die Datenbank besteht somit aus Daten und einem Verwaltungssystem.

Somit ist klargestellt, daß eine Datensammlung erst dann als Datenbank bezeichnet werden kann,

- wenn sie eine strenge Trennung der Daten von den Anwendungsprogrammen ermöglicht,
- wenn sie verschiedenen Anwendern eine individuelle Sicht der Daten bietet, die auf die jeweiligen Erfordernisse eingeht,
- wenn sie Anfragen, Neueintragungen und Korrekturen nur über wohldefinierte Schnittstellen zuläßt und dort gleichzeitig auch eine Überprüfung der Anwender in bezug auf deren Zugriffsberechtigung und der Daten auf ihre Konsistenz vornimmt.

Das Zusammenspiel von Anwenderprogrammen und einer Datenbank, welche die Daten der Anwender verwaltet, wird in Abb. 4.1 beschrieben (siehe auch Pflieger 1982). Anhand dieser Abbildung ist unsere erste Forderung gut begründbar: je weniger ein Anwenderprogramm von der internen Datenstruktur wissen muß, desto besser. Anwenderprogramme sollen grundsätzlich Anfragen an die Datenbank richten, die mit der Formulierung "was wird gebraucht?" umschrieben werden. Die Datenbank ihrerseits bringt dies mit der Beantwortung der Frage "wie kann diese Anforderung realisiert werden?" in Verbindung. Auf diese Weise erreicht man, daß ein Anwenderprogramm leicht gegen ein anderes austauschbar ist. Dasselbe gilt für die Datenbank: falls sie ausgetauscht wird, müssen die Anwenderprogramme nicht neu geschrieben werden.

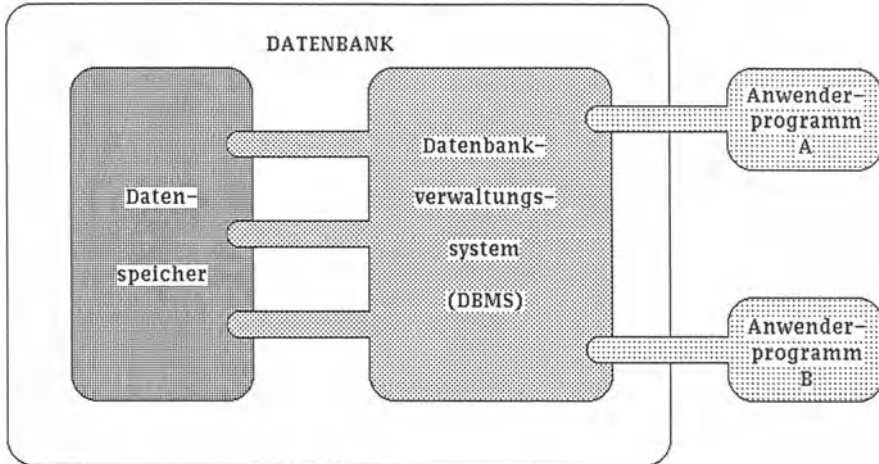


Abb. 4.1. Aufbau einer Datenbank

Die zweite Forderung, die wir oben gestellt haben, gibt Anlaß zur Erklärung der sogenannten *Datenbank-Schemata*.

4.2. Datenbank-Schemata

Man unterscheidet zwischen

- dem **internen Schema**,
- dem **konzeptionellen Schema**
- und dem **externen Schema** einer Datenbank.

Während das interne Schema die physische Gruppierung der Daten und die Speicherplatzbelegung beschreibt, gibt das konzeptionelle Schema die logischen Zusammenhänge der Daten wieder. Mehrere externe Schemata spiegeln bestimmte Teilbereiche des konzeptionellen Schemas wider, die auf die jeweiligen Anwenderanforderungen zugeschnitten sind.

Als Beispiel halten wir uns eine Punktedatei vor Augen. Das **konzeptionelle Schema** legt fest, daß ein Punkt folgende Informationen beinhalten kann:

P-Nr.	X	Y	Z	Typ	Grundstuecks-Nr.
-------	---	---	---	-----	------------------

(4-1)

Ein Anwender hingegen, der die Punkte nur für Detailpunkt-berechnungen braucht, ist nicht an der Grundstücksnummer

interessiert. Das für ihn maßgeschneiderte externe Schema blendet daher diese Information aus. Dies dient nicht nur einer größeren Übersichtlichkeit, sondern es ist auch aus Gründen des Datenschutzes sinnvoll. Nicht jeder Benutzer darf in alle Datenfelder Einsicht nehmen, und nicht jeder darf diese Felder verändern.

Das externe Schema ergibt sich aus dem konzeptionellen Schema nicht nur durch Ausblenden von Informationen. Die Daten werden auch in einem benutzerspezifischen Format ausgegeben. So kann man sich etwa vorstellen, daß in einem RIS alle Punkte in geographischen Koordinaten gespeichert sind (dieses Faktum wäre dem konzeptionellen Schema zuzuordnen). Nun gibt es Anwender dieser Daten, die auch mit geographischen Koordinaten arbeiten, andere wiederum, die Gauß-Krüger-Koordinaten verwenden; daher unterscheiden sich die jeweiligen externen Schemata entsprechend.

Es ist auch denkbar, daß das externe Schema dem Benutzer auch Daten in einem Zusammenhang präsentieren können soll, der im konzeptionellen Schema gar nicht explizit vorgegeben ist. So kann etwa ausgehend von einer Punkttabelle und einer Grundstückstabelle eine Verbindung zwischen den Koordinaten von Punkten und der Fläche eines Grundstückes hergestellt werden, indem die Fläche aus diesen Koordinaten berechnet wird, obwohl das konzeptionelle Schema gar kein Feld für die Fläche eines Grundstückes reserviert hat.

Das interne Schema schließlich gibt an, daß beispielsweise für die Punktnummer 12 Zeichen vergeben werden können, daß Koordinaten in Gleitkomma-Genauigkeit auftreten, und daß die Speicherung der Punkte und Grundstücke in Listen realisiert ist. Jedes dieser Schemata ist für einen anderen Personenkreis interessant:

- die externen Schemata für die Anwender,
- das konzeptionelle Schema für den DB-Administrator,
- und das interne Schema für das DB-Entwicklungsteam.

Der Vorzug der Trennung in verschiedene Schemata kommt zum Tragen, wenn man nun etwa die Speicherung der Punkte, die bisher in Listen erfolgte, auf eine raumbezogene Datenstruktur (siehe Kapitel 5) umändert. Im konzeptionellen Schema (und auch in den externen Schemata und den zugehörigen Anwenderprogrammen) ändert sich dadurch nichts; der Anwender merkt nur, daß Zugriffe auf die Punktlage schneller erfolgen als bisher.

4.3. Beziehungen zwischen Datengruppen

Um der Fülle von Daten Herr zu werden, teilt man sie in Gruppen gleichartiger Datenelemente ein. Wir haben dies bereits getan, als wir von der Gruppe der Punkte und der Gruppe der Grundstücke sprachen. Wir können Beziehungen zwischen Gruppen formulieren und dann zu Recht erwarten, daß sie auch zwischen den einzelnen Datenelementen dieser Gruppe bestehen.

Eine solche Gruppe kann man sich als **Rahmen (Frame)** vorstellen, innerhalb dessen verschiedene **Realisierungen (Instances)** möglich sind. Oft wird in diesem Zusammenhang bildlich und sehr treffend von "Momentaufnahmen" des Inhaltes eines Rahmens gesprochen.

Ein Beispiel möge den Zusammenhang zwischen dem Rahmen und seinen möglichen Realisierungen verdeutlichen:

RAHMEN	REALISIERUNG
PUNKT	P-123 P-124 -----
EIGENTUEMER	Hans Meier Erna Gruber -----

(4-2)

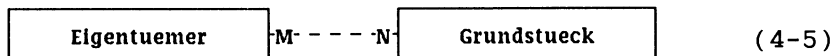
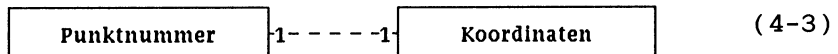
Wenn man nun die Beziehungen zwischen einzelnen Datengruppen untersucht, so wird man feststellen, daß gewisse Beziehungen umkehrbar eindeutig sind. Zum Beispiel gibt es zu jeder Punktnummer genau ein Koordinatentripel und umgekehrt. Dies wird im Bereich der Datenbanken als **1:1-Beziehung** ("one to one") bezeichnet.

Es gibt aber auch viele Beziehungen, die einem Element einer Gruppe im allgemeinen mehrere Elemente einer anderen Gruppe zuordnen. Als Beispiel sei die Beziehung Bundesland - Gemeinde angeführt. Dies ist eine **1:M-Beziehung** ("one to many"). Sie erlaubt grundsätzlich auch, daß für gewisse Datenelemente doch wieder nur ein Partner gefunden wird (wie etwa im Fall des Bundeslandes Wien, das gleichzeitig auch Gemeinde ist), bzw. sogar, daß manche Datenelemente keinen Partner finden.

Am kompliziertesten wird der Fall, wenn eine **M:N-Beziehung** ("many to many") vorliegt. Als Beispiel sei die Beziehung zwischen der Gruppe der Grundstücke und der Gruppe der Eigentümer genannt. Es kann Grundstücke geben, die mehrere Eigentümer haben. Ein Eigentümer kann aber auch an mehreren Grundstücken beteiligt sein.

Für jede Beziehung, die definiert wird, muß demnach auch festgelegt werden, welchem der drei Fälle 1:1, 1:M, M:N sie entspricht. Die unterschiedliche Problematik ergibt sich auch sofort, wenn wir - unabhängig von unseren derzeitigen Überlegungen bezüglich einer Datenbank - eine lokale Datei für ein Anwenderprogramm programmieren wollen. Wenn wir wissen, daß es zu jedem Koordinatentripel genau eine Punktnummer geben muß, so wird ein Feld dafür reserviert. Anders müssen wir vorgehen, wenn ein Punkt der Grenzpunkt mehrerer Grundstücke sein kann; in diesem Fall müssen wir die Möglichkeit schaffen, eine variable Zahl von Feldern für die Eintragung der jeweiligen Grundstücksnummern zu schaffen; wir müssen uns auch auf ein Ungültigkeitszeichen einigen, denn gewisse Punkte sind (noch) gar keinem Grundstück zugeordnet.

Die verschiedenen Varianten der Beziehungen zwischen Datengruppen werden auch oft in folgender Weise dargestellt:



Wir wollen nochmals hervorheben, daß dies Beziehungen zwischen Datengruppen sind, die einen allgemeinen Rahmen abstecken, in den dann einzelne Realisierungen hineinpassen. Obzwar die Gruppen "Eigentümer" und "Grundstück" in einer M:N-Beziehung stehen, kann es einen Eigentümer "Fritz Huber" geben, der mit seinem Grundstück "123/4" in einer 1:1-Beziehung steht.

4.4. Schlüssel, Attribute und Zugriffspfade

Der Zugriff auf Daten einer DB erfolgt über **Schlüssel** ("keys"). Schlüssel sind jene Elemente einer Datengruppe, die für deren Identifizierung geeignet sind, und die für die Verknüpfung verschiedener Datengruppen sorgen.

Der **Primärschlüssel** ("primary key") einer Datengruppe ist mit jedem anderen Element der Datengruppe durch eine 1:1-Beziehung oder durch eine M:1-Beziehung verbunden. Er gibt den **Zugriffspfad** zu den Elementen dieser Gruppe an. Ein Beispiel ist die Punktnummer in einer Punktetabelle (Abb. 4.2).

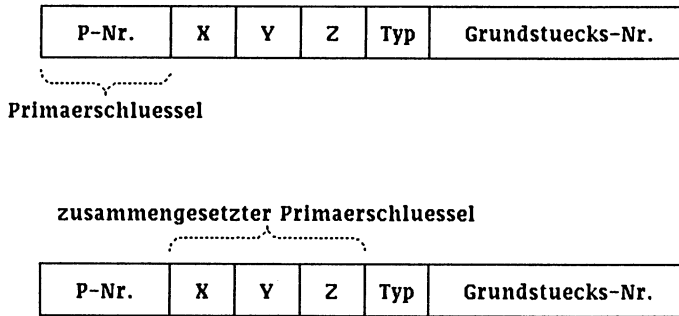


Abb. 4.2. Beispiele für Primärschlüssel

Als Primärschlüssel kann auch eine Kombination von Datenelementen fungieren (**zusammengesetzter Schlüssel** oder concatenated key). In unserem Beispiel kann die Kombination der Elemente X, Y und Z als Primärschlüssel dienen. Dies entspricht einem Zugriff über die Koordinaten (d.h. die Lage) eines Punktes. Ein zusammengesetzter Schlüssel muß allerdings die Forderung erfüllen, daß er "keine überflüssigen Bestandteile enthält"; präziser ausgedrückt, darf man aus ihm kein Element entfernen, ohne daß sein Charakteristikum als Primärschlüssel verlorengeht: die Kombination von X und Y allein kann kein Primärschlüssel sein, weil im allgemeinen mehrere Realisierungen der Punktgruppe diesem Koordinatenpaar entsprechen können.

Wir haben also gesehen, daß es im allgemeinen mehrere Möglichkeiten gibt, einen Primärschlüssel zu vergeben; hier war es entweder die Punktnummer oder das Koordinatentripel. In einem Grundstücksverzeichnis etwa könnte man wahlweise die Grundstücksnummer oder einen inneren Punkt des Grundstückes (z.B. das arithmetische Mittel aller Grenzpunkte) als Primärschlüssel anführen. Es gibt also mehrere **candidate keys** (Kandidaten für einen Primärschlüssel), aus denen einer dann ausgewählt wird. Kennzeichnend für diese Kandidaten ist es, daß sie alle untereinander in 1:1-Beziehungen stehen.

Ein **Sekundärschlüssel** (secondary key) ist ein Schlüssel, der nicht zu den candidate keys gezählt werden kann, weil er mit einem dieser Kandidaten in einer 1:M-Beziehung steht. In unserem Beispiel ist dies der Punkttyp (Abb. 4.3). Es wird

im allgemeinen viele Punkte mit demselben Punkttyp geben. Das Resultat eines Zugriffes über einen Sekundärschlüssel ist dann eben im allgemeinen eine Menge von Punkten.

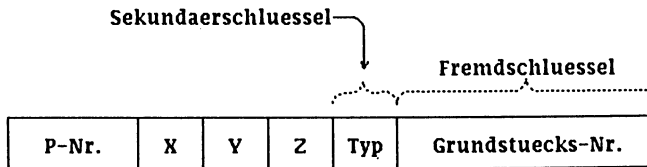


Abb. 4.3. Beispiele für Sekundär- und Fremdschlüssel

Schließlich spricht man auch noch von einem **Fremdschlüssel** (foreign key), wenn das entsprechende Datenelement als Primärschlüssel in einer anderen Datengruppe auftritt. So kann etwa die Grundstücksnummer im Punkteverzeichnis ein Fremdschlüssel sein, weil sie im Grundstücksverzeichnis als Primärschlüssel auftritt (Abb. 4.3).

Eine Datengruppe wird neben den Schlüsseln normalerweise noch andere Informationen beinhalten, die den jeweiligen Repräsentanten einer Gruppe näher beschreiben, selbst aber nicht als Zugriffskriterium verwendet werden. Wir nennen sie **Attribute**. In einem Punkteverzeichnis könnten das Datum der erstmaligen Aufnahme und der Name des Bearbeiters als Attribute auftreten. In einer Gebäudestatistik sind die Anzahl der Stockwerke und das Baujahr Attribute. In einem Grundstücksverzeichnis könnte die postalische Adresse ein Attribut sein.

Allerdings sind Attribute fest an eine Datengruppe gebunden. Würde man nämlich bei den Eigentümern von Grundstücken ebenfalls die Adresse vermerken, so ist sie kein Attribut mehr, sondern muß als eigenständiges Element geführt werden, da man ansonsten Gefahr läuft, daß sich mit der Zeit aufgrund des Fehlens von Querverweisen zwischen Attributen inkompatible Datenbestände anhäufen.

Die Entscheidung über die Gruppierung der Datenelemente, ihre Beziehungen untereinander, sowie die Vergabe von Schlüsseln und Attributen erfolgt bereits in der Entwurfphase der Datenbank. Sie muß sich am Rahmen der Anwendererfordernisse orientieren, und hat dementsprechend weitreichende Konsequenzen. Die Suche nach Punkten über ihre Punktnummern oder über Koordinaten ist ein Beispiel dafür (siehe auch Kapitel 5).

Nachträgliche Änderungen an der Gruppierung von Datenelementen sind - je nach dem gewählten Datenbankmodell (siehe Abschnitte 4.5 bis 4.7) - mehr oder weniger

schwierig. Einige Änderungen (wie etwa die Änderung des Primärschlüssels) bewirken eine Neuorganisation der gesamten Datenbank.

Wir haben bereits oben erwähnt, daß der Zugriff über die Schlüssel erfolgt. Damit ist aber noch nicht alles über die Effizienz des Zugriffes gesagt. Wenn wir das Beispiel des Zugriffes auf Punkte über ihre Nummer betrachten, so ist es einleuchtend, daß die Suche beschleunigt wird, wenn die Punkte nach Nummern geordnet vorliegen; in diesem Fall kann man etwa durch binäres Suchen eine beträchtliche Zeitersparnis erreichen.

Diese Ordnung der Datenelemente kann natürlich physisch verstanden werden; die Punkte sind in einer Tabelle nach Punktnummern aufsteigend sortiert, und die Tabelle ist in dieser Form als Ganzes abgespeichert:

P-Nr.	X	Y	Z	Typ	Grundstuecks-Nr.

P-128					
P-129					
P-130					

(4-6)

Die Ordnung kann aber auch durch das Anlegen eines Index herbeigeführt werden:

P-Nr.	Index	Index	X	Y	Z	Typ	Grundstuecks-Nr.
-----	-----	1					
P-128	2	2					
P-129	1	3					
P-130	5	4					
-----	-----	5					

(4-7)

Dies hat den Vorteil, daß beim Verändern des Punkteverzeichnisses nicht ständig die Speicherplätze umbesetzt werden müssen. So müßte etwa im obigen Beispiel eine Löschung des Punktes P-129 ein Nachrücken aller nachfolgenden Eintragungen nach sich ziehen. Im unteren Fall muß nur die Verkettung der Indices nachgeführt werden. Dies ist im allgemeinen mit viel weniger Aufwand verbunden als die Reorganisation der Daten selber.

Indices müssen grundsätzlich beim Entwurf der Datenbank angelegt werden. Sie richten sich stark nach den Anwendererfordernissen und legen fest, welche Datenelemente hinsichtlich des Zugriffes bevorzugt behandelt werden

sollen. Die nachträgliche Einführung eines Index für ein bestimmtes Datenelement ist wieder je nach dem gewählten Datenbankmodell (siehe Abschnitte 4.5 bis 4.7) mehr oder weniger schwierig. Außerdem ist zu beachten, daß jeder zusätzliche Index das Zeitverhalten der Datenbank belastet.

Der Index hat nicht immer eine solch einfache Gestalt wie im obigen Beispiel. So läßt sich etwa eine raumbezogene Speicherung der Punkte durch Einführung eines zweidimensionalen Index bewerkstelligen. Der gesamte Bereich der zu bearbeitenden Punkte wird in Rechtecke zerlegt. Jedes dieser Rechtecke entspricht einem zusammenhängenden Speicherbereich und enthält im allgemeinen mehrere Punkte. So kann man rasch "in die Nähe" des gewünschten Punktes kommen und lokal im jeweiligen Rechteck detailliert suchen. Natürlich muß diese Methode noch in vieler Hinsicht verfeinert werden, um praxismässige Anforderungen zu genügen. Eine entsprechend detaillierte Schilderung ist in Kapitel 5 zu finden.

4.5. Das hierarchische Datenbankmodell

Jede Datenbank gehört einem von drei Grundtypen an, die sich in der Art und Weise unterscheiden, wie die Beziehungen zwischen einzelnen Datengruppen realisiert werden. Das hierarchische Modell bietet sich für Daten an, die immer nur durch 1:M-Beziehungen verknüpft sind, und bei denen sich diese Verknüpfung fortpflanzt. Querverbindungen sind in einem hierarchischen Modell unmöglich. Als Beispiel führen wir die Beziehung zwischen Verwaltungseinheiten an (siehe Abb. 3.14). Gemäß unseren Überlegungen in den vorigen Abschnitten entspricht diese Beziehung zwischen den (abstrakten) Datengruppen einer baumartigen Beziehung, die zwischen den einzelnen Realisierungen dieser Datengruppen besteht.

Zu jedem Element gibt es genau einen unmittelbaren Vorgänger. Damit äquivalent ist die Forderung, daß zwischen zwei beliebigen Elementen nur eine Verbindung hergestellt werden kann, oder, wenn man den Baum der Elemente als Graphen betrachtet, so darf es in diesem Baum keine Zyklen geben. (Zyklen sind Wege, die von einem Element ausgehen, und auf einem nicht-trivialen Weg wieder in den Anfangspunkt zurückkehren; siehe Abschnitt 3.7).

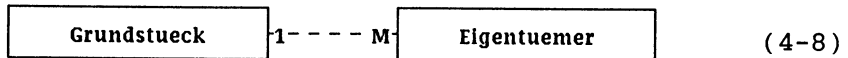
Baumartige Strukturen sind leicht zu modellieren und in eine Datenbank abzubilden; es gibt auch viele hinreichend bekannte und erprobte Hilfsmittel aus der Graphentheorie, die es ermöglichen, effiziente Einfügungs- und Suchstrategien für solche Daten zu entwickeln.

Hierarchische Modelle eignen sich gut für einzelne Themen, sind aber kaum als Datenbank für ein raumbezogenes Informationssystem geeignet. Dieses hat komplexe Beziehungen zwischen verschiedenartigen Daten zu bewältigen, und hierarchische Strukturen kennen von ihrer Natur her keine Querverweise.

Außerdem ist das nachträgliche Einfügen von Beziehungen im allgemeinen nicht durchführbar, weil diese die Hierarchie gefährden.

4.6. Das Netzwerkmodell

Netzartige Strukturen sind allgemeiner als hierarchische Strukturen. In ihnen sind auch Beziehungen möglich, die nicht nur hierarchischer Natur sind. In einer Welt, in der jeder Eigentümer nur Besitzansprüche an ein Grundstück erheben darf, könnte man beispielsweise eine Hierarchie zwischen der Gruppe der Grundstücke und der Gruppe der Eigentümer herstellen.



Im allgemeinen wird es aber so sein, daß die Eindeutigkeit in keiner Richtung gegeben ist: ein Eigentümer kann an mehreren Grundstücken beteiligt sein, und ein Grundstück kann mehrere Eigentümer haben. Die Beziehung ist daher vom Typ M:N



So wie in diesem Beispiel sind fast alle Beziehungen in großen Datenbanken (von wenigen einfachen Ausnahmen abgesehen) netzartig. Aus diesem Grund sprechen viele Autoren nicht von drei, sondern von zwei möglichen Datenbankmodellen, indem sie das hierarchische Modell als spezielle Ausprägung des Netzmodells betrachten, das nur in gewissen Ausnahmesituationen zum Tragen kommt. Datenbanken mit Netzstruktur bezeichnet man auch als CODASYL-Datenbanken (Olle 1981).

Netzmodelle erlauben eine Vielzahl von Verknüpfungen, die effizient gespeichert werden können. Diese Verknüpfungen werden durch **Zeiger (Pointer)** realisiert. Dies sind Verweise zu den Positionen der jeweils zugeordneten Elemente. Dies ist die herkömmliche Art der Speicherung, wie man sie auch für lokale Dateien verwendet.

In einer lokalen Punkt- und Beobachtungsverwaltung etwa wird oft in jedem Punkt ein Hinweis zu jener Position im Beobachtungsverzeichnis vermerkt, wo die erste Beobachtung des in diesem Punkt gemessenen Satzes steht. Bei den Beobachtungen steht vielleicht wieder ein Hinweis zum Standpunkt und zum Zielpunkt. Auch die in (4-7) dargestellte Verknüpfung des Index mit der tatsächlichen Punktinformation folgt dem Netzwerkgedanken.

Haben wir vorher als Vorteil der netzartigen Strukturen eine platzsparende Speichermöglichkeit erwähnt, die auch (zumindest für die vorgesehenen Zugriffspfade) ein rasches Auffinden der Information erleichtert, so liegt ihr großer Nachteil in der zu geringen Dynamik. Sie können sich geänderten Bedingungen nur schwer anpassen. Die Verknüpfung über Pointer erweist sich bei Veränderungen der Daten als Hemmschuh. Je mehr solche Verbindungen es gibt, desto aufwendiger wird eine konsistente Anpassung. Im obigen Beispiel würde es die Veränderung der Punktinformation erfordern, daß man auch im Beobachtungsverzeichnis alle Hinweise zu den Punkten durchforstet.

Die Einführung neuer Datenelemente und neuer Zugriffswege schließlich bedeutet im allgemeinen eine Neuorganisation der gesamten Datenbank.

Zusammenfassend kann man sagen, daß sich das Netzwerkmodell für Daten eignet, die wenig Dynamik haben, und Anwenderumgebungen, die vorhersehbar sind. Dafür kann dieses Modell im Hinblick auf Speicherplatz und Zugriffszeit optimal auf bestimmte Anwendersituationen eingehen.

4.7. Das relationale Datenbankmodell

Das relationale Modell ist das jüngste der bisher vorgestellten Modelle. Es wird auch allgemein als das mächtigste und vielseitigste anerkannt. Für eine übersichtsweise Darstellung dieses Modells sei auf Grill (1982) verwiesen.

Alle Daten werden in Tabellenform gespeichert, so wie wir dies bereits in einigen der oben angeführten Beispiele angenommen haben. Für jede Datengruppe wird eine Tabelle angelegt. So könnte dies etwa eine Punktetabelle sein, wie wir sie in (4-1) formulierten. Die Spalten einer Tabelle werden **Domänen** genannt, die Zeilen werden als **Tupel** bezeichnet. In Abb. 4.4 wird eine Tabelle tachymetrischer Beobachtungen mit ihren Tupeln und Domänen gezeigt.

Entscheidend im relationalen Konzept ist, daß zwei Datengruppen nicht explizit über Pointer (also Speicherpositionen) miteinander verknüpft werden, sondern *implizit* über die

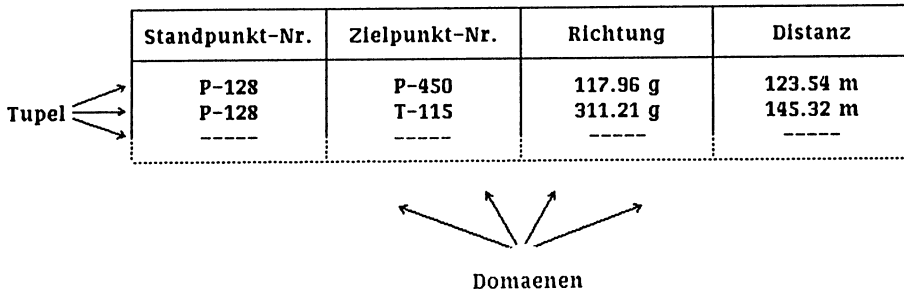


Abb. 4.4. Domänen und Tupel einer Beobachtungstabelle

Werte. In unserem Beispiel ergibt sich für die Beobachtungstabelle ein zusammengesetzter Primärschlüssel aus Standpunkts- und Zielpunktsnummer. Gleichzeitig sind Standpunktsnummer und Zielpunktsnummer, für sich allein betrachtet, Fremdschlüssel zur Punktetabelle.

Diese Konstruktion hat entscheidende Vorteile:

- Die interne Organisation einer Tabelle ist von den anderen Tabellen gänzlich unabhängig. So kann man etwa die Punktetabelle beliebig sortieren, ohne daß dadurch in der Beobachtungstabelle Änderungen nötig wären. Ebenso kann der Punkt neue (z.B. ausgeglichene) Koordinaten bekommen; auch dies wirkt sich nicht auf die Beobachtungstabelle aus.
- Das Anwenderprogramm formuliert eine Anfrage und muß keine Kenntnis davon haben, wie diese Anfrage intern behandelt wird. Das heißt, der Anwender entscheidet über das "WAS?", während der Datenbank die Entscheidung über das "WIE?" zukommt. Diese Art der Kommunikation wird auch als **nicht-prozedural** bezeichnet. Das Anwenderprogramm ist frei von den Problemen des **Navigierens** entlang vordefinierter Datenpfade, was wiederum entscheidende Vorteile bringt. In hierarchischen Modellen und Netzwerkmodellen hingegen muß das Anwenderprogramm sehr wohl über die möglichen internen Datenpfade informiert sein.
- Tabellen entsprechen am ehesten der Art und Weise, wie wir normalerweise Zusammenhänge darstellen. Die Punktetabelle etwa ist jedem Vermessungsingenieur geläufig.
- Tabellen können in einfacher Art und Weise kombiniert, verändert und abgefragt werden. Für diese Operationen gibt es einen eigenen **Kalkül**, also eine mathematisch präzise Beschreibung, die den Regeln der Prädikatenlogik entspricht. Aufbauend auf diesem Kalkül gibt es verschiedene **Zugriffssprachen**, die alle Grunderfordernisse bei der Manipulation der Tabellen abdecken. Sie kommen dabei mit

einigen wenigen Sprachelementen aus und sind daher leicht erlernbar. Wir werden auf diese Sprachen in Abschnitt 4.9.2 eingehen.

- Tabellen lassen sich im nachhinein leichter erweitern als andere Datenstrukturmodelle. So läßt heute bereits jedes relationale Datenbankmodell das Hinzufügen neuer Spalten (Domänen) zu.
- Tabellen erlauben es, benutzerspezifische Ansichten (Views) der Daten herzustellen, so wie wir dies in Abschnitt 4.2 von einem externen Datenbankschema forderten.
- es gibt einfache Regeln, die bei der Erstellung der Tabellen für eine minimale (d.h. möglichst redundanzfreie) Speicherung der Daten sorgen, und die auch eine stabile Datenstruktur begünstigen. Diese Regeln wollen wir im nächsten Abschnitt näher untersuchen.

4.8. Voraussetzungen für stabile Datenstrukturen

Bei der Verwaltung der Tabellen ist die Einhaltung gewisser Bedingungen vorteilhaft. Sie begünstigen eine **stabile Datenstruktur**. Darunter versteht man eine Datenstruktur, die sich veränderten Gegebenheiten anpassen kann, ohne dadurch Inkonsistenzen heraufzubeschwören.

Gerade im Bereich der raumbezogenen Informationssysteme ist der Rahmen der möglichen Anwendungen nicht immer vorhersehbar. Eine Langzeitspeicherung der Daten ist nur dann kostengünstig, wenn diese Daten in vielerlei Hinsicht kombinierbar sind, wenn sie leicht veränderbar sind, und wenn diese Veränderungen nicht unerwünschte Nebeneffekte nach sich ziehen.

Redundant gespeicherte Daten etwa verringern die Stabilität, weil die Gefahr besteht, daß bei einer Veränderung dieser Daten nicht alle Zusammenhänge beachtet werden und gewisse Beziehungen nicht konsistent mitgeführt werden. Man spricht in diesem Fall von "Anomalien" der Datenbank.

Eine stabile Datenstruktur soll auch unempfindlich gegenüber kleinen Störungen sein. Denken wir an die Definition des stabilen Gleichgewichtes; kleine Ursachen dürfen nicht große Wirkungen nach sich ziehen. Wir wollen verhindern, daß wir durch ein fehlerhaftes Anwenderprogramm große Teile der Daten beeinflussen. Die Datenbank soll großzügig auf unvollständige oder unrichtige Anfragen reagieren, indem sie dem Benutzer ein Höchstmaß an Komfort anbietet und solche Anfragen in 'vernünftiger' Weise ergänzt bzw. zurückweist.

Unser Ziel muß es sein, die Beziehungen in einer Datenbank in eine Form zu bringen, welche am ehesten eine Garantie für diese Stabilität bietet. Sie wird auch als **Normalform** bezeichnet und kann verschieden stark ausgeprägt sein. Im relationalen Datenbankmodell läßt sich die Einhaltung der Normalform leichter überprüfen als in anderen Modellen.

4.8.1. Die erste Normalform

Von einer Datenstruktur in der ersten Normalform spricht man dann,

- wenn die Verknüpfung der Daten nicht über physische Adressen, sondern über logische Verweise erfolgt,
- wenn jede Datengruppe einen Primärschlüssel hat,
- und wenn jedes Datenelement einer Datengruppe einen eindeutigen Namen hat, der sich innerhalb derselben Datengruppe nicht wiederholt.

Die erste Forderung ist bei relationalen Modellen natürlich von vornherein erfüllt. Die Forderung nach einem Primärschlüssel ist ebenso einleuchtend: eine Tabelle, in der kein eindeutiger Zugriff möglich ist, hat keine Aussagekraft.

Die letzte Forderung zwingt uns, in einer Tabellenzeile (einem Tupel) höchstens ein Vorkommen eines Datenelementes zu erlauben. So können wir die oben erwähnte 1:M-Beziehung zwischen Bundesland und Gemeinde nicht so speichern, daß das Bundesland als Primärschlüssel fungiert:

Bundesland	Hauptstadt	Gemeinde 1	Gemeinde 2	usw. (4-10)
------------	------------	------------	------------	-------------

Wir würden eine variable Anzahl von Spalten zulassen müssen, und es ist klar, daß die resultierende Liste von Gemeinden viel schwieriger zu manipulieren ist, als wenn man von einer fixen Zahl von Spalten ausgeht. Wir müssen daher die Bestandteile der Tabelle trennen. In einer (Bundesland-) Tabelle ist für jedes Land dessen Hauptstadt angegeben, in einer anderen (Gemeinde-)Tabelle ist für jede Gemeinde das entsprechende Bundesland angeführt:

Bundesland	Hauptstadt	Gemeinde	Bundesland	(4-11)
------------	------------	----------	------------	--------

Es ist somit klar, daß der Primärschlüssel mit jedem anderen

Element der Gruppe in einer M:1-Beziehung stehen muß. In analoger Weise sind wir oben vorgegangen, als wir die Punktetabelle mit der Beobachtungstabelle in Beziehung setzten. Auch dort haben wir es vermieden, für jeden Punkt eine variable Zahl von Beobachtungen zu speichern; vielmehr haben wir den Umstand ausgenützt, daß die Eindeutigkeit in der anderen Richtung gegeben ist, nämlich daß jede Beobachtung genau einen Standpunkt und genau einen Zielpunkt hat:

P-Nr.	X	Y	Z	Typ
-------	---	---	---	-----

(4-12)

Standpunkt-Nr.	Zielpunkt-Nr.	Richtung	Distanz
----------------	---------------	----------	---------

(4-13)

Beim Betrachten dieses Beispiels ergibt sich eine interessante Fragestellung. Es scheint so, als würden wir auf diesem Wege eine Redundanz von Punktnummern erhalten, die wir als unerwünscht hingestellt haben. Was geschieht mit den Beobachtungen eines Punktes, wenn dieser umbenannt wird? Die Verknüpfung über den Wert - früher als Vorteil einer relationalen Struktur apostrophiert - scheint uns hier einen Streich zu spielen. Denn das, was diesen Vorteil ausmacht, nämlich die Unabhängigkeit der beiden Tabellen, scheint hier zum Nachteil zu werden, weil die Beobachtungen dieses Punktes verlorengehen.

Zur Klärung dieses scheinbaren Widerspruches müssen wir unsere Terminologie präziser fassen: wir müssen zwischen dem Schlüssel der Datengruppe PUNKT und dem Punktnamen unterscheiden. Der *Schlüssel* tritt in der Beobachtungstabelle (als Fremdschlüssel) auf, der *Name* jedoch soll dies aus Gründen der Redundanzfreiheit gar nicht. In diesem Fall muß eine dritte Tabelle geschaffen werden, welche die Punktnummer (als Schlüssel) mit dem Punktnamen (der nun dort als Attribut aufgefaßt werden kann) verbindet:

P-Nr.	P-Name
-------	--------

(4-14)

Der (eindeutige) Schlüssel wird intern vergeben; er wird auch als **Surrogat** bezeichnet (Meier et al. 1987), während der Punktnamen vom Anwender vergeben wird.

Nun wenden wir uns den Problemen einer M:N-Beziehung zu, wie sie etwa zwischen Eigentümern und Grundstücken besteht. Hier kann man weder das Grundstück noch den Eigentümer als Primärschlüssel verwenden. Der Fall wird auf klassische Weise so gelöst, daß ein weiteres Datenelement *Grundanteil*

(Teil eines Grundstückes, der einer Person gehört) eingeführt wird, und zwei Tabellen definiert werden:

Anteil	Grundstueck	Anteil	Eigentuerer	(4-15)
--------	-------------	--------	-------------	--------

Somit hat man nun zwei 1:M-Beziehungen. Einerseits besteht jedes Grundstück aus mehreren Anteilen, und jeder Eigentümer kann mehrere Anteile besitzen - jedoch kann jeder Anteil nur für ein Grundstück und für einen Eigentümer geltend gemacht werden.

4.8.2. Die zweite Normalform

Von einer Datenstruktur in der zweiten Normalform spricht man dann,

- wenn die Datenstruktur bereits den Bedingungen der ersten Normalform genügt,
- und wenn jedes Datenelement einer Gruppe funktional voll von einem Schlüssel abhängt (sofern es nicht selbst Teil eines Schlüssels ist).

Dies wird am besten an dem folgenden Beispiel erläutert. Wir legen eine Tabelle von Grundstücken an, auf die über den zusammengesetzten Schlüssel "Gemeinde-Nr. + Grundstücks-Nr." zugegriffen werden kann. (Der zusammengesetzte Schlüssel ist notwendig, weil in Nachbargemeinden dieselben Grundstücksnummern verwendet werden können.)

Gemeinde-Nr.	Grundstuecks-Nr.	Verkehrswert	(4-16)
--------------	------------------	--------------	--------

Würden wir nun eine zusätzliche Spalte "Kleinregion" einführen, so wäre die 2. Normalform verletzt, denn während etwa die Spalte "Verkehrswert" vom gesamten Schlüssel "Gemeinde-Nr. + Grundstücks-Nr." abhängt, würde die Spalte "Kleinregion" nur von einem Teil dieses Schlüssels abhängen, nämlich von "Gemeinde-Nr.". Eine solche Anordnung widerspricht dem Prinzip einer möglichst redundanzfreien minimalen Speicherungsstrategie; der Verweis zur Kleinregion wäre so oft vorhanden, wie es Grundstücke gibt; es genügt aber, die Kleinregion in einer eigenen Tabelle zu berücksichtigen, indem wir sie dort für jede Gemeinde angeben:

Gemeinde-Nr.	Kleinregion	(4-17)
--------------	-------------	--------

4.8.3. Die dritte Normalform

Von einer Datenstruktur in der dritten Normalform spricht man dann,

- wenn die Datenstruktur bereits den Bedingungen der zweiten Normalform genügt,
- und wenn es keine "transitive" Abhängigkeit eines Datenelementes von einem Schlüssel gibt.

Wieder wird dies am besten anhand eines Beispiels erklärt. Erinnern wir uns an die Beziehung Gemeinde(n)-Bundesland-Hauptstadt, wo wir zwei Tabellen Bundesland-Hauptstadt bzw. Gemeinde-Bundesland erstellten, um einer Forderung der ersten Normalform Genüge zu tun - wir konnten die Zeilen der Tabelle nicht nach Bundesländern anordnen, da ansonsten eine variable Anzahl von (Bundesländer-)Spalten entstanden wäre.

Wollen wir nun zwar den Forderungen der ersten Normalform nachkommen, aber die Einrichtung einer zweiten Tabelle vermeiden, so können wir dies in folgender Weise tun:

Gemeinde	Bundesland	Hauptstadt	(4-18)
----------	------------	------------	--------

Dann wäre aber die Bedingung für die dritte Normalform verletzt, weil sich die Hauptstadt bereits aus dem Bundesland ergibt und somit transitiv vom Schlüssel "Gemeinde" abhängt. Die Vermeidung einer zusätzlichen Tabelle ist hier nur ein scheinbarer Gewinn, der einen vermehrten Speicherplatzbedarf auslöst und unnötige Redundanzen mit sich bringt; die Tabelle Bundesland-Hauptstadt wird sehr kurz sein, während eine Spalte "Hauptstadt" in der Gemeindetabelle sehr oft denselben Wert annehmen wird. Über kurz oder lang wird die Redundanz zum Problem, wenn etwa ein Bundesland eine neue Hauptstadt bekommt (wie dies 1986 in Österreich der Fall war); dann muß die gesamte Gemeindetabelle durchforstet werden, während in einer stabilen Datenstruktur in der dritten Normalform nur eine Zeile zu ändern ist, nämlich die Eintragung des Bundeslandes in der kurzen Zusatztable.

Das eben vorgebrachte Beispiel ist lehrreich, obwohl es dem Anwender trivial erscheint. Es ist zu bedenken, daß im konkreten Anwendungsfall sehr vielschichtige Beziehungen auftreten, deren Entflechtung nicht so leicht durchschaubar gemacht werden kann. Beim Entwurf einer solchen Datenbank muß daher die Datenbank-Entwicklungsgruppe gemeinsam mit den künftigen Anwendern eine präzise Darstellung der Beziehungen finden. Außerdem können dann (im Sinne unseres "offenen"

Systems) auch im nachhinein neue Beziehungen entstehen. Die Datenbank eines raumbezogenen Informationssystems kann kein statisches Gebilde sein; sie ist einem dynamischen Wachstums- und Transformationsprozeß unterworfen.

Liegt nun ein relationales Datenbankmodell vor, und sind diese Daten in der dritten Normalform, so kann man unter Beachtung einiger weniger einfacher Regeln aus diesen Daten neue, veränderte Datenstrukturen aufbauen, die ebenfalls der dritten Normalform genügen. Dieser Vorgang wird **kanonische Synthese** genannt.

Wir wollen hier betonen, daß man im Rahmen von Auswertungen zwar sehr oft zusammengesetzte Tabellen erzeugt, die eine oder mehrere dieser Bedingungen verletzen. Sie sind jedoch nur von kurzzeitigem Interesse, während die Tabellen in Normalform als langfristig bedeutsame Elementarbausteine einer stabilen Datenstruktur anzusehen sind.

4.9. Die relationale Algebra

Die Relationen sind Beschreibungen unserer Daten und deren Beziehungen. Wir können (einfache) Relationen als Bausteine für die Zusammensetzung (komplexer) Relationen verwenden. Dazu ist es notwendig, eine Reihe von Operationen zu definieren, die wir auf Relationen anwenden wollen, und ebenso die Regeln, welche diese Operationen steuern sollen. Das Ergebnis jeder Operation ist wieder eine Relation.

Als Analogie kann das Rechnen mit Zahlen herangezogen werden. Werden diese Zahlen den arithmetischen Grundoperationen unterworfen, so entstehen wieder Zahlen. Die Manipulation von Relationen (und speziell die zugrunde liegenden Regeln) bezeichnet man daher auch als *relationale Algebra*.

4.9.1. Elementaroperationen der relationalen Algebra

Die folgenden Elementaroperationen können auf Relationen angewendet werden:

- die *Vereinigung* von Relationen
- die *Differenz* von Relationen
- die *Projektion*
- die *Selektion*
- das *kartesische Produkt* von Relationen

Die **Vereinigung** entspricht dem mengentheoretischen Begriff der Vereinigung zweier Mengen, d.h. es werden alle Zeilen aus beiden Tabellen genommen, wobei doppelt vorkommende Zeilen nur einmal gespeichert werden. Natürlich müssen die beiden Ausgangstabellen gleichartig sein.

So könnte man als Beispiel die Verschmelzung der Daten zweier Projektbearbeitungen durch eine Vereinigung der entsprechenden Tabellen bewerkstelligen. Da sich die beiden Projekte überschneiden können, wird es doppelt vorkommende Elemente geben; diese dürfen im verschmolzenen Projekt nur einmal auftreten.

Wir müssen natürlich anmerken, daß wir hier der Einfachheit halber die Problematik der teilweisen Übereinstimmung unberücksichtigt lassen. So müssen etwa Randanpassungen zwischen verschiedenen Koordinatenpaaren durchgeführt werden, ja, diese Paare müssen zuerst als solche identifiziert werden.

Die **Differenz** zweier Relationen A und B ergibt eine Relation C, die alle Tupel der Relation A enthält, mit Ausnahme derjenigen Tupel, die auch in B vorkommen. Natürlich müssen die Ausgangsrelationen auch hier kompatibel sein, damit wir eine Differenzoperation anwenden können.

Als Beispiel ziehen wir eine Auschnittsbildung aus einer Punktetabelle heran: Neben dieser Punktetabelle möge als Resultat einer Ausgleichung eine Tabelle von ausgeglichenen Punktkoordinaten vorliegen. Die Differenz aus der ursprünglichen Tabelle und dem Resultat des Ausgleiches ergibt genau jene Punkte, die noch nicht dem Ausgleichungsprozeß unterworfen waren.

Dieses Beispiel mag trivial erscheinen. Bedenken wir aber, wie einfach wir mit einer Formel $C = A - B$ einen Vorgang beschreiben können, den wir mit herkömmlichen Programmiermethoden nur durch das Anlegen umständlicher Schleifen durchführen können. Zwar muß der Such- und Vergleichvorgang intern trotzdem durchgeführt werden, aber dies fällt in die Kompetenz der Datenbank, und der Anwendungsprogrammierer braucht sich darum nicht zu kümmern.

Die **Projektion** ist eine Auswahl von Spalten einer Tabelle. So können etwa aus einer Punktetabelle neben der Punktnummer nur X und Y extrahiert werden. Das Ergebnis ist eine Tabelle, die als Projektion der ursprünglichen Tabelle verstanden werden kann:

P-Nr.	X	Y
-------	---	---

(4-19)

In diesem Zusammenhang sei an das externe Schema einer Datenbank erinnert, das im Gegensatz zum konzeptionellen Schema eine eingeschränkte Sicht der Daten bietet (aus Gründen der Übersichtlichkeit oder des Datenschutzes). Die Projektion bietet die dazu nötigen - und höchst einfachen - Werkzeuge.

Die **Selektion** ist eine der wichtigsten Elementaroperationen. Sie erzeugt aus einer Tabelle eine neue, wobei nur jene Tupel übernommen werden, welche eine bestimmte Bedingung erfüllen. Als Beispiel möge die Grundstückstabelle dienen:

Grundstuecks-Nr.	Flaeche	Verkehrswert	
271/2	1215 m2	S 400.000.-	(4-20)
271/3	980 m2	S 370.000.-	
271/4	1149 m2	S 280.000.-	

Nun könnte eine Selektion darin bestehen,² daß man alle Grundstücke auswählt, die größer als 1000 m² sind, jedoch weniger als S 300.000.- kosten. Es ergibt sich eine neue Tabelle

Grundstuecks-Nr.	Flaeche	Verkehrswert	
271/4	1149 m2	S 280.000.-	(4-21)

Hier ist anzumerken, daß in den meisten Fällen aus der resultierenden Tabelle nur einzelne Spalten interessieren, so daß bei der entsprechenden Abfrage auch gleich eine Projektion im Spiel ist. Deshalb verwendet man im Sprachgebrauch den Begriff "Selektion" oft auch für eine kombinierte Zeilen- und Spaltenauswahl.

Die Bedingungen, welche das Ausmaß einer Selektion bestimmen, können sowohl numerische Vergleiche (wie im obigen Beispiel) sein, aber auch alphanumerische Abfragen (Beispiel: Alle Grundstücke, an denen N.N. einen Anteil hat). Die Mächtigkeit einer relationalen Abfragesprache hängt im wesentlichen von den Möglichkeiten ab, die sie der Selektion einräumt.

Das **kartesische Produkt** zweier Relationen A und B erzeugt eine Tabelle, in der jede Zeile von A mit jeder Zeile von B in Verbindung gebracht wird. Dies ist eigentlich immer nur als Zwischenschritt in einer zusammengesetzten Operation sinnvoll. (Die Bedeutungen der Spalten aus A bzw. aus B können im allgemeinen auch paarweise verschieden sein.) Wichtiger ist jedoch eine spezielle Form des kartesischen

Produktes, nämlich der Join zweier Tabellen, der über gemeinsame Werte eine Verbindung herstellt.

Möge eine Tabelle von Punkten und eine Tabelle von Beobachtungen gegeben sein. Wollen wir nun alle Punkte herausfinden, die als Zielpunkte von Beobachtungen auftreten, so müssen wir einen Join der Punktetabelle und der Beobachtungstabelle machen. Dabei werden aus dem kartesischen Produkt aller Punkte und aller Beobachtungen nur jene Zeilen übernommen, bei denen der Inhalt der Spalte "Punktnummer" der Punktetabelle mit dem Inhalt der Spalte "Zielpunktnummer" der Beobachtungstabelle übereinstimmt. Natürlich nimmt man nur eine dieser beiden Spalten in die Ergebnistabelle mit:

Standpunkt	Zielpunkt	X	Y	Z	Typ	Richtung	Distanz
------------	-----------	---	---	---	-----	----------	---------

 (4-22)

Damit hat man für jede Beobachtung die Koordinaten und den Typ des Zielpunktes zur Verfügung.

Suchen wir nun die Punkttypen jener Punkte, die als Zielpunkte von Fernvisuren auftreten, so ist dies eine zusammengesetzte Operation, die in folgender Weise ablaufen könnte (es gibt auch alternative Wege dazu):

1) Selektion jener Zeilen der Beobachtungstabelle B, für welche die gemessene Distanz größer als 1 km ist; es entsteht die Tabelle R1.

2) Join von R1 mit der Punktetabelle P über gemeinsame Punktnummern bzw. Zielpunktnummern; es entsteht die Tabelle R2.

3) Projektion von R2 auf die Spalte "Punkttyp". Es entsteht das Endresultat R3.

Als Bemerkung sei noch angefügt, daß für den Join auch mehrere Spalten aus beiden Tabellen als Kriterium herangezogen werden können.

4.9.2. Sprachelemente für den Zugriff auf eine

relationale Datenbank

Um die im vorigen Abschnitt erwähnten Grundoperationen der relationalen Algebra und die darauf aufbauenden komplexen Operationen durchführen zu können, brauchen wir eine Sprache, die in eindeutiger Weise die Daten und deren Beziehungen beschreiben kann. Die mathematisch präzise

Version einer solchen Sprache finden wir im relationalen Kalkül, der auf der Prädikatenlogik beruht.

Für eine anwendergerechte Formulierung muß dieser Kalkül jedoch in eine einfach aufgebaute und leicht erlernbare Sprache eingebettet sein. Ein Beispiel für eine solche Sprache ist SQL ("structured query language"). SQL ist bereits seit einigen Jahren eingeführt und hat sich zu einem Standard entwickelt, den heute jeder Datenbankanbieter beherrscht. Die nachfolgende Übersicht erhebt keinen Anspruch auf Vollständigkeit, zumal sich die Sprache ständig weiterentwickelt und auch bereits in einer Vielzahl von "Dialekten" an spezifische Anwenderumgebungen angepaßt ist.

Der wichtigste Befehl für den Anwender ist der SELECT-Befehl, mit dem eine Reihe der vorhin beschriebenen Elementaroperationen bewältigt werden können. Er besteht im wesentlichen aus den folgenden Teilen:

```
SELECT (Spalten)
FROM   (Tabellen)
WHERE  (Bedingungen);
```

(4-23)

Ein Beispiel wäre die Auswahl aller Punkte aus der Tabelle PUNKT, deren Punkttyp den Wert PP (Polygonpunkt) hat. Das Ergebnis dieser Auswahl bildet wieder eine Tabelle, die jedoch nur die Punktnummern der gewählten Punkte beinhalten soll. Der entsprechende SQL-Befehl würde lauten:

```
SELECT NR
FROM   PUNKT
WHERE  TYP = 'PP';
```

(4-24)

Als Ergebnis dieser Abfrage erhalten wir die folgende Tabelle:

```
NR
-----
P123
P140
```

(4-25)

Natürlich können mehrere Bedingungen herangezogen werden, die bei der Auswahl beachtet werden sollen. Möchte man etwa die Auswahl noch dahingehend einschränken, daß nur Punkte gewählt werden sollen, deren Höhe weniger als 1000 m ausmacht, so würde die Abfrage so lauten:

```
SELECT NR, XX, YY, ZZ
FROM   PUNKT
WHERE  TYP = 'PP'
AND    ZZ < 1000.000;
```

(4-26)

Als Ergebnis dieser Abfrage erhalten wir die folgende Tabelle (sie enthält mehr Spalten als vorhin, weil wir diesen Wunsch im SELECT-Befehl geäußert haben):

NR	XX	YY	ZZ	
-----	-----	-----	-----	(4-27)
P140	1181.420	423.550	980.142	

Neben numerischen und alphanumerischen Vergleichen kann der Bedingungsteil auch Bereichsangaben (BETWEEN) beinhalten, ebenso Angaben in Listenform (IN) und Angaben über Teile von Zeichenketten (LIKE). Schließlich kann man noch Reihenfolgen erzwingen (ORDER), Mehrfachresultate unterdrücken (DISTINCT) und vieles andere.

Eine besondere Form des SELECT-Befehls kann die Grundoperation des JOINS zweier Tabellen über gemeinsame Werte ausführen. Wir erinnern uns an ein früher gebrachtes Beispiel, bei dem wir die Beobachtungen mit den Koordinaten der Zielpunkte in Verbindung bringen wollten. Das Bindeglied war dort der gemeinsame Wert für die Punktnummer in der Punktetabelle und für die Zielpunktnummer in der Beobachtungstabelle. Mit dem SELECT-Befehl formulieren wir dies so:

```
SELECT STANDP, ZIELP, XX, YY, ZZ, TYP, RI, DIS
FROM   PUNKT, BEOB
WHERE  PUNKT.NR = BEOB.ZIELP;
(4-28)
```

Wir sehen, daß hier zwei Tabellen PUNKT und BEOB verkettet wurden, wobei wir die Zugehörigkeit einzelner Spalten zu Tabellen durch das Zeichen '.' deklariert haben. Das Ergebnis ist wie immer eine Tabelle, die dem Anfragenden als Antwort auf seine Anfrage präsentiert wird. Möchten wir diese Verkettung öfter benützen, so können wir einen View anlegen.

```
CREATE VIEW ZIELCO
AS      SELECT STANDP, ZIELP, XX, YY, ZZ, TYP, RI, DIS
        FROM   PUNKT, BEOB
        WHERE  PUNKT.NR = BEOB.ZIELP;
(4-29)
```

Dies ist eine logische Verknüpfung, die (obzwar sie intern nicht explizit hergestellt wird), unter ihrem Namen (hier ZIELCO) abrufbar ist und fortan wie jede andere Tabelle verwendet werden kann. Damit wird unsere Forderung nach einer benützerspezifischen Sichtweise der Daten voll erfüllt. Der Befehl CREATE VIEW wird nicht mit der Ausgabe einer Tabelle quittiert, sondern mit der Mitteilung "View created." Natürlich können wir eine spezielle Sicht der Daten auch wieder mittels DROP VIEW fallen lassen.

Bevor wir eine Tabelle abfragen können, müssen wir sie zuerst einmal anlegen und dann mit Daten füllen. Dies kann für die Tabelle PUNKT etwa in folgender Weise geschehen:

```
CREATE TABLE PUNKT (NR CHAR (10),
                    XX NUMBER (12,3) NOT NULL,
                    YY NUMBER (12,3) NOT NULL,          (4-30)
                    ZZ NUMBER ( 8,3),
                    TYP CHAR (2));
```

Dieser Befehl würde eine Tabelle mit fünf Spalten anlegen, wobei in der ersten Spalte eine zehnstellige Punktnummer (alphanumerisch) steht; dann folgen drei Koordinatenspalten, und zuletzt eine Spalte für den Punkttyp. Die Koordinaten XX und YY können 12 Stellen umfassen, ZZ kann maximal 8 Stellen haben. Außerdem sind drei Nachkommastellen vorgesehen. Die Anzahl der vorgesehenen Stellen muß nicht immer ausgenutzt werden. Die Datenbank wird dann im allgemeinen den nicht benötigten Platz "einsparen".

Noch eine andere Bedingung haben wir der Tabelle bereits zu Beginn auferlegt: die Koordinatenplätze XX und YY dürfen nicht leer sein (NOT NULL); die Eingabe einer Zeile mit ungültigen X/Y-Koordinaten wird somit verweigert.

Haben wir eine Tabelle angelegt, so können wir zeilenweise Werte einfügen (in unserem Fall sind es Punktinformationen):

```
INSERT INTO PUNKT
VALUES ('KT120', 137540.321, 87421.543, 402.750, 'PP');
... (4-31)
```

Diese Eingabe wird mit der Meldung *"1 record created"* quittiert. Wir können die eingetragenen Werte aber auch elementweise verändern. Soll etwa der Punkttyp des Punktes KT120 von 'PP' (Polygonpunkt) auf 'TP' (trigonometrischer Punkt) geändert werden, so gelingt uns dies mit folgendem Befehl:

```
UPDATE PUNKT
SET     TYP = 'TP'
WHERE  NR = 'KT120';
(4-32)
```

Die WHERE-Klausel haben wir bereits im SELECT-Befehl kennengelernt. Sie tritt hier ebenso auf und kann in derselben Weise für komplexe Bedingungen ausgebaut werden. Wir sehen, wie schnell sich der Benutzer in dieser einfachen Sprache zurechtfindet. Wir sehen aber auch, wie mächtig diese Sprache - trotz ihrer Einfachheit - ist. Denken wir nur daran, wie umständlich ein solches Problem in vielen Programmiersprachen zu lösen ist. Unser Zugang ist

nicht-prozedural, wir können die umständlichen Schleifenzähler herkömmlicher Sprachen vergessen und den Kern des Problems damit treffender formulieren.

Schließlich bleibt noch zu klären, wie wir die Eintragungen in der Datenbank vor absichtlicher oder unbeabsichtigter Zerstörung schützen können. Mittels des GRANT-Befehles können wir bestimmten Benutzern den Zugang zu einer Tabelle gestatten.

```
GRANT SELECT
ON PUNKT
TO HUBER; (4-33)
```

Der Benutzer HUBER darf nun SELECT-Abfragen an die Tabelle PUNKT richten. Natürlich kann man die Berechtigung auch auf INSERT-, UPDATE- und andere Funktionen ausdehnen. Nicht nur für Tabellen ist diese Erteilung einer Zugriffsberechtigung möglich, auch für Views. Berechtigungen können mit einem REVOKE-Befehl jederzeit rückgängig gemacht werden. Wir können auch mittels LOCK gewisse Tabellenbereiche sperren; dies ist dann wichtig, wenn mehrere Anwender gleichzeitig mit denselben Daten arbeiten und einander gegenseitig behindern könnten.

Neben diesen Möglichkeiten eines wirksamen Datenschutzes bietet eine Datenbank aber auch die Führung eines Protokolles ("Log-file") an. Die Konsequenzen eines Systemabsturzes können somit ausgeschaltet werden. Dieser Systemabsturz kann durch den Ausfall einer der Komponenten Hardware / Systemsoftware / Speichermedium / Anwendersoftware bedingt sein. Aufgrund des Protokolles kann man alle Operationen bis zum Zeitpunkt des Absturzes nachvollziehen. Dies wird mit Roll Forward bezeichnet. Umgekehrt gibt es auch die Möglichkeit eines Roll Back, d.h. man macht alle Operationen bis zu einem bestimmten Zeitpunkt rückgängig.

4.10. Die Einbindung in ein Anwenderprogramm

Bisher beschäftigten wir uns mit der Frage, wie die Kommunikation mit der Datenbank in einer interaktiven Benutzerumgebung aussieht. Diese Umgebung beinhaltet ein (nicht unbedingt graphik-fähiges) Terminal und fallweise einen Drucker. Daten können somit interaktiv eingegeben und abgefragt werden.

Ist die Datenbank in ein Anwenderprogramm eingebunden, so muß auch die Möglichkeit geschaffen werden, direkt im Anwenderprogramm die Fähigkeiten der Datenbank ansprechen zu

können. In den meisten gebräuchlichen Programmiersprachen fehlen die dazu nötigen Sprachelemente. Deshalb wird dieses Problem durch einen **Pre-Compiler** gelöst, der vor der sprachspezifischen Übersetzung auf das Programm angewendet wird. Dieser Pre-Compiler übersetzt dann die SQL-Befehle in eine Form, die vom Standard-Compiler der jeweils verwendeten Programmiersprache verstanden wird.

4.11. Relationale Datenbanken und Anwenderkomfort

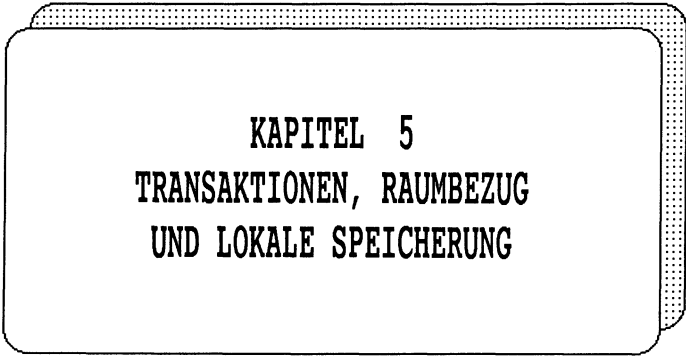
In diesem Kapitel haben wir gesehen, wie wir unsere Daten durch den Einsatz einer Datenbank effizient strukturieren können, wie wir sie vor unabsichtlicher Zerstörung und mißbräuchlicher Verwendung schützen können, und wie wir unsere Anwenderprogramme weitgehend unabhängig von den Daten halten können.

Neben diesen Aspekten der Datenverwaltung bieten Datenbanken (und im speziellen relationale Datenbanken) noch einiges an zusätzlichem Komfort für den Anwender. Darauf wollen wir nun zu sprechen kommen.

- Die **Vernetzung** ist ein häufig zitiertes Kriterium für eine effiziente Nutzung von Hardware und Software. Man versteht darunter ein kontrolliertes Nebeneinander von Anlagen verschiedener Größe (vom PC über Mikro und Mini bis zum Großrechner) mit verschiedenen Betriebssystemen (von MS/DOS bis VMS und UNIX), die es erlauben, ein Problem in Teile aufzuspalten und jeden Teil in der für ihn optimalen Umgebung zu lösen. Dazu muß natürlich vorrangig eine Kompatibilität der Daten gegeben sein; es ist aber auch wünschenswert, daß der Datenzugriff (also das Anwenderprogramm) von Rechner zu Rechner weitergegeben werden kann. Relationale Datenbanken unterstützen diese Forderung weitestgehend, da sie von Grund auf eine strikte Trennung von Anwenderprogrammen und Daten erlauben. Gängige Versionen von relationalen Datenbanken werden auf vielen Rechnertypen und in vielen Betriebssystemen angeboten.
- Ein **Anwendungsgenerator** wird oft als integraler Bestandteil einer solchen Datenbank angeboten. Mit dessen Hilfe kann der Anwender im Rahmen eines einfachen Dialoges (eines Frage- und Antwortspieles) Formulare erstellen, die als interaktive Schnittstelle für den jeweiligen Einsatzbereich verwendet werden können. Daten können somit gewartet und abgefragt werden, ohne daß dazu eine Programmierung im herkömmlichen Sinn notwendig ist. Dazu gibt es dann auch einen **Listprogrammgenerator**, der die Ergebnisse der Abfrage auch noch in eine Form bringt, die (mit Überschriften, speziellen Formaten und Kommentaren

versehen) gedruckt werden kann. Texte und Graphiken können dann zu **Berichten** zusammengefaßt werden, die etwa im geschäftlichen Briefverkehr verwendet werden.

- Ergebnisse von Abfragen können mittels **Farbgraphik** in Form von Diagrammen dargestellt werden, die ebenfalls wieder in Berichte eingespeist werden.
- Alle Tabellen, Anwendungen, Berichte und Graphiken eines Anwenders werden in einem **Datenkatalog** zusammengefaßt, der das Anwenderprofil wiedergibt. Dies sind **Metadaten** (Daten zweiter Stufe, "Daten über Daten"), die aber ihrerseits wieder mit allen Möglichkeiten der Datenbank manipuliert, abgefragt und ausgegeben werden können.



KAPITEL 5
TRANSAKTIONEN, RAUMBEZUG
UND LOKALE SPEICHERUNG

5.1. Raumbezug und Langzeitspeicherung: ein Gegensatz ?

Wir haben uns bereits in Kapitel 4 ausführlich mit der Speicherung raumbezogener Daten auseinandergesetzt und sind zu dem Ergebnis gelangt, daß die Konsistenzerhaltung, der Datenschutz und die Bereitstellung der Daten für unterschiedliche Anwenderprogramme am besten von einer **relationalen Datenbank (RDB)** wahrgenommen werden können, und daß erst durch die Unterstützung einer solchen Datenbank der Betrieb mehrerer im Netzwerk angeordneter Arbeitsstationen möglich ist. Wir können die Datenbank als "black box" betrachten, die über einige (wenige) Ein- und Ausgabekanäle mit den Anwenderprogrammen auf den einzelnen Arbeitsstationen verbunden ist.

Diese weitgehende Unabhängigkeit von der internen Datenorganisation und den dort implementierten Schutzmechanismen hat natürlich ihren Preis, der sich speziell beim Arbeiten auf einer graphisch-interaktiven Arbeitsstation auswirkt. Hier kommt es besonders auf einen schnellen Zugriff über die Lage (**raumbezogener Zugriff**) an. Die geometrisch-topologische Komponente der Daten kommt beim graphisch-interaktiven Arbeiten eher zum Tragen, und Anfragen, die nicht exakt formuliert werden können, sind besonders häufig.

Wenn wir beispielsweise einen Punkt identifizieren möchten, so positionieren wir das Fadenkreuz (den Cursor, die Maus usw.) in dessen Nähe. Die Koordinaten müssen mit den gespeicherten Punktkoordinaten verglichen werden, und der nächstliegende Punkt wird ausgewählt. Wir können also keinen exakten Wert, sondern nur einen Näherungswert für den Schlüssel x,y angeben. Ähnlich verhält es sich, wenn wir ein Grundstück identifizieren und zu diesem Zweck in dessen Mitte hineinzeigen, oder wenn wir alle infrastrukturellen Gegebenheiten aufzeigen möchten, die "in der Nähe" eines bestimmten Ortes liegen. Es handelt sich hier um die Auswertung topologischer Beziehungen, die sehr schnell ablaufen muß, wenn das interaktive Arbeiten attraktiv sein soll.

Dasselbe gilt für den Bildaufbau. Man gibt einen räumlichen Ausschnitt vor - etwa dessen südwestliche und nordöstliche Ecke - und möchte eine bestimmte Auswahl von Themen in diesem Ausschnitt darstellen. Unter Umständen soll die Darstellung sogar noch dynamisch entlang eines definierten Weges mitwandern: dies erscheint etwa bei der Einspiegelung eines Datenbestandes in ein photogrammetrisches Auswertegerät als wünschenswert.

Es erweist sich demnach als günstig, in solchen Situationen die Daten - natürlich nur lokal und für eine beschränkte Zeit - so anzuordnen, daß diese Abfragen sehr schnell befriedigt werden können.

Durch das eben Gesagte könnte der Eindruck entstehen, daß relationale Datenbanken nicht in der Lage sind, auf solche topologische Abfragen einzugehen. Dem ist nicht so: es gibt Sprachelemente, die eine Bereichsangabe ("von - bis") ermöglichen; zur Beschleunigung raumbezogener Abfragen kann man auch einen Index über den Schlüssel "Lage" legen. (Für nähere Einzelheiten sei der Leser auf Kapitel 4 verwiesen.)

Dennoch sind die Akzente bei einer Datenbank eindeutig auf eine möglichst vielseitig nutzbare Langzeitspeicherung ausgerichtet. Eine Datenbank kann - schon von ihrer Definition her - nicht optimal auf eine spezielle Anwendersituation (z.B. eine hochgradig interaktive Editierarbeit) ausgerichtet sein. Bei einem solchen Editiervorgang können sehr viele Daten anfallen; denken wir etwa an das Einfügen einer Schichtenlinie, die Korrektur eines Straßenverlaufes, die Verschneidung des Themas "Waldgebiete" mit dem Thema "Emissionszonen": würden wir alle anfallenden Punkte sofort in die Punktetabelle der Datenbank eintragen, so müßte korrekterweise bei jedem Punkt der Datenschutz- und Konsistenzprüfungsmechanismus in Gang gesetzt werden, und dies würde das interaktive Arbeiten entscheidend verlangsamen. Wir nehmen also bewußt in Kauf, daß wir im Rahmen einer gewissen Zeitspanne sozusagen "ohne Netz" (also ohne Konsistenzprüfungen) arbeiten; am Ende einer solchen Transaktion werden dann alle Änderungen gemeinsam in die Datenbank übernommen; dies kann im Rahmen eines Batchprozesses geschehen, der die interaktive Arbeitsstation nicht belastet und der in betriebsschwachen Zeiten ausgeführt werden kann. Dort werden dann die Konsistenzprüfungen durchgeführt. Treten Inkonsistenzen auf, so gibt uns die Datenbank die Möglichkeit, die Transaktion als Ganzes zu verwerfen oder nur Teile davon zu übernehmen.

Manchmal ist es auch gar nicht notwendig, das Ergebnis einer Transaktion zu speichern, so etwa bei der Planaufbereitung: es geht hier um kartographische Maßnahmen zur optisch ansprechenden Gestaltung eines Planes, wie zum Beispiel die Text- und Symbolplatzierung und -freistellung. Die Ergebnisse sollen in diesem Fall gar nicht in die langfristige Datenspeicherung einfließen; sie können nach Abschluß der Transaktion (d.h. nach der Ausgabe des Planes) gelöscht werden.

Wir bauen also eine lokale Datenstruktur auf, die speziell auf den raumbezogenen Zugriff ausgerichtet ist. Diese Struktur kann keineswegs die Datenbank als Langzeitspeicher ersetzen, sie kann aber besser auf interaktive Arbeitsbedingungen eingehen. Dabei ist es zunächst unwesentlich, ob die Daten physisch neu organisiert werden, oder ob nur ein zusätzlicher raumbezogener Index angelegt wird. Manchmal strebt man sogar eine Mischform an, indem man jene Daten, die häufig benötigt werden, auch physisch

umordnet; für andere - weniger oft angesprochene - Daten wird lediglich ein Index mitgeführt.

Dem aufmerksamen Leser wird nicht entgangen sein, daß wir in den einleitenden Kapiteln immer zwischen Raster- und Vektorstrukturen unterschieden haben, während im Kapitel 4 nur mehr Vektorstrukturen besprochen wurden; auch im jetzigen Abschnitt setzen wir bisher immer ein Vektormodell voraus. Der Grund liegt natürlich darin, daß Datenbanken ihrem Wesen nach Abbildungen vektorhafter Modelle sind; niemand käme auf die Idee, einzelne Pixel in die Datenbank einzutragen; der Speicherplatzbedarf wäre im Vergleich zum Nutzen ungerechtfertigt hoch. Für die Speicherung von Rasterstrukturen eignen sich jene Methoden besser, die bereits in Abschnitt 2.4 kurz vorgestellt wurden. Wir werden sie im folgenden näher besprechen. Ein solches "Bild" (z.B. ein Orthophoto, ein Satellitenbild) kann dann wohl als Ganzes mit Datenbankmethoden verwaltet werden; wird es benötigt, so baut man eine lokale Struktur auf; ansonsten wird es zweckmäßigerweise extern (auf Bändern oder dergleichen) abgelegt.

Diese Methode wird mitunter auch sinngemäß bei Vektorstrukturen angewandt: denken wir an Schichtenlinien, Flußläufe und dergleichen. Dort spielen die Zwischenpunkte eine untergeordnete Rolle. Es sind dies jene Punkte, die keine Knoten im graphentheoretischen Sinne sind (siehe Abschnitt 2.3). Müssen wir sie interaktiv gelegentlich ansprechen, so können sie bei der Langzeitspeicherung in ihrer Gesamtheit (als **Bulk**, d.h. als eine nicht näher aufgeschlüsselte Zusatzinformation mit variabler Länge) abgelegt werden (siehe auch Abb. 5.16). Es genügt also, in der Datenbank die Linie in einer Linientabelle abzulegen, und dieser Linie einen Bulk von Koordinatenpaaren zuzuordnen, der nur lokal in einzelne Punkte aufgelöst wird.

Im nächsten Abschnitt wollen wir uns näher mit lokalen Speichermethoden für Rasterstrukturen auseinandersetzen und dann die Möglichkeiten für Vektorstrukturen ausloten. Hier (wie in anderen Bereichen auch) zeigt sich, daß Rasterstrukturen einfacher zu bewältigen sind, daß man aber auch gewisse Strategien auf Vektorstrukturen ausdehnen kann; dies begünstigt wiederum Mischformen, also das Überlagern von Raster- und Vektordaten (z.B. Orthophoto und Lageplan).

5.2. Speicherung von Rasterstrukturen

Es ist naheliegend, eine Rasterstruktur, die ja eine zeilen- und spaltenmäßige Anordnung von Rasterzellen ist, als Matrix abzuspeichern. (Wir sprechen im folgenden immer vom zweidimensionalen Fall; die Verallgemeinerung auf den dreidimensionalen Fall wäre jedoch einfach.) Eine solche Rasterzelle kann beispielsweise einem Bildpunkt eines Satellitenbildes entsprechen; diesem wird dann ein Grauwert zugeordnet, der in der Matrix abzuspeichern ist. Wir haben jedoch bereits in Abschnitt 3.2 gesehen, daß der Rasterstruktur auch eine regelmäßige Aufteilung eines Gebietes zugrunde liegen kann, wobei der "Grauwert" jeder rechteckigen Zelle die jeweils zugeordnete Thematik widerspiegelt. Wenn diese Thematik etwa 256 mögliche Ausprägungen hat, so müssen wir an Speicherplatz für jede Zelle 8 Bits (1 Byte) vorsehen. Bei einer Anordnung von 500 x 500 Zellen benötigen wir dann bereits einen Speicherplatz von 250 KByte.

Wir sind also gezwungen, einen Weg einzuschlagen, der weniger speicherplatzintensiv ist. Wir können nämlich die Tatsache ausnützen, daß Nachbarzellen im allgemeinen denselben Grauwert besitzen. Bei einem zeilenweisen Durchlauf würde es daher genügen, sich die Stellen zu merken, wo sich der Grauwert verändert. Diese Strategie wird als **Run length encoding** (Lauf längencodierung) bezeichnet. Wir haben sie bereits in Abschnitt 2.4 kennengelernt.

Im allgemeinen kann durch diese Strategie eine beträchtliche Einsparung an Speicherplatz erzielt werden, wenn man von dem Extremfall absieht, daß es keine Nachbarschaften im obigen Sinn gibt. Natürlich wird damit auch gleichzeitig die Verwaltung komplizierter, als sie es beim einfachen Fall der Matrix war. Der Speicherplatzbedarf und der Verwaltungsaufwand verfolgen auch bei allen im folgenden vorgestellten Strategien zwei einander entgegenlaufende Tendenzen, zwischen denen ein vernünftiger Kompromiß gefunden werden muß.

Wir können noch weitere Einsparungen erzielen, wenn wir jene Stellen, wo sich der Grauwert ändert, näher untersuchen. Er wird sich dort meist nur geringfügig ändern. Ein Beispiel: wenn eine Zelle den Grauwert g hat, so sind für die Nachbarzelle die Grauwerte $g+1$ bzw. $g-1$ wahrscheinlicher als die Grauwerte $g+2$, $g-2$ usw. Wir müssen also nur die **Grauwertdifferenzen** speichern. Diese Methode bezeichnet man daher als **differential run length encoding**.

Beide Methoden haben den Nachteil, daß sie zeilenorientiert sind und daher eine Koordinatenrichtung bevorzugen. Dasselbe Problem ergibt sich, wenn wir sie auf die Spalten

anwenden. Man behilft sich manchmal so, daß man für die jeweils unterrepräsentierte Koordinatenrichtung einen Graphen einführt, der den Zusammenhang für diese Richtung wiedergibt. Die Blöcke homogenen Grauwertes einer Zeilencodierung entsprechen den Knoten des Graphen. Zwei Knoten in benachbarten Zeilen sind genau dann durch eine Kante verbunden, wenn sie denselben Grauwert aufweisen und (zumindest teilweise) aneinanderstoßen. Dieser Graph wird **Line adjacency graph** (LAG oder Graph der Linienadjazenzen) genannt (Abb. 5.1).

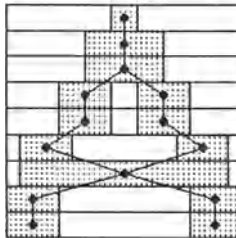


Abb. 5.1. Laulängencodierung und Graph der Linienadjazenzen

Am besten passen wir uns den zweidimensionalen Gegebenheiten an, wenn wir Zeilen und Spalten gleich behandeln und benachbarte Zellen gleichen Grauwertes zu größeren Rechtecken zusammenfassen, egal, ob sich diese Nachbarschaft von links nach rechts oder von oben nach unten ergibt. Wir können dabei sowohl vom Kleinen ins Große vorgehen (Zusammenfassung benachbarter Zellen, die gleiche Grauwerte aufweisen), wie auch umgekehrt: in einem ersten Schritt überprüfen wir, ob es in der Matrix unterschiedliche Grauwerte gibt; wenn nicht, so ergibt sich der triviale Fall, daß ein einheitliches Grauwertbild vorliegt. Ansonsten teilen wir die Matrix in Teilmatrizen und wiederholen den Vorgang für jede dieser Teilmatrizen. Zum Schluß bleiben Teilmatrizen unterschiedlicher Größe, aber mit jeweils homogenen Grauwerten übrig. Dies entspricht dem Aufbau einer **Baumstruktur**. Die Wurzel des Baumes ist die ursprüngliche Matrix, die Blätter sind die übrigbleibenden Teilmatrizen mit homogenem Grauwert.

Es gibt eine Fülle von Varianten, die alle auf diesem gemeinsamen Grundgedanken aufbauen; am bekanntesten sind wohl die **Quad-tree-Methode**, die jeweils vier Nachfolger (links oben, rechts oben, rechts unten, links unten) erzeugt, und der **KD-Baum**, der alternierend einmal links und rechts und dann oben und unten teilt (Abb. 5.2).

Die Speicherung und die Suche in Baumstrukturen sind Standardprobleme der Datenverarbeitung, so daß es hier nicht

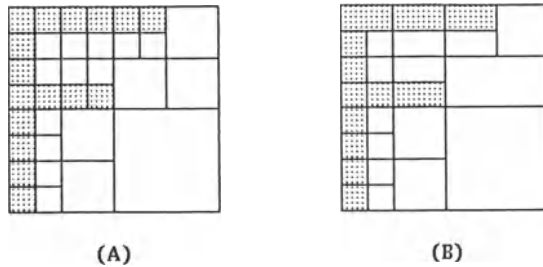


Abb. 5.2. Quad tree (A) und KD-Baum (B)

notwendig ist, darauf näher einzugehen. Der interessierte Leser sei auf Knuth (1972) verwiesen.

Alle bisher vorgestellten Methoden bewirken eine *Komprimierung* und erlauben eine exakte *Rekonstruktion* des ursprünglichen Modells. Wenn die Details im Modell jedoch sekundär gegenüber dem Gesamteindruck sind, so kann man Komprimierungstechniken verwenden, die auf die Reproduzierbarkeit der Einzelheiten verzichten. Ein interessanter Ansatz basiert auf der Theorie der chaotischen dynamischen Systeme. Damit kann man eine Reduktion des Speicherplatzes um das Tausendfache und mehr erreichen. Wir verweisen dazu auf Barnsley u. Sloan (1987).

Bevor wir im nächsten Abschnitt auf die Speicherung von Vektorstrukturen übergehen, wollen wir noch auf eine Mischform hinweisen, die für die Speicherung von linienförmigen Strukturen im Rasterformat geeignet ist. Wir haben auf sie bereits in Abschnitt 2.4 hingewiesen. Es handelt sich um den *chain code* (Kettencode). Wir gehen davon aus, daß ein Hintergrund mit einem einheitlichen Grauwert vorliegt, von dem sich eine Linie abhebt, die selbst wieder durch einen einheitlichen (Vordergrund-) Grauwert gekennzeichnet ist. Die Linie wird in einem Rastermodell in eine Folge von Rasterzellen aufgelöst. Wir können nun zu jeder Zelle, die auf der Linie liegt, die Richtung zur nächsten auf der Linie befindlichen Zelle speichern. Natürlich müssen wir dazu eine "Anfangszelle" kennen. Diese Richtungen können etwa die vier Himmelsrichtungen sein, denen wir die Zahlen 0, 1, 2, 3 zuordnen. Damit genügt es, pro Verbindung 2 Bits an Speicherplatz vorzusehen. Andere "Metriken" verlangen mehr Speicherplatz. Ähnlich wie bei der Lauflängencodierung gibt es auch beim chain code eine differentielle Variante, den *differential chain code*, der nur die Richtungsänderungen berücksichtigt.

5.3. Speicherung von Vektorstrukturen

Wir wenden uns zunächst dem Punkt als einfachsten Repräsentanten einer Vektorstruktur zu. Er ist zugleich auch der wichtigste, denn wir können jede andere Struktur auf Punkte zurückführen. Dies gilt zunächst für topologische Strukturen; aber da in einem raumbezogenen Informationssystem thematische Inhalte ebenfalls letzten Endes auf die Lage - und damit auf die Koordinaten von Punkten - bezogen werden, gewährleisten Punkte die Basis eines raumbezogenen Zugriffs.

Wollen wir einen Punkthaufen so speichern, daß ein raumbezogener Zugriff (ein Zugriff über gegebene oder näherungsweise bekannte Koordinaten) möglich ist, ohne daß alle Punkte sequentiell durchsucht werden müssen, so müssen wir die Punkte der Lage nach ordnen. Wir könnten etwa alle Punkte nach ihren x-Koordinaten sortieren und Punkte gleicher x-Koordinaten sodann nach ihren y-Koordinaten. Nachbarschaften, die entlang der y-Richtung angesiedelt sind, würden in diesem Fall schnell als solche erkannt werden, während andere Himmelsrichtungen stark benachteiligt wären. Aber nicht nur beim Suchen erweist sich diese Strategie als undurchführbar; auch wenn man die Koordinaten eines Punktes geringfügig verschiebt - eine Maßnahme, die in unseren Anwendungen natürlich sehr häufig notwendig ist -, würde dies eine aufwendige Umschichtung der Daten erfordern. Lokale Änderungen würden globale Auswirkungen nach sich ziehen; eine solche Datenstruktur hätte daher einen geringen Stabilitätsgrad. Probleme dieser Art entstehen aber auch bei allen anderen Versuchen, eine (lineare) Ordnung in einen zweidimensionalen Sachverhalt einzubringen.

Es zeigt sich, daß es zweckmäßiger ist, wenn man es mit einer groben Einteilung bewenden läßt, etwa indem man ein *regelmäßiges Maschennetz* über den Bereich legt (Abb. 5.3). Jede Gitterzelle enthält im allgemeinen mehrere Punkte, während umgekehrt jedem Punkt genau eine solche Masche zugeordnet wird (für Punkte, die exakt auf Gitterlinien liegen, legt man beispielsweise fest, daß sie jeweils der östlichen bzw. nördlichen Masche anheimfallen). Diese Methode hat gegenüber den vorher erwähnten den Vorteil, daß sie *flächig* wirkt. Bei Vorgabe eines Suchbereiches ist die Ermittlung der jeweiligen Masche trivial. Im ungünstigsten Fall bewegt man sich am Kreuzungspunkt von vier Maschen. Innerhalb einer solchen Masche muß dann lokal gesucht werden.

Lokale Operationen würden in diesem Fall auch keine globalen Auswirkungen haben. Es ist in den meisten Fällen nur eine Masche beteiligt, und im ungünstigsten Fall sind es vier Maschen. Es bietet sich an, daß der Inhalt einer Masche auch *physisch* benachbart gespeichert wird (etwa in einem

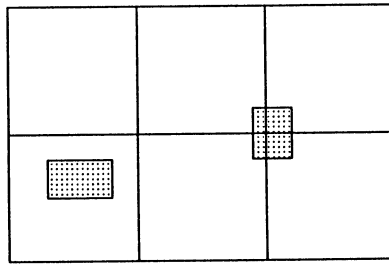


Abb. 5.3. Regelmäßiges Gitter zur Verwaltung eines Punkthaufens; zwei Beispiele für die Vorgabe eines Suchbereiches

Massenspeicherblock). Dies bringt den Vorteil mit sich, daß man mit einem einzigen Diskzugriff "eine ganze Nachbarschaft" mitzieht. Die Diskzugriffe sind ja der (zeitliche) Flaschenhals eines Informationssystems; da die Positionierung des Lesekopfes im Verhältnis zum eigentlichen Lesevorgang viel Zeit kostet, erweist es sich als günstig, die Anzahl der Positionierungen zu minimieren und mit einem Lesevorgang *möglichst viele* Daten zu behandeln. In den Anwendungen, die wir anstreben, ist die *Nachbarschaft* ein häufiges Kriterium; man arbeitet also längere Zeit mit Daten, die in einem solchen Massenspeicherbereich (Page) liegen, oder schlimmstenfalls in einigen wenigen aneinandergrenzenden Pages, bevor man in ein anderes Gebiet wechselt.

Die Anzahl der Maschen, also die Feinheit des Gitters, muß den jeweiligen Erfordernissen angepaßt werden. Zu wenige Maschen würden die Kosten für das lokale Suchen überproportional ansteigen lassen, zu viele Maschen wiederum würden sich ungünstig auf die Verwaltung des Gitters auswirken; zudem würden dann sehr viele Maschen leer sein. Wenn wir außerdem jede Masche mit einer Page des Massenspeichers gleichsetzen, so ergibt sich aus unseren eben angestellten Überlegungen, daß eine Aufteilung in extrem kleine Pages nicht empfehlenswert ist. Ebenso ungünstig ist eine Aufteilung in große Pages, die sehr viele Punkte beinhalten: das Betriebssystem würde - ohne daß wir dies beeinflussen können - eine Aufspaltung in kleinere Teile vornehmen.

Die Gleichsetzung der Masche mit einer Page bringt eine wichtige Frage ins Spiel, die wir bis jetzt nicht berücksichtigt haben: die starre Aufteilung in gleich große Pages ist für unsere Anwendungen ungeeignet, denn zum einen ist die Verteilung der Daten eines raumbezogenen Informationssystems im allgemeinen inhomogen; im städtischen Bereich etwa ist die Größenordnung der zu bewältigenden Daten um vieles höher als anderswo. Und zum anderen entwickelt sich ein raumbezogenes Informationssystem

dynamisch; dies bedeutet, daß wir bei der Anlage eines Datenbestandes nicht wissen, wann und wo Daten im Laufe der Zeit anfallen. Wir müssen daher unser Konzept dahingehend verfeinern, daß es auf die *Inhomogenität* und die *Dynamik* besser eingeht.

5.3.1. Baumstrukturen

Nehmen wir der Einfachheit halber vorerst an, daß wir nur Punktinformationen zu speichern haben, und daß der Platzbedarf für jeden Punkt gleich hoch ist. Beispielsweise seien neben den Koordinaten x, y, z auch eine n -stellige Punktnummer, ein numerischer Punkttyp und das Datum der Erfassung zu speichern. Es ergibt sich somit eine feste Zahl von Punkten, die in einer Page Platz finden. Wenn diese Zahl überschritten wird, so muß ein Überlaufbereich angelegt werden, der die zusätzlichen Punkte aufnimmt. Bei einem hohen Inhomogenitätsgrad werden diese Überlaufbereiche - und damit der Zeitanteil der lokalen Suchstrategien - beträchtlich anwachsen.

Besser ist es, wenn die Ausdehnung der Pages nicht von einer fest vorgegebenen Maschenweite bestimmt wird, sondern vom *Inhalt*, also von der Datenverteilung im jeweiligen Gebiet. Im Fall eines Überlaufes wird die Page *geteilt*. Dies geschieht nach Art einer Zellteilung. Die Punkte der ursprünglichen Page werden auf die einzelnen Folgepages aufgeteilt. Dieser Prozeß der Zellteilung wiederholt sich bei jedem Überlauf, und die so entstandenen Pages weisen unterschiedliche räumliche Ausdehnungen auf, während das Datenvolumen, das sie beherbergen, jeweils durch die oben erwähnte Maximalanzahl beschränkt ist. Es pendelt sich jedoch für eine immer mehr anwachsende Datenmenge asymptotisch bei einer Zweidrittelbelegung ein; in Prozenten ausgedrückt, liegt der Wert in der Nähe von $\ln 2$ (dem Logarithmus der Zahl 2). Die Aufteilung entspricht dem Anlegen einer *Baumstruktur*, eines Graphen, der von einer gemeinsamen Wurzel ausgeht und sich immer mehr verzweigt, ohne daß dabei *Zyklen* entstehen: es gibt also keine Äste, die in irgendwelchen Folgeknoten wieder zusammenlaufen.

Während sich praktisch alle raumbezogenen Speichermechanismen diese *dynamische Aufteilung des Datenraumes gemäß einer Baumstruktur* zunutze machen, gibt es bei der Art der Aufteilung viele Varianten. Der *Quad tree* liegt einer Aufteilung in jeweils vier gleich große Folgepages zugrunde, während der *KD-Baum* - jeweils alternierend - einmal längs der Ostrichtung, das andere Mal längs der Nordrichtung teilt.

Wir haben beide Varianten bereits im vorigen Abschnitt im Rahmen der Speicherung von Rasterstrukturen kennengelernt. Dort stellte jeder Endknoten des Baumes ein Gebiet einheitlicher Färbung dar, während er hier eine Gruppe von Vektordaten beschreibt, die nahe beieinander liegen. Die Färbung, der "Grauwert", ist also ein Symbol für eine grobe räumliche Zuordnung. Es ergibt sich somit die Schlußfolgerung, daß hier - zumindest, was die grobe Einteilung betrifft - Rastermethoden als Vorbild genommen werden. Wir können somit diese Vorgangsweise den **hybriden Methoden** zurechnen (siehe auch Abschnitte 2.5 und 3.6). Global liegt der Methode der rastermäßige Zugang zugrunde, während man im Detail auf die Vektorstruktur der Daten eingeht. Diese Idee läßt sich - wie wir sehen werden - auch auf komplexe Vektorstrukturen erfolgreich anwenden.

Baumstrukturen haben den Vorteil, daß sie, bedingt durch ihren hierarchischen Aufbau, das lokale Verhalten zeigen, das wir vorhin postuliert haben: Beim lokalen Arbeiten bewegen wir uns vornehmlich auf einem Zweig des Baumes bzw. auf einer Gruppe von nahe beieinanderliegenden Zweigen, während die überwiegende Mehrheit der Knoten davon unbetroffen ist. Baumstrukturen können darüber hinaus auch optimal auf einen dynamischen Datenerfassungsprozeß reagieren. Außerdem können wir bei der Implementierung auf eine Fülle von Algorithmen zurückgreifen, die es erlauben, Knoten einzufügen oder zu löschen, zu einem Knoten den Vorgänger oder den Nachfolger zu suchen, oder den Baum zu balancieren; letzteres mag notwendig werden, wenn einige Zweige zu stark belastet werden; das bedeutet, daß sie sich extrem stark verästeln, während sich andere Zweige kaum weiterentwickeln; die Balance wird durch eine Umorganisation von Daten erreicht, so daß die einzelnen Zweige gleichmäßiger ausgelastet sind.

Ein wesentlicher Aspekt bei der Verwaltung von Punktdaten gemäß einer Baumstruktur ist auch die Ermittlung aller jener Pages, die zu einer gegebenen Page *benachbart* sind. Denken wir an den Suchbereich aus Abb. 5.3, der im allgemeinen mehrere Pagegrenzen schneidet. Wir müssen also sehr rasch zu einer Page die angrenzenden Bereiche auffinden können. Auch dafür gibt es eine Fülle von Algorithmen. Wir beschränken uns darauf, auf das Standardwerk von Knuth (1972) zu verweisen, das die Aspekte von Baumstrukturen detailliert darstellt.

Baumstrukturen sind nicht auf die beiden vorgestellten Varianten beschränkt. Eine Aufteilung in jeweils neun Folgebereiche (AID-tree; siehe Rätzsch 1984) trägt einer verfeinerten Metrik Rechnung, die neben den vier Himmelsrichtungen auch noch die Zwischenrichtungen zuläßt (Abb. 5.4). Außerdem bietet der AID-tree den Vorteil, daß er ein zentrales Element besitzt; diese Zentriertheit erleichtert die Anwendung vieler Algorithmen.

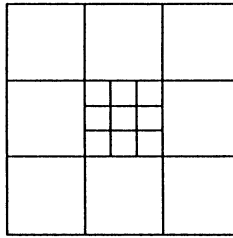


Abb. 5.4. AID-tree und Aufteilung in jeweils neun Folgebereiche

Ferner ist es denkbar, daß Teilungslinien nicht immer exakt in der Mitte, sondern jeweils durch den Schwerpunkt der Punktdaten gezogen werden (Abb. 5.5). Der Vorteil, der sich durch die ökonomischere Auslastung der Pages ergibt, muß dann natürlich durch einen erhöhten Verwaltungsaufwand bezahlt werden.

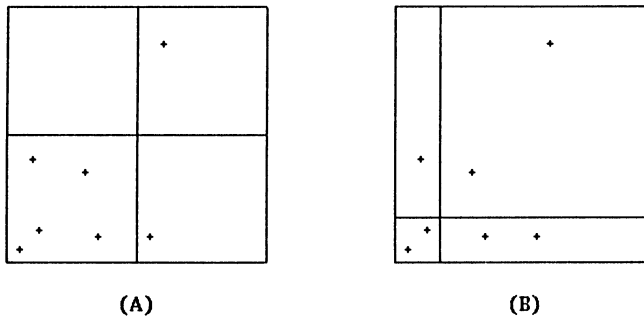


Abb. 5.5. Aufteilung nach geometrischen Kriterien (A) bzw. Rücksichtnahme auf die Datenverteilung (B)

In allen bisher vorgestellten Varianten haben wir stillschweigend vorausgesetzt, daß jeweils alle Informationen, die einem Punkt zugeordnet werden, auch *physisch* in der entsprechenden Page abgespeichert werden. Dies läßt sich sicher in vielen Anwendungen mit Erfolg realisieren. Es ergeben sich auch beträchtliche Vorteile dadurch, daß alle einem bestimmten Punkt zugeteilten Daten durch einen einzigen Ladevorgang bereitgestellt werden können. Wenn aber der Umfang der pro Punkt anfallenden Informationen immer mehr ansteigt, müssen wir sicher eine Einteilung in **primäre** und **sekundäre** Daten vornehmen. Jene Punktinformationen beispielsweise, die wir häufig benötigen (Koordinaten), werden als Primärdaten in den Pages abgespeichert, während die Sekundärdaten (Zusatzinformationen) in irgendwelchen anderen Dateien abgelegt werden (Abb. 5.6). Dies

bedingt natürlich, daß es eindeutige Schlüssel gibt, die den Übergang zwischen den beiden Datenbereichen ermöglichen.

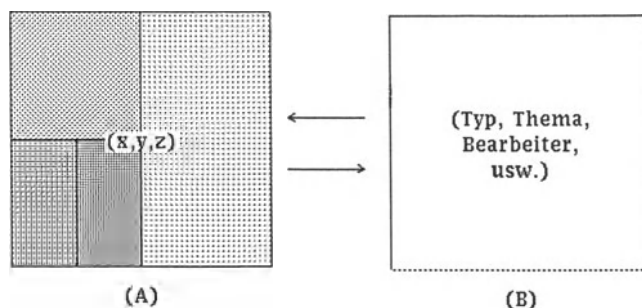


Abb. 5.6. Primärbereich (A) und Sekundärbereich (B) von Punktdaten

So könnten wir etwa die *geometrisch-topologischen* Sachverhalte in den Pages einer Baumstruktur ablegen, während die *Thematik* in einer Datenbank oder in einer datenbank-ähnlichen Filestruktur liegt. Damit würden wir den Raumbezug, der vornehmlich geometrische und topologische Aspekte widerspiegelt, optimal durch die Baumstruktur realisieren, während thematische Abfragen durch eine dafür besser geeignete Abfragesprache einer relationalen Datenbank erledigt werden können. (Siehe dazu auch Abschnitt 5.5.) Wir können die Einteilung in Primär- und Sekundärdaten sogar so weit treiben, daß wir im primären Bereich lediglich die Lage speichern, während alle anderen Informationen im Sekundärbereich liegen. Diese Vorgangsweise entspricht der Erstellung eines zusätzlichen *raumbezogenen Index* für eine bestehende Datenstruktur. Sie stellt somit nur ein Beispiel für die Möglichkeit einer Datenbank dar, für bestimmte Schlüssel eine Beschleunigung des Suchprozesses durch den Aufbau eines solchen Index zu ermöglichen.

5.3.2. Directory-orientierte Methoden

Wir sind in diesem - dem Raumbezug gewidmeten - Kapitel vom (naiven) Konzept der regelmäßigen Aufteilung eines Bereiches in Blöcke gleicher Ausdehnung ausgegangen; die Probleme, die durch die Inhomogenität und die Dynamik der Datenerfassung entstehen, haben wir im vorangegangenen Abschnitt durch die Einführung von Baumstrukturen in den Griff bekommen. Die Regelmäßigkeit des Maschennetzes und die damit verbundenen Annehmlichkeiten beim Suchen mußten wir dabei aufgeben.

Wir fragen deshalb, ob es möglich ist, beide Vorteile (dynamische Aufteilung des Datenraumes und ein regelmäßiges Suchgitter) zu nutzen. Die Antwort darauf gibt die EXCELL-Methode (Tamminen 1981). Dabei wird neben dem - bisher ausschließlich angesprochenen - Datenraum mit seiner dynamischen Page-Einteilung ein Adreßraum angelegt, der aus einem regelmäßigen Suchgitter besteht (Abb. 5.7). Die Maschengröße des Suchgitters entspricht der Ausdehnung der jeweils kleinsten Page des Datenraumes. Dem Suchgitter entspricht eine Matrix ("Directory", daher der Name "directory-orientierte Methoden" für EXCELL und ähnliche Methoden).

Zu Beginn der Speicherung ist der Datenraum leer. Eine einzige Page überdeckt ihn. Die Nummer dieser Page steht in einer 1x1-Matrix. Nun werden Punkte in diese Page gespeichert, bis sie überläuft und geteilt werden muß. Wenn wir die KD-Strategie anwenden, so wird die Page zunächst in eine linke und in eine rechte Hälfte geteilt. Der Adreßraum wird zu diesem Zeitpunkt demnach durch eine 1x2-Matrix repräsentiert, welche die beiden Pagenummern enthält. Läuft eine der beiden Hälften neuerlich über (etwa die linke), so wird sie in einen oberen und einen unteren Teil gespalten. Der Adreßraum ist nun durch eine 2x2-Matrix gegeben. Nach einer neuerlichen Teilung ergibt sich eine 2x4-Matrix, danach eine 4x4-Matrix usw. Bei Verwendung einer Quadtree-Strategie wachsen die Zeilen- und die Spaltenanzahl der Matrix konform. Die Elemente der Matrix beinhalten die Nummer der jeweiligen Page im Datenraum. In Abb. 5.7 werden die Page-Nummern durch unterschiedliche Grauwerte dargestellt. Jeder Gitterzelle des Adreßraumes entspricht genau eine Page des Datenraumes, während sich natürlich eine Page über mehrere Gitterzellen erstrecken kann.

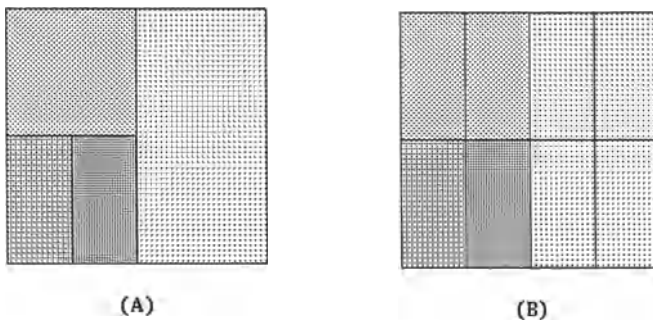


Abb. 5.7. EXCELL: Datenraum (A) und Adreßraum (B)

Die Ermittlung jener Page, die einen bestimmten Punkt x,y überdeckt, ist somit höchst einfach. Zeilen- und Spaltenindices können durch eine einfache Rechnung ermittelt

werden: bezeichnen wir mit L die Gesamtlänge des Bereiches in x -Richtung und mit n die Anzahl der Maschen des Suchgitters längs der x -Richtung, so ergibt sich der Spaltenindex ind durch:

$$\text{ind} = [x * n / L] \quad (\text{aufgerundet}) \quad (5-1)$$

Der Zeilenindex entlang der y -Richtung ergibt sich auf analoge Weise. Am Kreuzungspunkt von Zeile und Spalte steht die Nummer der Page, die den gesuchten Punkt überdeckt. Genauso einfach ist ein Bereichszugriff, bei dem alle Punkte x, y gesucht werden, die in einem vorgegebenen Bereich (x_{\min}, x_{\max}) und (y_{\min}, y_{\max}) liegen: wir müssen das Suchrechteck nur mit der Directory-Matrix schneiden und erhalten so eine Liste von Pagenummern, aus welcher wir nur mehr die Redundanzen eliminieren müssen. Auch alle weiteren räumlichen Zugriffe (etwa Nachbarschaftsbeziehungen) lassen sich durch triviale Algorithmen befriedigen. In vielen Fällen ist diese Art der Suche über Directories der Suche in einer Baumstruktur überlegen.

Das Directory benötigt sehr wenig Speicherplatz; in vielen Fällen kann es sogar im Memory verbleiben, so daß kein zusätzlicher Verwaltungsaufwand anfällt, was den Diskzugriff betrifft. Für den Fall, daß das Directory überläuft, läßt sich die Strategie verallgemeinern: das Directory wird selbst wieder geteilt, und diese Teilung wird in einem "Super-Directory" evident gehalten. Es zeigt sich jedoch in der Praxis, daß solche Überläufe eher selten auftreten, solange die Inhomogenität der Punktdaten nicht allzu kraß ist.

Der **GRID FILE** als zweites Beispiel einer directory-orientierten Methode geht besser auf Inhomogenitäten ein (siehe Hinrichs u. Nievergelt 1983). Die Maschen des Suchgitters sind nicht mehr alle gleich groß. Teilungslinien gehen jedoch immer durch das ganze Gitter, so daß dieses ebenfalls durch eine Matrix dargestellt werden kann. Die Suche nach einem bestimmten Punkt und der Page, die ihn überdeckt, kann allerdings nicht mehr durch eine derart einfache Formel erfolgen wie bei der EXCELL-Methode; vielmehr müssen wir für jede Koordinatenrichtung einen Adreßvektor mitführen, der die Koordinaten der (orthogonal zu dieser Richtung) verlaufenden Teilungslinien beinhaltet (Abb. 5.8). Den Zeilen- und Spaltenindex des Directory können wir durch binäres Suchen in diesen Adreßvektoren ermitteln. Auch hier gilt die Überlegung, daß die Matrix samt den beiden Adreßvektoren in vielen Fällen im Memory Platz findet; schlimmstenfalls genügt es, die beiden Vektoren im Memory zu halten. Auch eine hierarchische Verfeinerung durch einen "Super-Gridfile" ist denkbar.

Bei beiden Varianten - EXCELL und GRID FILE - können Parallelen zur sogenannten Hashing-Methode festgestellt

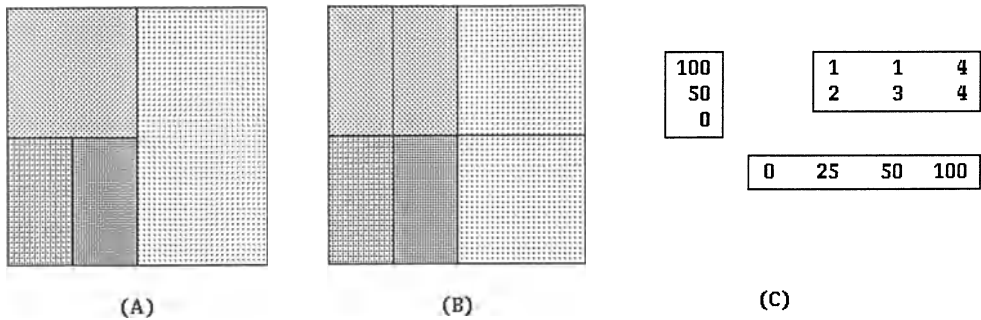


Abb. 5.8. GRIDFILE: Datenraum (A), Adreßraum (B) und Directory (C)

werden, einer Standardmethode zur Speicherung und Auffindung von Informationen gemäß eines vordefinierten Suchbegriffes. Unsere Suchbegriffe sind Koordinatenpaare, weshalb man auch oft vom *zweidimensionalen hashing* spricht. Natürlich können diese Konzepte jederzeit auf drei und mehr Dimensionen verallgemeinert werden. Der dritten Dimension muß dabei nicht unbedingt die Interpretation der Höhe (bzw. z-Koordinate) auferlegt werden. In unserem Beispiel eines Punktspeichers könnte die dritte Dimension etwa der Punktnummer entsprechen. Längs dieser Dimension wird ein Punktnummernindex angelegt. Das (dreidimensionale) Directory können wir als Quader auffassen, der räumliche Zugriffe und Abfragen bezüglich der Punktnummer gleichberechtigt behandelt; wenn wir alle Punkte suchen, die in einem Rechteck liegen, so müssen wir das Directory mit einer vertikalen rechteckigen Säule schneiden, um so die entsprechenden Pagenummern zu ermitteln (Abb. 5.9). Suchen wir hingegen Punkte aufgrund der Nummer, dann schneiden wir das Directory ebenfalls mit einem Quader, diesmal in seiner Ausformung als dünne Platte. Kombinierte Abfragen ergeben sich als Durchschnitt der beiden Quader. Wir verweisen dazu auf ein ähnliches Konzept, das wir bei der Einbringung thematischer Informationen in Rastermodelle vorgestellt haben (siehe Abschnitt 3.2).

5.3.3. Verallgemeinerung auf Linien- und Flächenobjekte

Alle bisherigen Überlegungen gelten streng nur für punktförmige Objekte. Wenn wir jedoch Linien, Flächen und zusammengesetzte Gebilde speichern wollen, so ergibt sich sofort das Problem, daß viele solche Objekte von den

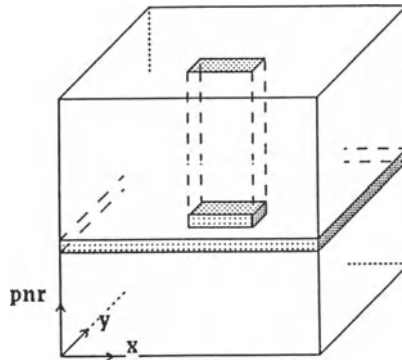


Abb. 5.9. Kombinierte Abfrage nach Lage und Punktnummer

Grenzlinien der Pages geschnitten werden. Es gibt eine Reihe von Möglichkeiten, wie wir dieses Problem in den Griff bekommen:

- a) die Objekte werden längs der Pagegrenzen zerschnitten (Abb. 5.10): diese Vorgehensweise kann dann gewählt werden, wenn die Struktur der Daten einfach ist (z.B. Linien), und wenn der Datenbestand eher statisch ist, wenn sich also die Änderungen der Daten in Grenzen halten. Der ursprüngliche Zusammenhalt muß durch Verweise zwischen den einzelnen Teilen gekennzeichnet werden. Das Zusammenfügen zerschnittener Objekte kann Schwierigkeiten verursachen, besonders wenn es sich um flächige Objekte handelt, in denen viele Inseln auftreten. Wir verweisen dazu auch auf die in Abschnitt 2.2 angestellten Überlegungen hinsichtlich der Zerschneidung flächiger Objekte.

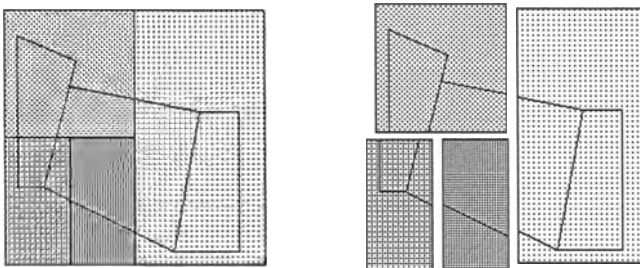


Abb. 5.10. Zerschneiden von Objekten längs der Pagegrenzen

- b) wir speichern Objekte grundsätzlich in jener Page ab, welche ihren Schwerpunkt (oder sonst einen ausgezeichneten Punkt, Zentroid) überdeckt (Abb. 5.11). Somit wird die Speicherung von Objekten auf die Speicherung von Punkten zurückgeführt, und alle bisherigen Betrachtungen können übernommen werden. So wäre etwa gewährleistet, daß Gebäudeumrisse dort abgelegt sind, wo sich der Hauptteil des Gebäudes befindet. Oft wird diese Methode dahingehend ergänzt, daß man in den einzelnen Gebäudepunkten zusätzlich Rückverweise zum jeweiligen Objekt speichert; trotzdem gibt es dann Sonderfälle, die nicht erkannt werden: es sind dies Linien, die eine Page kreuzen, ohne sich dort in einem Punkt zu verknoten.

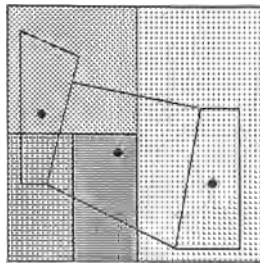


Abb. 5.11. Objekte und ihre Zentroide

- c) wir verwenden **überlappende Pages**; gibt man eine Maximalgröße für ein Objekt vor, so kann man errechnen, welche Überlappung notwendig ist, damit auch das größte dieser Objekte zerschneidungsfrei in einer Page untergebracht werden kann (siehe Frank 1983b). Natürlich wird die Speicherung und der Zugriff bei überlappenden Pages um einige Grade komplizierter.

- d) wir nützen die Baumstruktur aus, die sich bei der Pageteilung ergibt, und verwenden die **Zwischenknoten** des Baumes für die Speicherung großer Objekte. Diese Zwischenknoten hatten bei dem zuvor behandelten einfachen Fall von Punktdaten nur die Funktion des Weiterleitens von Informationen bezüglich der Lage der Nachfolger inne, während die eigentlichen Daten in den Endknoten lagen. Nun geben wir bei jeder Pageteilung nur jene Objekte an die jeweiligen Nachfolger weiter, die von diesen Nachfolgern zerschneidungsfrei beherbergt werden können. Die anderen Objekte verbleiben im Zwischenknoten. In Abb. 5.12 wird die Speicherung dreier unterschiedlich großer Objekte veranschaulicht. Das Objekt B ist so groß, daß es nur in der Wurzel Platz findet. Objekt A kann an die westliche,

Objekt C an die östliche Teilhälfte weitergegeben werden. Kleinere (hier nicht mehr dargestellte) Objekte finden im nordwestlichen Viertel oder in den beiden südwestlichen Teilen Platz.

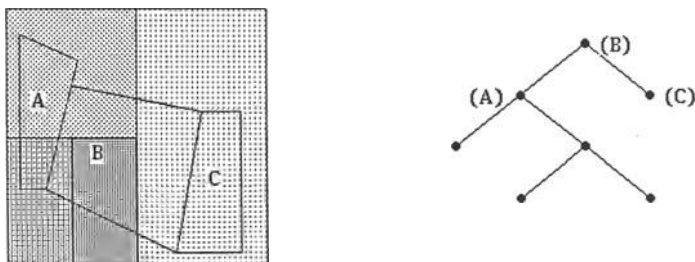


Abb. 5.12. Speichern großer Objekte in Zwischenknoten

Diese Strategie bietet mehrere Vorteile: Sie erlaubt sowohl ein übersichtswises Suchen "großer" Objekte im gesamten Bereich wie auch eine Suche nach Details in einem kleinen Bereich; beides sind häufige Anforderungen in einem raumbezogenen Informationssystem. Im ersten Fall wird der Baum in seiner gesamten Breite, jedoch nur bis zu einer bestimmten Tiefe, durchsucht. Im anderen Fall verfolgt man einen Ast bis zu dessen äußersten Verästelungen. Eine solche Speichermethode unterstützt beispielsweise die Erstellung generalisierter Karten: man zeichnet alle Objekte, die man bis zu einer bestimmten Tiefe im Suchbaum antrifft.

- e) wenn wir für den Zugriff eine der directory-orientierten Methoden verwenden, so können wir im Daten- wie im Adreßraum eine **dritte Dimension** einführen, welche die *Ausdehnung* eines Objektes (etwa den Umkreisradius) wiedergibt. Diese Ausdehnung wird den (zweidimensionalen) Zentroidkoordinaten beigefügt (Abb. 5.13). Wir wollen dies an einem Beispiel erklären, in dem wir die Dimensionen jeweils um 1 verringern; die Daten sind also (eindimensionale) Geradenstücke. In dem Beispiel, das in Abb. 5.13 dargestellt ist, seien mehrere solche Geradenstücke abzuspeichern. Wir führen also eine weitere Dimension ein (in diesem Fall also eine zweite), welche die Ausdehnung (in diesem Fall die halbe Länge) der Objekte darstellt. Je länger ein Geradenstück ist, desto höher oben wird sein Zentroid abgespeichert. Die horizontale Koordinate des Zentroides entspricht dem Mittelpunkt des Geradenstückes, während die vertikale Koordinate dessen halbe Länge ausmacht. Punkte als Sonderfälle von Linien werden ganz unten gespeichert (ihre Vertikalkoordinate ist Null).

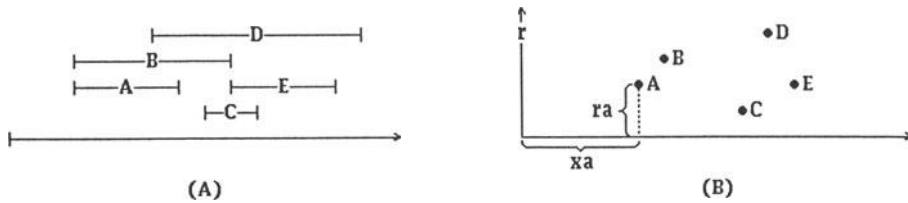


Abb. 5.13. Geradenstücke (A) und ihre Speicherung gemäß Mittelpunkt und Ausdehnung (B)

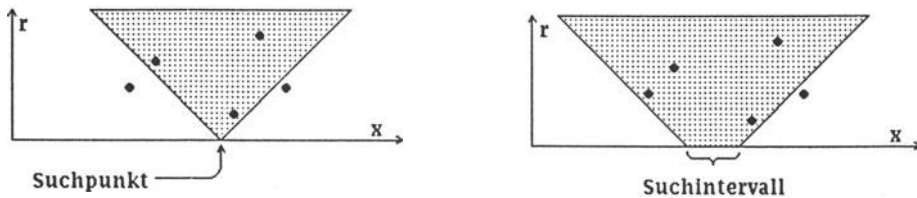


Abb. 5.14. Suchbereiche für Punkte und eindimensionale Objekte

Die so entstehenden Pages werden nun nach einer der zuvor vorgestellten Strategien (EXCELL oder GRID FILE) geteilt, und ein Directory entsteht, das man zur Suche verwenden kann. Wenn wir etwa jene Geradenstücke suchen, die einen bestimmten Punkt überdecken, so entspricht dies der Auswahl jener Zentroidpunkte, deren vertikale Zentroidkoordinate größer ist als der horizontale Abstand zum Suchpunkt: die Entfernung ihres Mittelpunktes vom Suchpunkt ist also geringer als die halbe Ausdehnung, und somit überdecken sie den Suchpunkt. Wir müssen demnach den Adreßraum mit einem auf die Spitze gestellten dreieckigen Suchbereich schneiden, wie er in Abb. 5.14 angedeutet ist. Suchen wir die Überdeckung eines Intervalles, so wird das Dreieck durch ein Trapez ersetzt.

Nach diesem Ausflug in den eindimensionalen Bereich können wir wieder zu unseren zweidimensionalen Anwendungen zurückkehren, in denen die Ausdehnung die dritte Dimension ausmacht. Die Vorgangsweise ist dabei ganz analog; die Suchbereiche sind hier Kegel, die auf die Spitze gestellt sind, und ähnliche Gebilde, die sich ergeben, wenn wir die Kegelspitze längs eines Geradenstückes verschieben oder damit um einen rechteckigen Bereich herumfahren (Abb. 5.15).

Natürlich können wir unsere Strategie noch weiter verallgemeinern, indem wir anstatt des Umkreisradius die

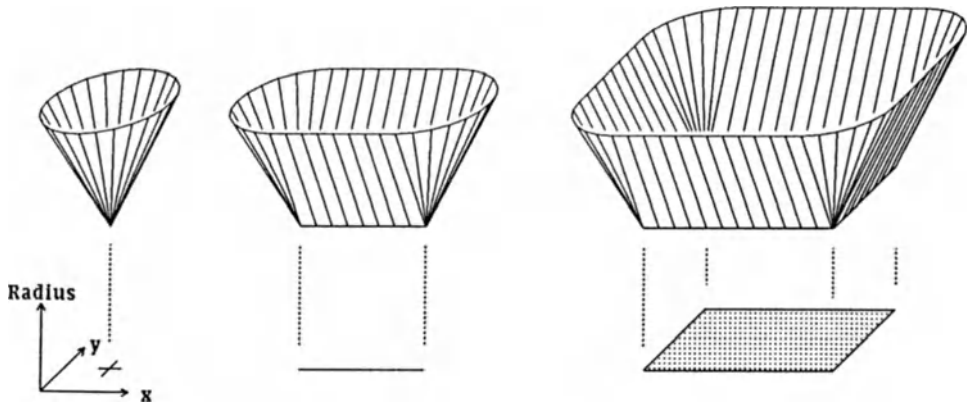


Abb. 5.15. GRIDFILE - Suchbereiche: Beispiele für null-, ein- und zweidimensionale Objekte

Ausdehnung längs der beiden Koordinatenrichtungen einführen; das resultierende Directory wäre dann ein vierdimensionales Array. Weitere Varianten sind zwar vom theoretischen Standpunkt aus interessant, in der Praxis bleibt man jedoch bei maximal 3-4 Dimensionen (Kalcher 1985).

- f) andere Möglichkeiten für die Berücksichtigung der Ausdehnung von Objekten ergeben sich aus der Kombination zweier oder mehrerer Ideen, die wir in den obigen Ausführungen diskutiert haben; so verbindet Frank (1983b) das Konzept überlappender Pages (siehe c) mit dem Gedanken, im Adreßraum sowohl nach der Lage als auch nach der Ausdehnung zu suchen (siehe e). Das Resultat nennt er **Field tree**.

5.4. Speicherung nach topologischen Kriterien

Bisher haben wir den Raumbezug in den Vordergrund gestellt, getreu unserer in den einleitenden Kapiteln getroffenen Feststellung, daß sich alle Daten in einem raumbezogenen Informationssystem letzten Endes auf Koordinaten zurückführen lassen. Die *Topologie* der Daten wurde dabei teilweise *explizit* mitgeführt; so sind etwa im Rahmen einer Objektstruktur "Gebäude" die Verbindungen zwischen den einzelnen Gebäudepunkten *explizit* gespeichert; dies gilt auch für die Reihenfolge, in der die Punkte durchlaufen werden. Andere topologische Eigenschaften, wie etwa die Nachbarschaft, wurden *implizit* mitgeführt: die Koordinaten benachbarter Objekte stimmen in den führenden Ziffern überein, und oft gehören benachbarte Objekte derselben Page an.

Es gibt jedoch gewisse Teilbereiche in einem raumbezogenen Informationssystem, in denen die Topologie eine tragende Rolle spielt. Ein Beispiel dafür ist ein dreiecksvermaschtes Höhenmodell. Dort ist es besonders wichtig, zu jedem Knoten die Nachbarknoten zu finden. Bei der Erstellung eines solchen Höhenmodells verwendet man daher oft eine Datenstruktur, welche in besonderem Maße auf die Topologie raumbezogener Daten eingeht.

Ein Beispiel dafür ist die **DIME-Struktur**, die auf graphentheoretische Zusammenhänge aufbaut und sich gut für flächendeckende Bereichsaufteilungen eignet (siehe Abschnitt 2.3.6 und Abb. 2.16). Eine Dreiecksvermaschung stellt eine solche Aufteilung dar. Der **duale Graph** einer DIME-strukturierten Dreiecksvermaschung (**Thiessen-Polygon** oder **Voronoi-Diagramm**) kann zum Übergang von einer Dreiecksfläche zu ihren Nachbarflächen verwendet werden.

Zum Abschluß verweisen wir noch auf den in Abschnitt 5.2 eingeführten **LAG** (Graph der Linienadjazenzen), welcher es (ähnlich wie der duale Graph einer DIME-Struktur) erlaubt, topologische Kriterien stärker zu betonen.

5.5. Datenbanken und lokale raumbezogene Methoden

In den vorangegangenen Abschnitten haben wir lokale Speicherungs- und Zugriffsmethoden kennengelernt, die dem Raumbezug der Daten in besonderem Maße entgegenkommen. Wenn wir sie verwenden wollen, so müssen wir allerdings einen wichtigen Punkt beachten, nämlich die Konsistenz dieser lokalen Methoden mit dem Langzeitspeicher, also mit der raumbezogenen Datenbank (siehe Kapitel 4). Wir müssen zwei konkurrierende Vorgaben miteinander in Einklang bringen: einerseits die Funktion einer Datenbank, welche die Konsistenz großer Datenbestände über lange Zeiträume hinweg zu garantieren hat, und die eine *Vielzahl* unterschiedlicher - und oft auch gleichzeitig vorgetragener - Anwenderwünsche zu erfüllen hat; und andererseits die Gewährleistung eines zufriedenstellenden Zeitverhaltens für ein graphisch-interaktiv ablaufendes Anwenderprogramm. Mit den zur Zeit verfügbaren Mitteln an Hardware und Datenbank-Software ist es nicht möglich, *beide* Vorgaben zufriedenstellend zu erfüllen.

Wie wir bereits in Kapitel 4 festgestellt haben, können thematische Beziehungen sehr gut in eine Datenbank abgebildet werden. Detaillierte und maßgeschneiderte Abfragen über thematische Verflechtungen sind mit Elementen der relationalen Algebra erstaunlich einfach zu formulieren (siehe Abschnitt 4.9). Wir kommen also zu dem Schluß, daß wir die thematischen Aspekte raumbezogener Informationen am

besten in der Obhut der Datenbank lassen, während wir uns für die geometrisch-topologischen Aspekte die Ideen zunutze machen können, die in den vorangegangenen Abschnitten erläutert wurden. Wir wollen einige denkbare Varianten durchleuchten:

5.5.1. Variante A: die ganzheitliche Lösung

Die Datenbank behält die Kontrolle über alle Daten, also sowohl thematische wie auch geometrisch-topologische Daten. Für den raumbezogenen Zugriff wird ein zusätzlicher Index angelegt, der sich an einer der Methoden orientiert, die wir in den einleitenden Abschnitten behandelt haben. Dies bringt entscheidende Vorteile: Redundanzen (Mehrfachspeicherungen) werden vermieden; der Anwender kann jederzeit sämtliche Vorteile eines relationalen Datenbankmodells nützen; insbesondere kann er die Struktur der Daten weitgehend selbst festlegen, und er kann die mächtigen Werkzeuge der relationalen Abfragesprache einsetzen.

Geometrisch-topologische Strukturen werden größtenteils in **Bulks** abgelegt; für die relationale Datenbank (RDB) ist ein Bulk eine unstrukturierte Zeichenkette, die sie unverändert an das Applikationsprogramm weitergibt; dies ist allein in der Lage, den Bulk zu entschlüsseln. Ein Beispiel für einen solchen Bulk stellt der Verlauf eines Flusses dar (Abb. 5.16): nur jene Punkte, an denen andere Flüsse einmünden, sind für die RDB als Knotenpunkte erkennbar, während die Zwischenpunkte im Bulk gespeichert sind. (In Kapitel 2 haben wir den Begriff der Kanten-Knoten-Struktur eingeführt. Jede Kante würde also als Bulk abgespeichert werden.)

All dies setzt voraus, daß die RDB Möglichkeiten besitzt, für Bulks eine variable Länge vorzusehen, um die Verschwendung an Speicherplatz hintanzuhalten. Außerdem muß die Abbildung der Reihenfolge in einer zufriedenstellenden Weise gelöst sein; die Reihenfolge der Punkte innerhalb einer Kante ist natürlich wesentlich. Schließlich ist es noch wünschenswert, daß die variable Länge auch für strukturierte Datenbankinhalte ermöglicht wird; dadurch kann man verhindern, daß für jede 1:M-Beziehung eine eigene Tabelle angelegt wird. Auch eine Berücksichtigung von mehreren Versionen ein- und desselben logischen Objektes ist vorteilhaft, wenn wir etwa an das Nebeneinander von geometrisch exakten und rechtlich verbindlichen Punktkoordinaten denken (Petry 1987). All diese eben erwähnten Eigenschaften gehören nicht zum Standard einer RDB, sondern sind als *Erweiterung* zu sehen.

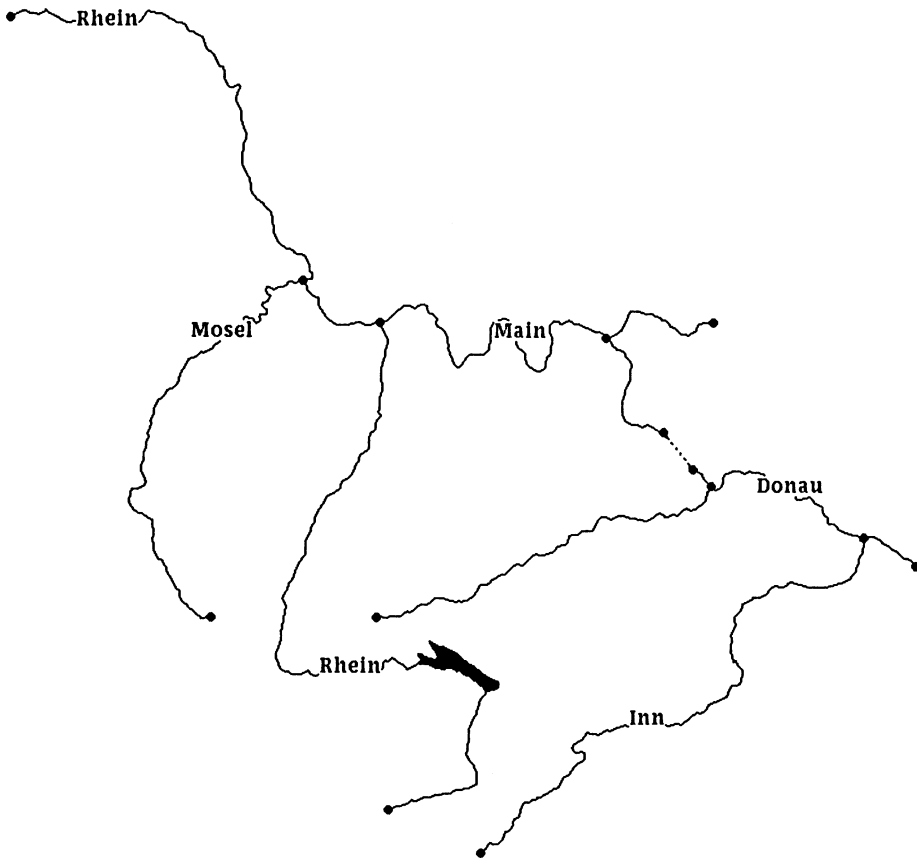


Abb. 5.16. Knoten eines Gewässernetzes und dazwischenliegende Kantenbulks

Der schnelle Zugriff über den Raumbezug kann durch das Einfügen einer Zwischenschicht ermöglicht werden, welche aufgrund eines raumbezogenen Index (Baumstruktur, Gridfile, Excell-Methode und ähnliches) physisch benachbarte Objekte in einen Zwischenpuffer holt (Abb. 5.17). Dieser Zwischenpuffer erfüllt die Funktion einer Page, wie wir sie in den vorangegangenen Abschnitten erklärt haben. Man geht davon aus, daß sich der Inhalt des Zwischenpuffers im interaktiven Betrieb nur langsam ändert, weil man meist lokal arbeitet und mehrmals auf dieselben Objekte zugreift. Ein Beispiel für eine solche Vorgangsweise bietet das Datenbankkonzept des Systems WILD S9 (Charlwood et al. 1987).

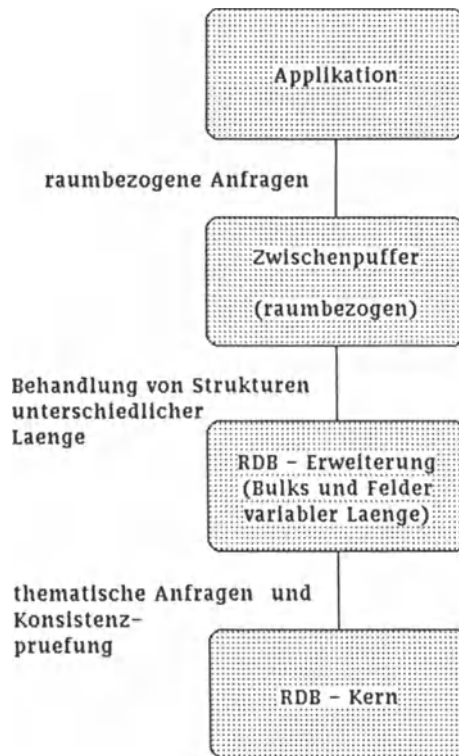


Abb. 5.17. Variante A: Schema für den raumbezogenen Zugriff

5.5.2. Variante B: Trennung von Geometrie und Thematik

Diese Variante greift die einleitenden Bemerkungen hinsichtlich der jeweiligen Vorzüge für thematische bzw. geometrisch-topologische Sachverhalte auf. Sie beläßt die Thematik in einer relationalen Datenbank (RDB) und legt dazu parallel raumbezogene Datenstrukturen für die Geometrie und Topologie an (Abb. 5.18). Der Vorteil einer solchen Vorgangsweise ist klar ersichtlich: man verwendet für beide Bereiche die jeweils optimale Datenstruktur. Außerdem ist es möglich, die Geometrie und Topologie jeweils als gemeinsamen Kern des Informationssystems anzusehen, dem verschiedene Themen (und damit verschiedene benützerspezifische Datenbanken) überlagert werden können. Zwischen den beiden Teilen des Informationssystems bestehen Querverbindungen über räumliche Identifikatoren und thematische Schlüssel. Ein solches Datenbankkonzept ist etwa im System ARC-INFO realisiert.

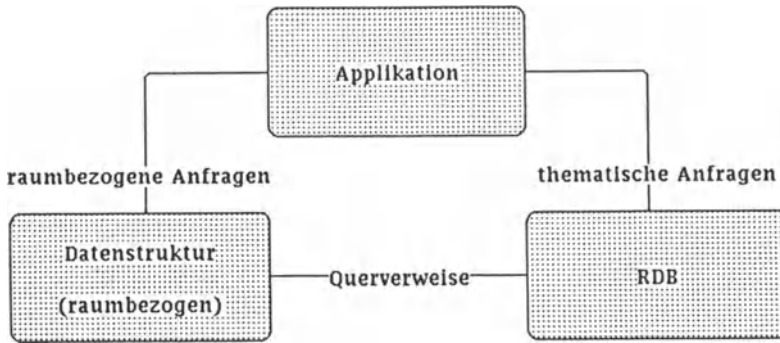


Abb. 5.18. Variante B: raumbezogener und thematischer Zugriff

Der Nachteil ist ebenso deutlich: geometrische und topologische Aspekte kommen nie in den Genuß einer strengen Konsistenzkontrolle, wie sie eine RDB durchführen kann. Ein solches Konzept ist daher dann angebracht, wenn geometrische Aspekte (wie etwa die Punktgenauigkeit) hinter thematischen Anforderungen zurücktreten: ein großräumiges Geoinformationssystem, das Lagerstätten, Waldschadensgebiete, meteorologische Modelle und ähnliches abbildet, kann auf eine penible Kontrolle der Redundanzfreiheit von Punktkoordinaten verzichten, ein Grundstückskataster allerdings nicht.

5.5.3. Variante C: Raumbezogene Transaktionen

Wir übernehmen aus der Variante A die Vorteile einer einheitlichen Langzeitspeicherung und aus der Variante B den Gedanken, daß thematische Aspekte einerseits und geometrisch-topologische Aspekte andererseits eine jeweils unterschiedliche Behandlung rechtfertigen. Die Agenden der Langzeitspeicherung werden also hier für beide Bereiche von der RDB übernommen, wobei wieder Ergänzungen hinsichtlich der unterschiedlich langen Felder und Bulks stillschweigend vorausgesetzt werden. Beginnt nun ein Anwender eine Sitzung, so werden jene Teile der Datenbank, die für ihn interessant sind, *selektiert*. Er definiert auf diese Weise ein **Projekt** und dessen thematischen und geometrisch-topologischen Datenbereich. (Wir kommen in Kapitel 7 noch näher auf die Projektbildung, die Selektionskriterien und die Zugriffsberechtigung im Multi-User-Betrieb zu sprechen.)

Nach der Selektion werden die entsprechenden Geometriedaten auf eine lokale Datenstruktur abgebildet, welche den raumbezogenen Zugriff unterstützt. Für die thematischen Daten hingegen wird innerhalb der RDB ein View (also eine logische Zusammenfassung von Daten) angelegt. Danach arbeitet der Anwender weitgehend unabhängig von den eigentlichen Daten der RDB. Geometrische und topologische Operationen können lokal erledigt werden. Für thematische Operationen können alle Möglichkeiten einer RDB ausgeschöpft werden. Trotzdem hat der Anwender den Eindruck, daß auch seine thematischen Daten lokal gehalten werden. Das Datenbankverwaltungssystem führt über alle seine thematischen Aktionen ein Protokoll. Veränderungen der Geometrie müssen ebenfalls protokolliert werden; dies geschieht natürlich ohne Unterstützung der RDB.

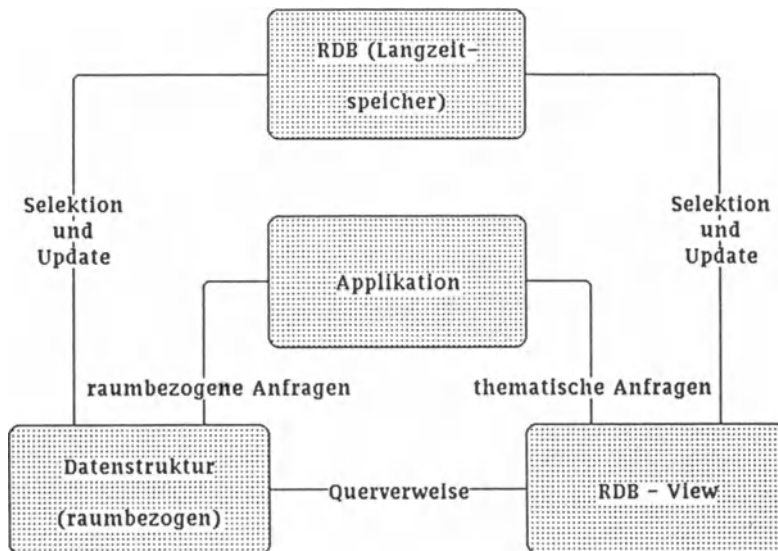


Abb. 5.19. Variante C: Transaktionskonzept

Nach Abschluß einer solchen Projektsitzung findet dann ein *Update* der Daten statt. Aufgrund der beiden Protokolle werden sowohl die Thematik wie auch die Geometrie im Langzeitspeicher auf den neuesten Stand gebracht. Alle Veränderungen an den Daten, die zwischen einer solchen Selektion und dem Update registriert werden, bezeichnet man in ihrer Gesamtheit als **Transaktion** (Abb. 5.19). Für die RDB als Langzeitspeicher gilt die gesamte Transaktion als *eine Update-Operation*, die entweder als Ganzes erfolgreich durchgeführt wird, oder - beim Auftreten von etwaigen Konsistenzproblemen - ebenso pauschal abgelehnt wird. Dabei

können die Fähigkeiten des roll forward und des roll back einer RDB ausgenutzt werden (siehe Kapitel 4). Für die Vorgangsweise bei der Konfliktlösung, speziell im Netzwerk- und Multi-User-Betrieb, verweisen wir auf Kapitel 7. Wir wollen noch anmerken, daß die Transaktion in vielen Fällen gar kein Update beinhaltet; so etwa bei der Aufbereitung eines Planes, wo man Korrekturen nur zur Erzeugung eines optisch ansprechenden Bildes macht, und diese Korrekturen nach der Erstellung der Zeichnung wieder vergißt. Als Beispiel für die Variante C sei die Datenbanklösung von KERN-INFOCAM genannt (Bartelme u. Späni 1988).

Das Transaktionskonzept der Variante C stellt für viele Anwendungen ein vernünftiges Gleichgewicht zwischen den - oftmals erwähnten - einander widersprechenden Aufgabenstellungen dar. Sein Erfolg hängt allerdings eng mit einer solchen Lösung des Selektions- und Updateprozesses zusammen, die den Anwender zeitlich nicht - oder nur wenig - belastet; so ist es denkbar, daß diese Prozesse im Stapelbetrieb in betriebsschwachen Zeiten abgewickelt werden. Im übrigen entspricht dieses Konzept weitgehend einem betrieblichen Organisationsmodell, in dem große Projekte in einzelne Teilprobleme untergliedert werden, die dann dezentral ("vor Ort") gelöst werden; gelegentlich findet dann eine Angleichung an das globale Modell statt.

5.5.4. Andere Varianten

Die bisher vorgestellten Varianten können natürlich beliebig abgeändert und miteinander vermischt werden. Es ist auch denkbar, auf die RDB gänzlich zu verzichten und sich auf eine raumbezogene Datenstruktur zu beschränken. In diesem Zusammenhang müssen wir wieder an Rasterdaten erinnern, die ja in diesem Abschnitt bis jetzt keine Rolle gespielt haben. Solche Rasterdaten (Photos, Satellitenbilder und ähnliches) müssen ohnehin immer in baum- oder zeilenorientierte Strukturen abgebildet werden, so wie wir sie in Abschnitt 5.2 erläutert haben. Diese Strukturen weisen implizit einen Raumbezug auf. Oft wird für jedes einzelne Bild eine solche Rasterstruktur angelegt; es ist aber auch denkbar, daß mehrere Bilder, die dasselbe Gebiet überdecken, gemeinsam in einer Struktur Platz finden. Für die Langzeitspeicherung dieser Daten werden hauptsächlich Magnetbänder verwendet. Eine *Transaktion* im Sinne von Variante C beginnt dann beim Einspannen und Lesen eines Magnetbandes und endet beim Beschreiben und Auslagern. Zwischen diesen beiden Zeitpunkten sind die Daten lokal auf den eben erwähnten Rasterstrukturen vorhanden. Für die *Verwaltung* der Magnetbänder und der jeweils zugeordneten Informationen ist allerdings wieder der Einsatz einer Datenbank zu empfehlen.



KAPITEL 6
WERKZEUGE

6.1. Übersicht

In den Kapiteln 2 und 3 haben wir uns ausführlich mit den geometrischen, topologischen und thematischen Aspekten raumbezogener Daten befaßt. Ein raumbezogenes Informationssystem ist jedoch mehr als eine Datenbank. Es beinhaltet neben den Daten auch eine Palette von "Werkzeugen" zur Bearbeitung dieser Daten (Dangermond 1986). Diese **Software-Werkzeuge** sind in unserem Fall Algorithmen, die dem Anwender zur Verfügung gestellt werden, um ihn bei der Lösung von Standardproblemen zu unterstützen. In den folgenden Abschnitten wollen wir uns näher mit einzelnen Bearbeitungsschritten und den dabei benötigten Algorithmen auseinandersetzen. Sie haben eines gemeinsam: sie verändern die Geometrie und fallweise auch die Topologie der Daten; sie werden aber alle von thematischen Überlegungen angetrieben und gesteuert. Dies entspricht dem Ergebnis, zu dem wir in diesen früheren Kapiteln gekommen sind: die Geometrie und die Topologie raumbezogener Daten wird wesentlich durch die darüber liegende Thematik beeinflusst.

Wir kommen in Abschnitt 6.2 auf Transformationen zu sprechen, also Koordinatentransformationen in der Ebene ebenso wie Transformationen, die einer räumlichen Darstellung dienen. In Abschnitt 6.3 werden wir Schnittaufgaben behandeln; dabei geht es um die Beantwortung der Frage, wie zwei topologische Elemente (Punkt, Linie oder Fläche) zueinander liegen: ob ein Punkt auf einer Linie bzw. innerhalb einer Fläche liegt, ob eine Linie eine Fläche schneidet, ob eine Fläche ein Teil einer anderen Fläche ist usw. Komplexe Probleme wie etwa die Berechnung von Pufferzonen oder auch die Platzierung und Freistellung von kartographischen Symbolen und Texten können auf diese elementaren Vergleiche zurückgeführt werden.

Bei all diesen Betrachtungen setzten wir bisher ein Vektormodell voraus; in Abschnitt 6.4 geht es um die Werkzeuge, die uns für die Beantwortung derselben Fragen in einem Rastermodell zur Verfügung stehen.

Fragen der Interpolation bzw. der Approximation von Kurven und Flächen haben einen großen Stellenwert, nicht nur, was die Darstellung betrifft, sondern auch in bezug auf eine effiziente Modellabbildung der Topologie, auf eine Filterung von unnötigen Informationen während der Erfassung und auf die Unterstützung lagemäßiger Auswertungen. Besondere Bedeutung kommt dabei der Erstellung eines digitalen Geländemodells zu. Die Abschnitte 6.5 und 6.6 sind diesen Problemen gewidmet.

In Abschnitt 6.7 besprechen wir Fragen der Generalisierung. Der Übergang von großen zu kleineren Maßstäben neben einer möglichst weitgehenden Beibehaltung des Informationsgehaltes

stellt eines der schwierigsten Probleme der automationsgestützten Kartographie dar. Für einzelne eingeschränkte Bereiche existieren jedoch brauchbare Lösungen.

Damit ist die Beschreibung der Werkzeuge eines RIS noch keineswegs umfassend. Es würde den Rahmen dieses Buches sprengen, wenn wir das gesamte Instrumentarium darstellen wollten, das im Rahmen eines RIS denkbar und wünschenswert ist. Wir können uns nur darauf beschränken, einige weitere Werkzeuge in Abschnitt 6.8 beispielhaft anzudeuten.

6.2. Geometrische Transformationen

6.2.1. Drehung, Skalierung und Verschiebung

Unter den Begriff "geometrische Transformation" fallen die Drehung, die Verschiebung und die maßstäbliche Vergrößerung bzw. Verkleinerung in der Ebene und im Raum. Für räumliche Transformationen kommt noch die Perspektive und ebenso die Abbildung eines dreidimensionalen Objektes in die Ebene hinzu. In diesem Abschnitt setzen wir für die Topologie ein Vektormodell voraus. Transformationen in Rastermodellen werden in Abschnitt 6.4 behandelt.

Die Drehung eines Vektors im dreidimensionalen Raum kann durch die Multiplikation mit einer Drehmatrix ausgedrückt werden. Handelt es sich speziell um eine Drehung um die z-Achse, so sieht die Transformationsgleichung folgendermaßen aus:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \underbrace{\begin{bmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R(a)} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6-2-1)$$

Eine allgemeine Drehung im dreidimensionalen Raum kann aus drei Drehungen um die einzelnen Koordinatenachsen zusammengesetzt werden; sie läßt sich also durch drei Drehwinkel beschreiben, die in die resultierende Drehmatrix eingehen:

$$XR = R(a) * R(b) * R(c) * X \quad (6-2-2)$$

wobei

$$R(a) * R(b) * R(c) = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \quad (6-2-3)$$

Die Elemente dieser Drehmatrix sind aus Summen und Produkten von Sinus und Cosinus der einzelnen Drehwinkel aufgebaut. Auch eine **Maßstabstransformation** können wir durch eine Multiplikation mit einer Matrix darstellen, die eine Diagonalgestalt aufweist. Die einzelnen Diagonalelemente entsprechen den Streckungsfaktoren längs der jeweiligen Koordinatenachsen.

$$XRS = S * XR \quad \text{mit} \quad S = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & sz \end{bmatrix} \quad (6-2-4)$$

Die **Verschiebung** um einen bestimmten Betrag in x-, y- und z-Richtung erfolgt durch die Addition des Verschiebungsvektors:

$$XRST = XRS + XT \quad \text{mit} \quad XT = \begin{bmatrix} xt \\ yt \\ zt \end{bmatrix} \quad (6-2-5)$$

Wir sehen also, daß eine allgemeine Transformation, welche aus einer Drehung, einer maßstäblichen Verzerrung und einer Verschiebung zusammengesetzt ist, durch die Multiplikation mit einer Matrix und die nachträgliche Addition des Verschiebungsvektors zustande kommt:

$$XRST = S * R * X + XT \quad (6-2-6)$$

Die Matrix ist dabei aus Rotations- und Maßstabsbestandteilen aufgebaut. Insgesamt sind bei dieser Transformation 9 Parameter im Spiel: je 3 für die Verdrehung, den Maßstab und die Verschiebung. Die Bauart dieser Formel bleibt gleich, auch wenn wir die Reihenfolge der einzelnen Operationen vertauschen; die Elemente der Matrizen und des Verschiebungsvektors ändern sich natürlich.

6.2.2. Homogene Koordinaten

Die Transformation läßt sich noch einfacher formulieren, wenn wir von dem uns geläufigen kartesischen Koordinatensystem auf ein System von *homogenen Koordinaten* übergehen: wir können jeden Punkt (x,y,z) auch durch seine homogenen Koordinaten darstellen, wobei es sich in diesem Fall um ein Quadrupel (r,s,t,u) handelt, dessen einzelne Elemente nur bis auf einen gemeinsamen Faktor bestimmt sind. So könnten wir etwa von (x,y,z) auf das Quadrupel $(x,y,z,1)$ übergehen; genauso denkbar wäre aber eine Darstellung $(w*x,w*y,w*z,w)$ mit einem beliebigen Wert für w . Umgekehrt erhalten wir aus einer homogenen Punktdarstellung (r,s,t,u) die kartesischen Koordinaten, indem wir jeweils die erste, zweite und dritte Komponente durch die vierte Komponente dividieren:

$$\begin{aligned}x &= r/u \\y &= s/u \\z &= t/u\end{aligned}\quad (6-2-7)$$

Die Verdrehung und die Maßstabsverzerrung lassen sich in einfacher Weise auf homogene Koordinaten übertragen, indem wir die 3 x 3-Matrix zu einer 4 x 4-Matrix erweitern:

$$R(a, b, c) = \begin{bmatrix} A & D & G & 0 \\ B & E & H & 0 \\ C & F & I & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-2-8)$$

$$S(sx, sy, sz) = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-2-9)$$

Nun läßt sich aber auch die Verschiebung durch eine Matrizenmultiplikation darstellen:

$$T(xt, yt, zt) = \begin{bmatrix} 1 & 0 & 0 & xt \\ 0 & 1 & 0 & yt \\ 0 & 0 & 1 & zt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6-2-10)$$

Dies können wir leicht nachprüfen:

$$T * (x, y, z, 1) = (x+xt, y+yt, z+zt, 1) \quad (6-2-11)$$

Somit können wir alle Transformationselemente in einheitlicher Weise durch eine Multiplikation mit einer 4 x 4-Matrix darstellen. Natürlich können wir auch homogene Koordinaten in der Ebene einführen und überhaupt alle unsere bisherigen Überlegungen auf die Ebene einschränken: wir gehen von einem kartesischen Koordinatenpaar (x,y) auf ein Tripel (x,y,1) über; die Verdrehung, die jetzt natürlich nur mehr einen Parameter d aufweist, die Skalierung (mit 2 Parametern sx, sy) und die Verschiebung (ebenfalls mit zwei Parametern xt, yt) erreichen wir durch die Multiplikation mit einer Matrix:

$$\begin{bmatrix} sx \cos d & -sx \sin d & xt \\ sy \sin d & sy \cos d & yt \\ 0 & 0 & 1 \end{bmatrix} \quad (6-2-12)$$

Homogene Koordinaten bieten aber auch noch andere Vorteile; in Abschnitt 6.3 verwenden wir den Umstand, daß Punkte (ausgedrückt durch ihre drei homogenen Koordinaten wx,wy,w) und Gerade (ausgedrückt durch die drei Koeffizienten a,b,c der impliziten Form ax + by + c = 0) dual sind: Viele Sachverhalte, die für eine Kategorie zutreffen, haben eine duale Entsprechung für die andere Kategorie. Homogene

Koordinaten bieten aber auch die Möglichkeit, die "Unendlichkeit in den Griff zu bekommen". Wird die letzte Komponente zu Null, so ist die homogene Darstellung nach wie vor definiert. Homogene Koordinaten sind daher besonders im Rahmen der *projektiven Geometrie* nützlich. Dort ist eine weitere Eigenschaft homogener Koordinaten vorteilhaft: die perspektivische Transformation wird linear und ist durch eine Matrizenmultiplikation darstellbar; wir wollen dies im folgenden näher begründen.

6.2.3. Perspektive

Nehmen wir an, daß das Auge des Betrachters auf der z-Achse liegt, und daß das Bild auf eine Ebene normal zur z-Achse projiziert wird (dies können wir durch eine geeignete Wahl des Koordinatensystems immer erreichen).

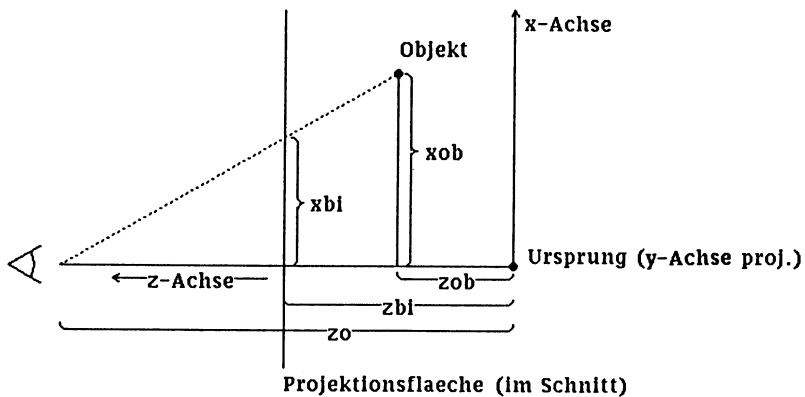


Abb. 6.1. Perspektivische Abbildung

Es ergeben sich folgende Verhältnisse (siehe Abb. 6.1):

$$x_{bi} : x_{ob} = (z_o - z_{bi}) : (z_o - z_{ob}) \quad (6-2-13)$$

oder

$$x_{bi} = x_{ob} * (z_o - z_{bi}) / (z_o - z_{ob}) \quad (6-2-14)$$

analog

$$y_{bi} = y_{ob} * (z_o - z_{bi}) / (z_o - z_{ob}) \quad (6-2-15)$$

Nehmen wir ferner an, daß die Projektionsfläche durch den Ursprung geht; es wird also $z_{bi} = 0$.

$$x_{bi} = x_{ob} * (1 - z_{ob} / z_o)^{-1} \quad (6-2-16)$$

$$y_{bi} = y_{ob} * (1 - z_{ob} / z_o)^{-1} \quad (6-2-17)$$

Bei Verwendung von homogenen Koordinaten lässt sich demnach die perspektivische Abbildung durch die Multiplikation mit einer Matrix P darstellen, die von einem Parameter z_o abhängt, der die Entfernung des Betrachters vom Ursprung angibt:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/z_o & 1 \end{bmatrix}}_P * \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ 0 \\ w(1-z/z_o) \end{bmatrix} \quad (6-2-18)$$

Nun können wir tatsächlich eine allgemeine Transformationsmatrix aufstellen, welche alle Aspekte, also Verdrehung, Maßstab, Verschiebung und Perspektive berücksichtigt; sie entsteht durch Multiplikation der Matrix P mit den vorhin abgeleiteten Matrizen und hängt von 10 Parametern ab. Diese Matrix wird also für jede Abbildung einmal berechnet und dann auf alle zu transformierenden Elemente angewendet. Ein Spezialfall ergibt sich, wenn wir das Resultat der Transformation in die Ebene - etwa in die Bildebene - abbilden: dann wird die dritte Zeile dieser Matrix zu Null:

$$M = P * T * S * R =$$

$$\begin{bmatrix} A \, s_x & D \, s_x & G \, s_x & x_t \\ B \, s_y & E \, s_y & H \, s_y & y_t \\ 0 & 0 & 0 & 0 \\ -C \, s_z/z_o & -F \, s_z/z_o & -I \, s_z/z_o & 1-z_t/z_o \end{bmatrix} \quad (6-2-19)$$

Auch lassen sich sämtliche vorhin beschriebenen Teiltransformationen durch entsprechende Vereinfachungen in dieser Matrix reproduzieren: wird etwa z_o unendlich groß, so wird P zur Einheitsmatrix; alle Sehstrahlen verlaufen parallel, und es ergibt sich wieder der frühere Fall.

Mit Hilfe dieser einfachen Formel können wir perspektivische Darstellungen von Geländemodellen erzeugen. Wir setzen die 10 Parameter der Matrix M fest und wenden dann diesen Matrizenoperator auf jeden einzelnen Gitterpunkt des Interpolationsrasters an. Das Höhenmodell in Abb. 6.21 ist ein Ergebnis einer solchen perspektivischen Transformation.

6.3. Schnittaufgaben

6.3.1. Relative Lage von Punkten und Kanten

Die Ermittlung topologischer Sachverhalte stellt eine der wichtigsten Hilfestellungen dar, die ein raumbezogenes Informationssystem anbieten muß. Es geht dabei um die Bestimmung der relativen Lage zweier topologischer Elemente zueinander, ob dies nun Punkte, Kanten oder Flächen seien. Vektorstrukturen stehen im Vordergrund der Überlegungen in diesem Abschnitt, denn bei Rasterstrukturen ist die Beantwortung topologischer Sachverhalte (fast) trivial; wir verweisen dazu auf Abschnitt 6.4. Wir stellen uns also folgende Fragen: Liegt ein Punkt auf einer Kante? Liegt er innerhalb oder außerhalb eines vorgegebenen Polygons? Schneiden zwei Polygone einander, berühren sie sich nur, oder haben sie keine gemeinsamen Punkte? Wir können eine Vielzahl von weiteren Fragen formulieren; dabei werden wir feststellen, daß alle diese Fragen auf ein Grundproblem zurückgeführt werden können, nämlich auf die Lage eines Punktes bezüglich eines Geradenstückes.

Dabei verwenden wir wieder *homogene Punktkoordinaten*: wir haben sie in Abschnitt 6.2 eingeführt, weil sie besonders einfache Transformationsformeln ermöglichen; auch hier ist dies der Fall, denn ein Punkt $P(x, y, w)$ liegt genau dann auf einem Geradenstück, das von den Punkten P_1 und P_2 begrenzt wird, wenn gilt:

$$\text{Det} \begin{bmatrix} x & x_1 & x_2 \\ y & y_1 & y_2 \\ w & w_1 & w_2 \end{bmatrix} = 0 \quad (6-3-1)$$

oder

$$x * (y_1 w_2 - w_1 y_2) + y * (w_1 x_2 - x_1 w_2) + w * (x_1 y_2 - y_1 x_2) = 0 \quad (6-3-2)$$

Daraus ergibt sich die konventionelle Darstellung einer Geraden, wenn wir - wie üblich - für die w -Komponenten den Wert 1 wählen. Die drei Punkte P , P_1 und P_2 sind also unter diesen Umständen kollinear. Wenn wir der Kante eine Richtung zuweisen, indem wir P_1 als Startpunkt und P_2 als Endpunkt deklarieren, so ist die Determinante (P, P_1, P_2) kleiner als Null, wenn der Punkt P *rechts* von der Kante liegt; sie ist größer als Null, wenn er *links* von der Kante liegt. Diese Determinante bietet somit ein einfaches Mittel zur Überprüfung der topologischen Beziehung von Punkten und Kanten.

Nun halten wir uns die Geradengleichung $ax + by + c = 0$ vor Augen und untersuchen die Lage der Geraden $G(a, b, c)$ bezüglich zweier Geraden G_1 und G_2 . Wenn die drei Geraden

durch einen gemeinsamen Schnittpunkt gehen sollen, so muß G , ausgedrückt durch seine Koeffizienten (a, b, c) , linear abhängig von den anderen beiden Koeffiziententripeln sein:

$$\text{Det} \begin{bmatrix} a & a_1 & a_2 \\ b & b_1 & b_2 \\ c & c_1 & c_2 \end{bmatrix} = 0 \quad (6-3-3)$$

oder

$$a * (b_1 c_2 - c_1 b_2) + b * (c_1 a_2 - a_1 c_2) + c * (a_1 b_2 - b_1 a_2) = 0 \quad (6-3-4)$$

Daraus folgern wir, daß die drei Klammerausdrücke genau die (homogenen) Koordinaten des Schnittpunktes sind. Es ergeben sich demnach interessanterweise duale Entsprechungen von Punkten und Geradenstücken. Diese Dualität wird in einer Reihe von Algorithmen ausgenutzt. Siehe dazu auch Pavlidis (1982) und Preparata u. Shamos (1985).

6.3.3. Verschneidung mit Polygonen

Die oben angestellten elementaren Überlegungen können wir nun sukzessive verallgemeinern. Zunächst interessiert uns die Lage zweier Kantenstücke. Da es sich um *endliche* Kantenstücke handelt, gibt es nicht nur die beiden Alternativen "Schnittpunkt oder Parallelität"; vielmehr sind die in Abb. 6.2 dargestellten Variationen denkbar, die wiederum durch entsprechende Kombinationen der obigen Determinantenbedingung gekennzeichnet sind:

$$\begin{array}{ll} S_1 = \text{Det} (P_1, P_3, P_4) & S_3 = \text{Det} (P_3, P_1, P_2) \\ S_2 = \text{Det} (P_2, P_3, P_4) & S_4 = \text{Det} (P_4, P_1, P_2) \end{array} \quad (6-3-5)$$

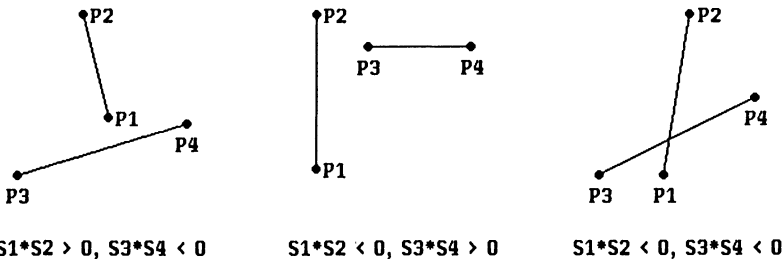


Abb. 6.2. Beispiele für die Lage zweier Geradenstücke

Aus Gründen der Einfachheit haben wir jene Fälle außer acht gelassen, die sich ergeben, wenn sich Kantenstücke nur berühren; das bedeutet, daß eine oder mehrere Determinanten zu Null werden. Diese Fälle sind gar nicht so selten, denn unsere Koordinaten sind mit (endlicher) Computergenauigkeit gespeichert. Da unsere Überlegungen jedoch nur grundsätzlicher Art sind, klammern wir die entsprechenden Verzweigungen im Algorithmus aus.

6.3.3. Verschneidung mit Polygonen

Als nächstes betrachten wir die Lage eines Punktes bezüglich eines **konvexen** Polygons. Durchschreiten wir die Kanten des Polygons in Uhrzeigerrichtung, so liegt ein Punkt im Inneren des Polygons, wenn er jeweils rechts zu liegen kommt. Im Beispiel aus Abb. 6.3 sind also für einen konkreten Punkt (x,y) die nebenstehenden Ungleichungen zu erfüllen, die sich aus den Determinantenbedingungen und nach der Überführung in inhomogene Koordinaten ergeben.

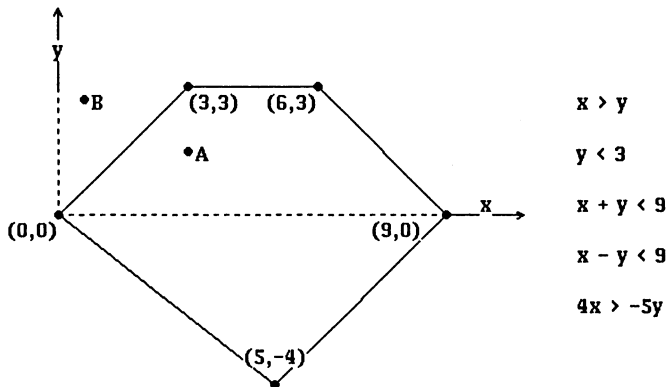


Abb. 6.3. Beispiele für die Lage von Punkten bezüglich eines Polygons

Der Schnitt einer Kante mit einem Polygon ist das nächste Ziel unserer Überlegungen. Wieder ist dabei zu beachten, daß unsere Kantenstücke endlich sind; es kann also durchaus sein, daß beide Endpunkte der Kante innerhalb des Polygons liegen; ist das Polygon konvex, so ergibt sich in diesem Fall natürlich kein Schnittpunkt. In den anderen Fällen müssen wir jede einzelne Polygonseite mit der vorgegebenen Kante vergleichen, so wie wir dies oben getan haben. Dabei können wir wieder den Umstand ausnützen, daß es höchstens zwei Schnittpunkte gibt; den Sonderfall, daß lediglich eine Berührung stattfindet, klammern wir abermals aus.

Der - schon recht allgemeine - Fall der Verschneidung zweier konvexer Polygone muß durch einen paarweisen Vergleich sämtlicher Seiten der beiden Polygone gelöst werden. Es gibt allerdings Möglichkeiten zur Beschleunigung des Verfahrens: es ist beispielsweise günstig, wenn wir zunächst einmal nur die umschreibenden Rechtecke der beiden Polygone vergleichen, denn für Rechtecke vereinfacht sich die Überprüfung beträchtlich; schneiden sich die Rechtecke nicht, so erübrigen sich weitere Untersuchungen; ansonsten genügt es, jene Seiten miteinander zu vergleichen, die im Durchschnitt der beiden Rechtecke liegen (siehe auch Abb. 2.28). Eine weitere Beschleunigung läßt sich erzielen, wenn eines der beiden Polygone fest bleibt, während das zweite variiert; in diesem Fall erweist es sich als günstig, wenn die Seiten des festen Polygons vorsortiert werden.

Wir haben bis jetzt Polygone ausgeklammert, die nicht konvex sind. Hier kann der Sachverhalt nicht so einfach geklärt werden; oft zerlegt man ein solches Polygon in konvexe Teilpolygone, oder man betrachtet seine konvexe Hülle. Es würde aber den Rahmen dieses Buches sprengen, wenn wir auf Einzelheiten eingingen. Dasselbe gilt auch für Schnittberechnungen mit Kreisbögen, Ellipsen oder gar Splines. In diesen Fällen ist ein iteratives Vorgehen oft unvermeidlich oder zumindest vom Zeitaufwand her günstiger, und man kann - zumindest lokal - die in diesem Abschnitt vorgestellten Ideen verwenden.

6.4. Rasteroperationen

6.4.1. Übersicht

Viele geometrisch-topologische Operationen im Umfeld von Rastermodellen können auf eine relativ kleine Anzahl von **Grundoperationen** zurückgeführt werden. Diese Grundoperationen der Rasterdatenverarbeitung zeichnen sich gegenüber Vektoroperationen (etwa den in Abschnitt 6.3 besprochenen Verschneidungen) durch ihre extreme Einfachheit aus. Ihr Nachteil besteht hauptsächlich in der *Fülle* solcher Grundoperationen, die für die Bearbeitung eines gesamten Bildes notwendig sind. Erst in letzter Zeit ist es durch die Steigerung der Hardwarequalität (Arrayprozessoren) möglich geworden, die theoretischen Entwicklungen auch in die Praxis umzusetzen.

Es kommen auch hier wieder die Vor- und Nachteile der beiden Strategien zum Vorschein. Deshalb können wir weder der einen noch der anderen Methode grundsätzlich den Vorzug geben; jede kann in dem ihr gemäßen Umfeld optimal operieren, und aus diesem Grund gibt es kein *Entweder-oder*, sondern vielmehr ein Nebeneinander, wo die beiden Strategien

einander ergänzen. Nicht zuletzt können Rastermodelle in Vektorform übergeführt werden und umgekehrt.

Es sei erwähnt, daß die im folgenden vorgestellten Grundoperationen aus der Bildverarbeitung kommen. Viele Bezeichnungen sind deshalb auch Entlehnungen aus deren Terminologie. So spricht man vom Spektrum der Grauwerte, von hohen und niedrigen **Frequenzen** usw. Eine detaillierte Darstellung wird hier nicht beabsichtigt. Für eine Einführung in dieses umfangreiche Forschungsgebiet - das auch bereits mit beachtlichen Ergebnissen aufwarten kann - sei auf Rosenfeld u. Kak (1976) und Kazmierczak (1980) verwiesen.

Wir wollen aber einige dieser Operationen herausgreifen, und wir werden sehen, daß diese genauso gut für thematische Belange angewendet werden können - wie ja überhaupt die Thematik in Rastermodellen von der Topologie kaum streng getrennt werden kann; wir haben darauf bereits im Kapitel 3 hingewiesen. Die thematische Kartographie kann sich also dieser Methoden bedienen und brauchbare Ergebnisse liefern. Eine gute Einführung in Rastermethoden der Kartographie bieten Weber (1982b) und Göpfert (1987).

Der nächste Abschnitt stellt kurz einige elementare Operationen vor; in den weiteren Abschnitten werden dann **Makro-Operationen** behandelt, die sich dieser elementaren Werkzeuge bedienen.

6.4.2. Elementare Operationen

Eine der wichtigsten Grundfunktionen der Rasterdatenverarbeitung ist die **Radiometrische Transformation**: Eine **Transferfunktion** wird auf die Grauwerte aller Zellen eines Rastermodelles angewendet. (Man beachte, daß im Sinne unserer bisherigen Überlegungen der Grauwert auch eine thematische Bedeutung haben kann, wie etwa das Mittel der Geoidhöhe in einem vorgegebenen Rechteck; wir werden in Zukunft von Grauwerten sprechen und deren etwaige allgemeinere Bedeutung stillschweigend voraussetzen.) Eine mögliche Transferfunktion wäre $y = 2 \cdot x$: sie würde in jeder Rasterzelle den Grauwert verdoppeln (Abb. 6.4a).

Eine interessantere Transformation erhalten wir, wenn die Transferfunktion einen bestimmten Bereich des Spektrums (der ursprünglichen Grauwerte) unterdrückt und andere Bereiche hervorhebt; dies wird in seiner einfachen Form als **Schwellwertbildung (thresholding)** bezeichnet (Abb. 6.4b). Alle Grauwerte unterhalb einer bestimmten Schranke werden zu Null; sie werden als "Störung" abgetan, während die Grauwerte oberhalb dieser Schranke einen anderen konstanten Wert zugewiesen bekommen. In diesem Fall wird aus dem

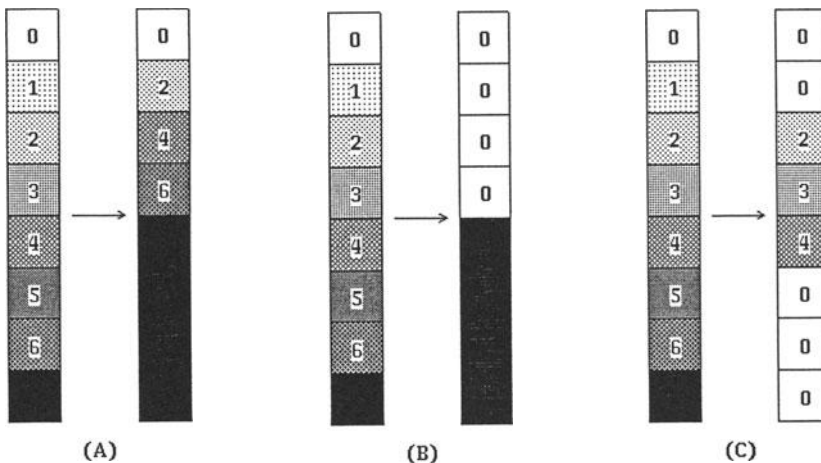


Abb. 6.4. Radiometrische Transformationen:
 Lineare Transferfunktion $y = 2 \cdot x$ (A)
 Schwellwertbildung am Grauwert 4 (B)
 Selektion der Grauwerte 2, 3, 4 (C)

Grauwertbild ein Bild mit nur zwei verschiedenen Graustufen (oder, in unserem Fall, Themen). Man spricht auch von einem *Zweistufenbild*.

Eine dritte Variante der radiometrischen Transformation ist die **Selektion** eines bestimmten Bandes aus dem Spektrum (**slicing**). Die Grauwerte aus diesem Band werden unverändert übernommen; alles, was unterhalb oder oberhalb dieses Bandes liegt, wird unterdrückt (Abb. 6.4c).

Neben der radiometrischen Transformation ist die **Parallelverschiebung** ein wichtiger Baustein für höhere Operationen (Abb. 6.5). Das Muster wird um einen bestimmten Betrag parallel verschoben, wobei dies in Nord-Ost-Süd-West-Richtung geschehen kann, aber auch (je nach der zugrundeliegenden **Metrik**, siehe Kapitel 2) in andere Richtungen.

Arithmetische und logische Kombinationen von Rasterbildern sind ein weiteres Beispiel für elementare Operationen. Bei der arithmetischen Kombination werden Grauwerte rasterzellenweise addiert, subtrahiert, etc. (Abb. 6.6). Bei der logischen Kombination werden Grauwerte logisch miteinander verknüpft (Abb. 6.7); beachten wir, daß man bei der logischen Kombination bereits von Zweistufenbildern ausgehen muß; die logischen Werte FALSCH und WAHR entsprechen den zwei möglichen Grauwerten pro Zelle. Die logische Kombination UND würde der Zelle des Ergebnisbildes genau dann den Wert WAHR zuordnen, wenn beide Ausgangsbilder in dieser Zelle WAHR sind. Die anderen logischen

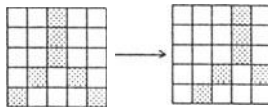


Abb. 6.5. Parallelverschiebung

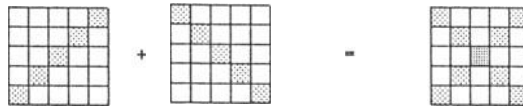


Abb. 6.6. Arithmetische Kombination (Addition)

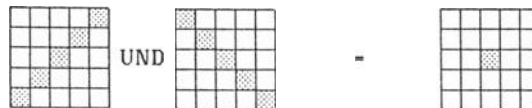


Abb. 6.7. Logische Kombination (UND)

Kombinationen (ODER, ausschließendes ODER, NICHT, usw.) verlaufen analog.

6.4.3. Verdicken und Verdünnen

Eine der wichtigsten *Makrooperationen*, die auf den oben erwähnten elementaren Methoden aufgebaut sind, ist die **Blow-Shrink-Methode** (Verdicken und Verdünnen). Dabei wird das Ausgangsmuster parallel verschoben, und zwar in alle Nachbarrichtungen. Anschließend wird eine logische ODER-Kombination des Ausgangsbildes mit allen seinen Parallelverschiebungen durchgeführt. Das Verdünnen geschieht analog, indem wir den Hintergrund verdicken. Aber wozu verdicken wir das Muster und machen dies wieder rückgängig? Der Grund liegt darin, daß wir zwar - im groben gesehen - dasselbe Muster wiedererhalten, von dem wir ausgegangen sind, allerdings sind kleine "Lücken" verschwunden.

Diese Methode kommt beispielsweise bei der Lückenfüllung zur Anwendung (Abb. 6.8). Lücken in einem Rasterbild sind Zellen, deren Grauwerte aufgrund einer schlechten Scan-Vorlage bzw. aufgrund einer unzureichenden Schwellwertbildung entstehen. Solche Lücken verhindern

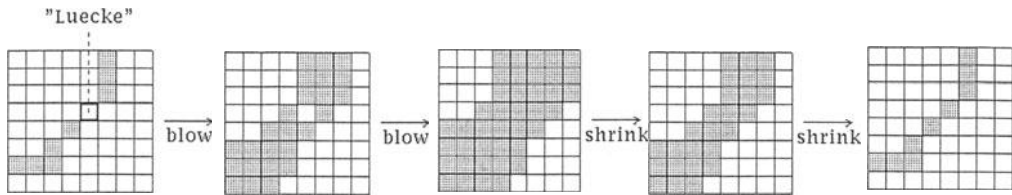


Abb. 6.8. Blow-Shrink-Methode zur Lückenfüllung

beispielsweise eine Vektorisierung des Bildes, sie verstümmeln thematische Zusammenhänge, und sie sind ein Hindernis bei der Füllung von Flächen (eine weitere Makrooperation, die wir weiter unten beschreiben). Das Füllen der Lücken ist also eine Vorbedingung für viele weitere Rasteroperationen.

Eine Lückenfüllung kommt aber auch im Rahmen der **Generalisierung** in Betracht, wo wir beim Übergang zu einem anderen Maßstab kleine "Störungen" eliminieren wollen (siehe auch Abschnitt 6.7). Ein Maß für den Grad der Generalisierung ist sicher die Anzahl der Verdickungen bzw. Verdünnungen, die wir unserer Blow-Shrink-Methode zugrunde legen. Verdicken wir das Muster extrem, so daß wir an den Bildrand stoßen, so haben wir auch eine extreme (wenn auch sinnlose) Generalisierung erreicht: das gesamte Bild wird homogen.

Neben der Lückenfüllung erzielen wir durch das sukzessive Verdicken und Verdünnen einen weiteren Generalisierungseffekt: Linien werden "geglättet", denn kleine Ausrundungen verschwinden beim Verdicken, treten aber beim darauf folgenden Verdünnen nicht mehr auf. Wir merken hier an, daß wir die Reihenfolge der Operationen auch umkehren können: wenn wir zuerst verdünnen, so verschwinden isolierte Grauwerte ("Störpixel"), die beim darauf folgenden Verdicken nicht mehr erscheinen. Die Operationen des Verdickens und Verdünnens **filtern** also das Bild. Wir beschäftigen uns weiter unten näher mit Filtermethoden.

Die Blow-Shrink-Methode dient aber auch zur rastermäßigen Erzeugung kartographischer Symbole. So läßt sich etwa eine Autobahnsignatur folgendermaßen erzeugen: man geht von einer Achse bzw. einem Skelett aus; diese Achse wird bis zur Innenbreite der zu erzeugenden Signatur verdickt. Außerdem verdickt man die Ausgangsachse auch bis zur Außenbreite. Nun folgt als arithmetische Operation eine Subtraktion von Außen- und Innenbreite; schließlich wird dem Ergebnis noch durch eine ODER-Operation die ursprüngliche Achse überlagert (Weber 1982b).

6.4.4. Füllen vorgegebener Flächen

Neben der Blow-Shrink-Methode ist das Füllen einer vorgegebenen Fläche ("area flooding") eine weitere wichtige Makrooperation. In diesem Fall ist eine Umrandung einer Fläche vorgegeben. Man setzt nun in das Flächeninnere einen **Füllungskeim**, den man sukzessive verdickt. Bei jedem Schritt verwendet man den vorgegebenen Rand als **Maske**, die verhindert, daß der verdickte Keim über diesen Rand hinauswächst. Das Maskieren geschieht etwa durch eine logische AND-NOT-Verknüpfung des verdickten Keimes mit dem Rand (Abb. 6.9). Wieder kann man hier verschiedene **Metriken** zugrunde legen.

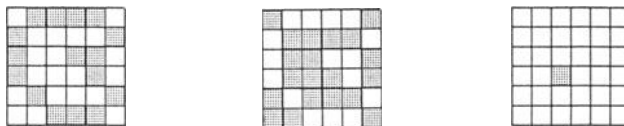


Abb. 6.9a. Rand R, dessen Komplement NOT R, sowie Füllungskeim $F(0)$

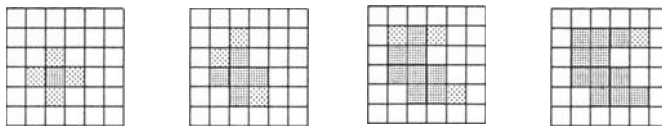


Abb. 6.9b. Sukzessives Verdicken und Maskieren:
 $F(k) = \text{BLOW } F(k-1) \text{ AND NOT } R$

Wichtig ist vor allem, daß der Rand geschlossen ist, sonst "tritt die Füllfarbe aus", und der Algorithmus versagt. Man muß daher gegebenenfalls den Rand durch eine Blow-Shrink-Operation vorbehandeln. Die Operation des Füllens hat eine sehr anschauliche Anwendung. Auf einem graphischen Bildschirm werden auf diese einfache Weise flächige Darstellungen mit Farben, Schraffuren etc. gefüllt. Das Löschen von Bildteilen kann man analog durch Füllen mit der Hintergrundfarbe bewerkstelligen. Für unsere Anwendungen ergeben sich jedoch viel allgemeinere Einsatzmöglichkeiten. So können wir zum Beispiel auf diese Weise flächigen Objekten eine Thematik zuordnen, wenn wir ihre Umrandung kennen, das heißt, wir können die Thematik vom Rand einer Fläche in ihr Inneres fortsetzen. Nebenbei bemerkt, funktioniert der Algorithmus auch dann, wenn Aussparungsflächen (Inseln) auftreten. Auch die Frage, ob sich zwei Punkte innerhalb derselben vorgegebenen Umrandung befinden, läßt sich auf diese Weise lösen: So können wir

"versuchsweise" in einem dieser Punkte einen Füllungskeim ansetzen und diesen verdicken. Wird der zweite Punkt letztendlich überdeckt, so befinden sich beide Punkte im selben Gebiet.

Schließlich wollen wir noch erwähnen, daß die Füllung einer vorgegebenen Fläche mit kartographischen Flächensymbolen (Moor, Wiese, Gewässer usw.) auch auf diesem Algorithmus aufbaut: Die zu füllende Fläche wird vorläufig mit einer Deckfarbe gefüllt, die es erlaubt, die Rasterzellen des Bildes in zwei Klassen einzuteilen: solche, die diese Deckfarbe aufweisen, und daher zur Fläche gehören, und andere, die nicht in der Fläche enthalten sind. Der Prototyp des Flächensymbols (der Basic Repeat) liegt in Form eines Bitmusters vor. Dieser Prototyp wird nun durch Parallelverschiebung und logische OR-Operationen auf ein genügend großes Gebiet vervielfältigt. Nun bildet man mittels AND den Durchschnitt des Symbolgebietes mit den Rasterzellen, welche die Deckfarbe enthalten, und erreicht dadurch auf höchst einfache Weise eine flächendeckende Füllung mit kartographischen Symbolen. (Probleme, die durch Verstümmelungen am Rand auftreten, oder Flächen, die kleiner als der Basic Repeat sind, klammern wir hier aus.)

6.4.5. Abstandstransformierte und Skelett

Eine weitere wichtige Makrooperation ist die Bildung der Abstandstransformierten. Man geht von einer Anhäufung von Rasterzellen gleichen Grauwertes aus. In unserem Fall kann es sich um Rasterzellen gleicher Thematik handeln, also etwa die Rasterzellen, die ein Waldgebiet überdecken. Für jede Rasterzelle kann man nun den *kürzesten Abstand* zum Rand (im Sinne der gewählten Metrik) ermitteln. Die Ergebnisse kann man in einem sekundären Rasterbild eintragen, eben in der Abstandstransformierten. Deren Grauwerte sind demnach als Abstände zu interpretieren. Tief im Inneren des Waldes werden also die Grauwerte der Abstandstransformierten hoch sein, während sie nach außen hin immer niedriger werden. Man spricht in diesem Fall auch vom "Abstandsgebirge". Die Maxima dieses Abstandsgebirges bezeichnet man als Skelett.

Die Abstandstransformierte läßt sich sehr einfach durch unsere elementaren Operationen erzeugen: wir stellen zuerst verdünnte Versionen unseres Zellhaufens her; wir magern diesen um eine, zwei, drei Zellenbreiten ab, solange, bis er verschwindet; sodann addieren wir alle diese Verdünnungen und erhalten so die Abstandstransformierte (Abb. 6.10).

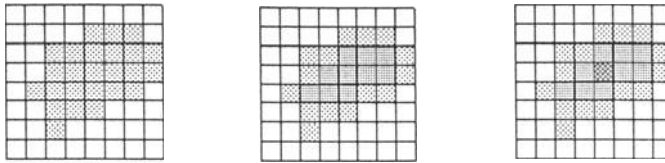


Abb. 6.10. Bildung der Abstandstransformierten

6.4.6. Filterung mittels Konvolution

Als nächste Makrooperation ist die Filterung zu nennen. Der Ausdruck kommt - wie so viele andere in der Bildverarbeitung gebräuchliche Ausdrücke - aus der Nachrichtentechnik; dort versteht man darunter die Unterdrückung bestimmter Frequenzbereiche einer durch periodische Schwingungen gekennzeichneten Erscheinung. Der zugrundeliegende Prozeß ist eine Funktion der Zeit. In unseren Anwendungen handelt es sich um *Änderungen der Grauwerte als Funktion des Ortes*. Ändern sich die Grauwerte von Ort zu Ort nur geringfügig (wie zum Beispiel im Inneren einer homogenen Fläche), so interpretieren wir dies als niedrigfrequente Schwingung. Am Rand einer solchen Fläche ändern sich die Grauwerte abrupt; wir haben es dort mit einer hochfrequenten Schwingung zu tun.

Wir können nun - je nach dem gewünschten Resultat - die hochfrequenten Änderungen unterdrücken und damit die niedrigen Frequenzen stärker betonen: dies ist eine **Tiefpaßfilterung**; sie macht die Ränder unscharf, mildert den gezahnten Treppeneffekt, unterdrückt kleine "Löcher" und Ausreißer. Unterdrücken wir aber die niedrigfrequenten Anteile und lassen nur die hochfrequenten Anteile zu, so ist dies eine **Hochpaßfilterung**. Sie hebt Ränder und Konturen stärker hervor. Für beide Varianten finden wir im Rahmen der digitalen Kartographie Anwendungen.

Wie können wir eine solche Filterung erreichen? Eine Möglichkeit ist die **Konvolution**. Dabei wird jeder Grauwert durch ein gewogenes Mittel seiner Umgebung ersetzt. Wir können dies als eine Folge von Elementaroperationen ansehen. Wir verschieben das Urbild etwa nach oben, rechts, unten und links und addieren dann das Urbild und alle Verschiebungen. Wir sehen, daß diese Operation ähnlich wie eine BLOW-Operation abläuft, nur daß wir die Bilder hier addieren, während wir sie dort einer logischen ODER-Verknüpfung unterzogen haben. Da wir jedoch in unseren Anwendungen den Grauwert nicht unbedingt nur wörtlich als solchen verstehen, sondern durch ihn eine Thematik ausdrücken, kann der Summe von Grauwerten nicht immer eine sinnvolle Deutung gegeben werden. In diesem Fall können wir

eine Schwellwertoperation anwenden, um die Resultate der Filterung in unsere Thematik abbilden zu können (Abb. 6.11a).

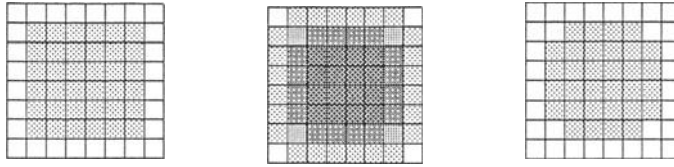


Abb. 6.11a. Tiefpaßfilterung mit anschließender Schwellwertbildung

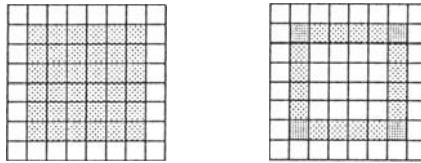


Abb. 6.11b. Hochpaßfilterung mit anschließender Unterdrückung negativer Werte

Dieser Konvolution liegt die folgende Formel zugrunde, in welcher $G(x,y)$ der Grauwert an der Stelle (x,y) ist, während $G'(x,y)$ den gefilterten Grauwert angibt, und w die Elemente einer Gewichtsmatrix W sind.

$$\begin{aligned}
 G'(x,y) = & G(x, y) * w(2,2) \\
 & + G(x-1, y) * w(2,1) + G(x+1, y) * w(2,3) \\
 & + G(x, y-1) * w(1,2) + G(x, y+1) * w(3,2) \\
 & \dots \quad (6-4-1)
 \end{aligned}$$

Die Gewichtsmatrix lautet in unserem Fall:

$$W1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (6-4-2)$$

In dieser Formel gehen wir von einem Ursprung aus, der links oben liegt; die x -Werte werden also von links nach rechts gezählt, die y -Werte von oben nach unten. Die Formel muß am Bildrand entsprechend modifiziert werden. Natürlich kann diese Gewichtsmatrix im allgemeinen größer als 3×3 sein. Je größer sie ist, desto stärker ist die Filterung. Nimmt man sie im Extrem so groß wie das gesamte Bild an, so wird ein einheitlicher Grauwert für dieses Bild erzeugt. Außerdem können neben Nullen und Einsen auch andere Zahlen auftreten. Interessant ist zum Beispiel folgende Gewichtsmatrix:

$$W_2 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (6-4-3)$$

Sie entspricht einer Multiplikation der Grauwerte des Urbildes mit 4 und einer nachfolgenden Subtraktion der nach den vier Himmelsrichtungen verschobenen Bilder. Das Ergebnis ist eine *Hochpaßfilterung*. Das Innere eines homogenen Zellhaufens weist nach der Filterung Nullwerte auf, während der Rand hervorgehoben wird. Offenbar sind die negativen Vorzeichen in der Matrix W verantwortlich für die Unterdrückung der niedrigfrequenten Schwingungen. Die zunächst auftretenden negativen Grauwerte am äußersten Rand können wir durch eine logische AND-Verknüpfung mit dem Urbild unterdrücken (Abb. 6.11b).

Die Filtermethode der Konvolution ist ein einfaches und doch mächtiges Werkzeug, das in vielfältiger Weise eingesetzt werden kann. Das nächstliegende Einsatzgebiet ist die Bildverbesserung. Wenn etwa die Rastereinteilung zu grob ist, so kann man durch eine Tiefpaßfilterung eine Milderung des Treppeneffektes erreichen. Ein Beispiel dafür ist in Abb. 6.12 dargestellt. Dort wurde jede Rasterzelle durch neun kleinere Zellen ersetzt, und daraufhin wurde das daraus entstandene Bild dem Filterungsoperator W_1 unterzogen.

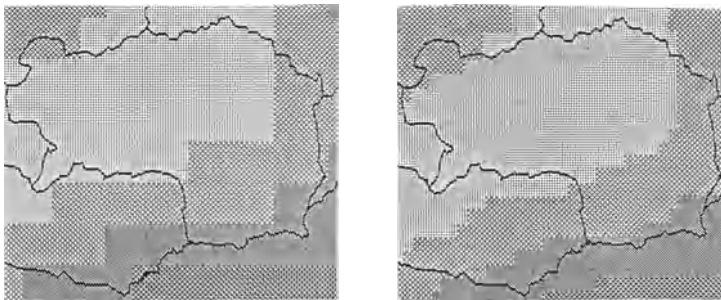


Abb. 6.12. Bildverbesserung durch Tiefpaßfilterung

Ein weiteres wichtiges Einsatzgebiet der Filterung ist die **Generalisierung** von Flächen im Rasterformat (Abb. 6.13). Die Tiefpaßfilterung schließt kleine Lücken zwischen homogenen Zellhaufen und füllt Einbuchtungen auf. Dadurch wachsen einzelne Häuser zu Ortskernen zusammen; "Inseln" - also Waldlichtungen und dergleichen - werden zum Verschwinden gebracht; all dies wird unter der Bezeichnung **Zusammenfassung** als Teilaspekt der Generalisierung betrachtet. Durch das Auffüllen der Einbuchtungen werden die Umrandungen von Rasterflächen **vereinfacht**. Außerdem **eliminiert** die Tiefpaßfilterung kleine isolierte Flecken,

deren Ausdehnung das Maß der Filterung nicht übersteigt; einzeln stehende Häuser oder kleine Baumgruppen scheinen in der generalisierten Karte nicht mehr auf. Dies wird als **Auswahl** bezeichnet.

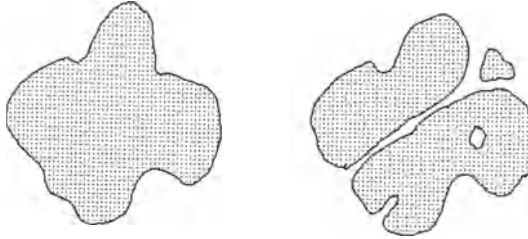


Abb. 6.13. Generalisierung mit Hilfe von Filtermethoden

Eine solche Generalisierung über ein automatisierbares Filterverfahren kann in Einzelfällen zu unerwünschten Ergebnissen führen. So müssen prägnante Merkmale einer Karte - wie etwa Steilhänge oder Bruchlinien - auch in der generalisierten Karte als solche erkennbar sein. Außerdem müssen Bauwerke, die der Orientierung in der Karte dienen (Kapellen, Bildstöcke, Schutzhütten), vor einer Elimination geschützt werden; das heißt, daß wir in einem solchen Fall bewußt eine Maßstabsverfälschung in Kauf nehmen, um den **Informationsgehalt** der Karte zu erhalten. Wir können hier mit einer Hochpaßfilterung gegensteuern, denn diese hebt ja gerade Details und Ränder (also die hochfrequenten Anteile des Bildes) hervor.

Oft wird neben der Zusammenfassung, der Vereinfachung und der Auswahl die **Verdrängung** als Generalisierungsaspekt genannt. Topologische Sachverhalte werden - wenn es nicht anders geht - der Geometrie vorgezogen. In einem engen Gebirgstal, das einen Fluß, eine Straße, eine Eisenbahnlinie und womöglich noch eine Autobahn aufnimmt, müssen diese linienhaften Strukturen gegebenenfalls von ihrer geometrischen Position verdrängt werden, um den Informationsgehalt der Karte sicherzustellen. Im Rahmen der automatisierbaren Rasteroperationen bietet sich hier der Einsatz der vorhin besprochenen **Abstandstransformierten** an. Sie spiegelt die Abstände der Rasterzellen verschiedener Themen (also zum Beispiel zwischen Straße und Eisenbahn) wider. Dort, wo diese Abstände eine bestimmte minimale Schranke unterschreiten, muß verdrängt werden. Auf die entsprechende Problematik wollen wir hier nicht näher eingehen. Es sei auf eine detaillierte Darstellung in Lay u. Weber (1983) verwiesen.

6.4.7. Filterung mittels Fourier-Transformation

Zur Filterung eines Grauwertbildes kann neben der Konvolution auch die diskrete Fourier-Transformation verwendet werden. Ihr liegt die Idee zugrunde, daß man die Ausgangsdaten einer durch periodische Schwankungen gegebenen Erscheinung - bei uns sind es die Grauwerte des Bildes - zunächst in einen **Frequenzbereich** transformiert, weil die Anwendung einer bestimmten Operation - etwa einer Filterung - dort einfacher ist. So vereinfachen sich beispielsweise Faltungsintegrale im Ortsbereich zu Multiplikationen im Frequenzbereich. Andererseits verursacht die Transformation in den Frequenzbereich einen gewissen Aufwand, und natürlich müssen wir das Ergebnis einer Filterung mittels der inversen Transformation wieder in den ursprünglichen Raum, den **Ortsbereich**, zurückbringen.

Wir transformieren also zunächst jeden Grauwert $G(x, y)$ des Urbildes in den **Frequenzbereich**:

$$S(a, b) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} G(x, y) e^{-2i\pi(ax + by) / N} \quad (6-4-4)$$

N ist die Dimension des Urbildes; bei der Berechnung eines transformierten Grauwertes werden also sämtliche Grauwerte des Urbildes benötigt. Anschließend wird der transformierte Grauwert im Frequenzbereich einer Filterung unterzogen:

$$S'(a, b) = H(a, b) * S(a, b) \quad (6-4-5)$$

Zuletzt kehren wir wieder in den **Ortsbereich** zurück:

$$G'(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} S'(a, b) e^{2i\pi(ax + by) / N} \quad (6-4-6)$$

Wir haben hier nur die einfachste Form der Fourier-Transformation dargestellt. Sie ist relativ aufwendig, da man immer gleichzeitig die Information des gesamten Urbildes benötigt. Es gibt verschiedene Varianten, die effizienter arbeiten, wie zum Beispiel die **Fast-Fourier-Transformation (FFT)**. Wir wollen hier jedoch nicht näher auf diese eingehen.

6.4.8. Geometrische Transformationen von Rasterdaten

Geometrische Transformationen, wie etwa überbestimmte Helmer-Transformationen über Paßpunkte, der Übergang zwischen geographischen und kartesischen Koordinatensystemen, Zusammenlegungen aneinandergrenzender oder überlappender Bilder mit Randausgleichung usw. müssen in jedem Paket von Algorithmen zur Rasterdatenverarbeitung vorhanden sein. Aber auch Entzerrungen sind durchzuführen. So müssen wir beispielsweise bei Satellitenbildern den Effekt der Erdrotation berücksichtigen: während der Aufnahme dreht sich die Erde weiter, so daß sowohl das Gitter als auch das einzelne Pixel nicht mehr rechteckig ist. Diese Problematik und die entsprechenden Lösungsmöglichkeiten werden in Monmonier (1982) behandelt.

Manchmal wird der Begriff "Transformation" auch nur auf jenen Teilbereich beschränkt, wo wir eines der beiden Systeme noch gar nicht kennen, sondern erst aufgrund der Paßpunktinformation ermitteln wollen. Die Überführung vorgegebener Koordinaten in andere - ebenso vorgegebene - Koordinaten jedoch (wobei Grauwerte entsprechend mitgeführt werden sollen), nennt man in diesem Fall Resampling.

Für eine geometrische Transformation müssen wir die Koordinaten (x,y) der Rasterzellen eines Bildes in Koordinaten (x',y') eines anderen Bildes umrechnen:

$$\begin{aligned} x' &= f(x,y) \\ y' &= g(x,y) \end{aligned} \quad (6-4-7)$$

Wir können uns so behelfen, daß wir für jede Rasterzelle (x',y') des transformierten Bildes alle Rasterzellen (x,y) des ursprünglichen Bildes sowie auch deren Grauwerte ermitteln, und den Grauwert an der Stelle (x',y') dementsprechend - etwa durch Mittelung - festsetzen. Das bedingt natürlich, daß wir die Umkehrabbildung kennen; dies kann zeitaufwendig sein. Auch hier gibt es - wie bei der Fourier-Transformation - Algorithmen, die - was die Rechenzeit betrifft - effizienter sind; hier sei auf Weber (1982b) verwiesen.

6.4.9. Effizienz von Rasteralgorithmen

Wir haben in den vorangegangenen Abschnitten mehrfach auf die Effizienz von Rasteroperationen hingewiesen. In jenen geometrisch-topologischen Modellen, denen eine Rasterstruktur zugrunde liegt, ist die Menge der Daten (und damit die Menge der einzelnen Verarbeitungsschritte) ein entscheidender Faktor. Wir haben gesehen, daß die Behandlung einer

einzelnen Rasterzelle bzw. die Behandlung einer Zellanhäufung beschränkten Ausmaßes extrem einfach ist, und dies ist der große Vorteil eines Rastermodelles.

Es kommt uns hier auch die Entwicklung der Hardware entgegen. Die Preise für Arrayprozessoren sind in den letzten Jahren so stark gesunken, daß die Wirtschaftlichkeit ihres Einsatzes durchaus gegeben ist. Es handelt sich dabei um eine Anordnung von mehreren Rechenwerken, die *simultan* arbeiten können, während herkömmliche Computer nur ein solches Rechenwerk besitzen, und daher sämtliche Rechenoperationen *hintereinander* ablaufen müssen. Es gibt bereits Arrayprozessoren mit mehreren tausend solcher Rechenwerke, die es erlauben, Millionen von Elementaroperationen pro Sekunde auszuführen.

Wie können wir nun die Möglichkeiten ausnützen, die ein solcher Arrayprozessor bietet? Wir müssen einen Algorithmus so auslegen, daß es möglich ist, ihn in mehrere Teile aufzuspalten, die gleichzeitig ablaufen können. Diese Aufspaltung (und auch die Überwachung und Steuerung der Rechenwerke) nimmt uns dann die Software ab, die mit dem Arrayprozessor im allgemeinen mitgeliefert wird. Es handelt sich dabei meistens um einen intelligenten Compiler, der ein bestehendes Programm so umformt, daß es auf dem Arrayprozessor optimal läuft. Für den Anwendungsprogrammierer ist somit eine weitgehende Unabhängigkeit vom jeweiligen Rechner erreicht: dasselbe Programm, das auf einem kleineren Rechner auf herkömmliche Art ausgeführt wird, kann nach der Compilation auf dem Arrayprozessor voll dessen Möglichkeiten ausnützen.

Allerdings muß der Algorithmus selbst gewisse Voraussetzungen mitbringen:

- der Algorithmus soll **lokal** sein: die Verarbeitung einer Rasterzelle berührt möglichst wenige andere Zellen. Die Menge aller dieser Zellen muß beschränkt sein. Die Fourier-Transformation ist demnach **global**, da man für jeden gefilterten Grauwert alle Grauwerte des Urbildes benötigt.
- der Algorithmus soll **homogen** sein: die Verarbeitungsvorschrift soll für alle Rasterzellen die gleiche sein (mit Ausnahme des Bildrandes, wo natürlich immer eine Sonderbehandlung erforderlich ist).
- der Algorithmus soll nur aus **Elementaroperationen** aufgebaut sein, so wie wir sie in den vorangegangenen Abschnitten beschrieben haben. Insbesondere darf es keine zeitlichen Bedingungen für die Verarbeitung geben; die Bearbeitung einer Zelle darf also nicht eine Vorverarbeitung einer anderen Zelle voraussetzen.

Bringt ein Algorithmus all diese Voraussetzungen mit, so ist er ideal für einen Einsatz auf einem Arrayprozessor geeignet. Der oben beschriebene Compiler kann ihn optimal umsetzen. Es ist aber klar, daß wir nicht immer alle diese Forderungen erfüllen können. Manchmal werden sie einander sogar widersprechen. Außerdem zahlen wir für die totale Gleichschaltung aller Zellen einen Preis, der darin besteht, daß viele Operationen unnötigerweise durchgeführt werden. Wir müssen im konkreten Anwendungsfall das Für und Wider abwägen und einen Kompromiß anstreben.

6.5. Interpolation und Approximation von Kurven

6.5.1. Modellbildung

Die Interpolation bzw. Approximation von Kurven ist zunächst (aber nicht nur) ein *Darstellungsproblem* im engeren Sinn, also die Suche nach einer möglichst glatten, dem Auge gefälligen Form, die trotzdem die wesentlichen Eigenschaften einer Kurve erkennen läßt. Es geht aber auch um die *interne Repräsentation*, also eine Minimierung des Speicheraufwandes, bei gleichzeitiger Beibehaltung eines möglichst großen Informationsgehaltes, und um die Möglichkeit, geometrische *Auswertungen* (z.B. Längenberechnungen) durchzuführen. Im Rahmen der *Generalisierung* stehen wir vor dem Problem, durch eine Reihe von gegebenen Punkten eine Kurve hindurchzulegen, welche "in der Nähe" dieser Punkte liegt und trotzdem "einfacher" als eine exakt durch diese Punkte verlaufende Kurve ist. Bei der *automatisierten Erfassung* von Kurven muß ebenfalls das Problem gelöst werden, wie wir aus der Fülle von angebotenen Kurvenpunkten einige wenige auswählen können, welche die Kurve möglichst gut beschreiben.

Dies sind nur einige Aspekte aus der Vielzahl von Bereichen, in denen die Interpolation und Approximation von Kurven eine Rolle spielt. Für Flächen gelten ähnliche Überlegungen; wir verweisen dazu auf Abschnitt 6.6. Auch die Verbindung von Flächen- und Kurveninterpolation, wie sie etwa bei der Rückinterpolation von Schichtenlinien aus einem digitalen Geländemodell auftritt, sei hier erwähnt.

Eine linienhafte Struktur wird in einem raumbezogenen Informationssystem auf ein geeignetes Modell abgebildet, dessen Verfeinerungsgrad vom Umfeld der möglichen Anwendungen abhängt. Das einfachste Modell für die Verbindung zweier Punkte besteht in der Annahme, daß es sich um eine geradlinige Verbindung handelt. Dieses Modell ist für viele Anwendungen ausreichend genau; das Liniennetz des öffentlichen Verkehrs in einer Stadt wird dadurch hinreichend gut beschrieben, zumindest was dessen *topologischen* Aspekt angeht (z.B. "wo kann man von Linie A auf Linie B umsteigen?"). Wir wissen, daß dazu noch ein

thematischer Aspekt kommt (z.B. "A ist eine eingleisig geführte Straßenbahnlinie"), der uns aber hier nicht weiter interessiert.

Oft jedoch ist die Annahme eines solchen einfachen Modells zu grob; wir verfeinern zum Beispiel das Netz von Straßenbahnlinien, indem wir im Modell Kreisbögen und Übergangsbögen einführen; auf diese Weise können wir dann direkt aus der Geometrie bzw. Topologie Entfernungen ableiten.

Ein derartiges Modell aus Geraden, Kreisbögen und Übergangsbögen versagt jedoch beispielsweise bei Schichtenlinien. Eine solche Linie wird uns im allgemeinen nicht den Gefallen tun, sich in ihrem gesamten Verlauf einer mathematisch definierbaren Kurve anzupassen. Wir können nun einen Ausweg darin erblicken, daß wir gar nicht erst versuchen, ein solches mathematisches Modell zu finden, sondern die Kurve durch eine sehr dichte Folge von Kurvenpunkten zu beschreiben. Dies wird etwa bei der Darstellung einer Kurve auf einem nach dem Rasterprinzip arbeitenden Ausgabemedium getan: die Kurve wird durch eine Folge von Pixeln gleicher Farbe beschrieben. Legen wir unsere Datenstruktur (und die darauf aufbauenden Bearbeitungsmethoden) nach dem Rasterprinzip an, so können wir diese Methode auch für das interne Datenmodell benutzen. Für eine Modellierung und Bearbeitung nach dem Vektorprinzip hingegen würde der Speicherplatzbedarf für eine solche Kurve, das Verhalten bei Maßstabsveränderungen, die Evaluierung der Geometrie u.a.m. ungünstig zu Buche schlagen.

Wir müssen uns daher darauf beschränken, die Kurve durch *wenige*, jedoch für den Kurvenverlauf *charakteristische* Punkte zu beschreiben, aus denen wir sie jederzeit reproduzieren können. Wir fordern also gleichzeitig ein Minimum von Punkten und ein Maximum in der Abbildungsqualität des Modells. Dort, wo sich die Kurve sehr stark ändert, benötigen wir mehr Punkte als in anderen Bereichen. Die Form der Kurve zwischen diesen gegebenen Punkten interpolieren wir nach bestimmten Kriterien, welche die Qualität des Modells widerspiegeln. Werden die gegebenen Punkte dabei reproduziert, so handelt es sich um eine Interpolation im engeren Sinn; ansonsten versucht man, die Kurve "möglichst nahe" an die gegebenen Punkte heranzuziehen (*Approximation*). Die Grenze zwischen den beiden Varianten verschwimmt, sobald wir die Abweichungen der Punkte von der interpolierten Kurve als *Meßfehler* betrachten; die folgenden Ausführungen gelten also nicht nur für die Interpolation, sondern - nach entsprechender Verallgemeinerung - auch für die Approximation. Außerdem sei angemerkt, daß wir uns zwar auf die Interpolation in der Ebene beschränken, daß aber einer Verallgemeinerung für Raumkurven nichts im Wege steht.

Eine interpolierte Kurve muß mehrere Forderungen erfüllen:

- der **Grad der Stetigkeit** soll genügend hoch sein;
- das Modell soll **achsenunabhängig** sein;
- die Stützpunkte sollen nur einen **lokalen** Einfluß ausüben;
- ein **Überschwingen** soll vermieden werden.

6.5.2. Interpolation

Die bekannteste Methode zur Interpolation benützt den **Spline**; es ist dies ein stückweise (d.h. in k Teilintervallen gesondert) definiertes Polynom $p(x)$ höchstens m -ten Grades, für welches man an den Nahtstellen Glattheitsbedingungen vorschreibt; die Ableitungen bis zu einer bestimmten Ordnung $r-1$ sollen links und rechts der Nahtstelle übereinstimmen.

$$p(x) = p_i(x) \quad x_i \leq x \leq x_{i+1} \quad i=0,1,\dots,k-1 \quad (6-5-1)$$

$$p_{i-1}^{(j)}(x_i) = p_i^{(j)}(x_i) \quad j=0,1,\dots,r-1 \quad i=1,2,\dots,k-1$$

Je nach der Wahl von m spricht man von **linearen Splines** ($m=1$), **quadratischen Splines** ($m=2$) und **kubischen Splines** ($m=3$). Sie erfüllen die geforderten Bedingungen an die Glattheit und den lokalen Einfluß (Abb. 6.14).

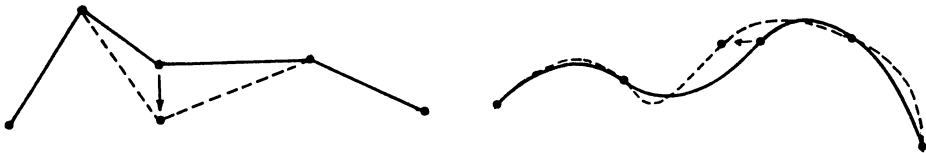


Abb. 6.14. Lokales Fehlerverhalten von Splines

Nach der Vorgabe von Datenwerten und Übergangsbedingungen ergibt sich ein Gleichungssystem für die einzelnen Polynomkoeffizienten, dessen Auflösung aber recht umfangreich werden kann, weil die beteiligten Polynome nicht nur in ihrem engeren Wirkungsbereich (im jeweiligen Teilintervall) sondern im gesamten Bereich ausgewertet werden müssen. Effizienter ist daher der Ansatz, der den Spline als Linearkombination von besonders einfach gearteten Bausteinen (**Basisplines**) darstellt:

$$p(x) = \sum_{i=-m}^{k-1} a_i N_{im}(x) \quad (6-5-2)$$

Jeder Basisspline ist nur in einem beschränkten Bereich von Null verschieden; er braucht daher nur an wenigen Stellen ausgewertet werden, was die Anzahl der erforderlichen Rechenschritte entscheidend vermindert. Ein Basisspline m -ter Ordnung läßt sich außerdem höchst einfach rekursiv berechnen: er ergibt sich durch Faltung zweier Basissplines $(m-1)$ -ter Ordnung; den Anfang in dieser Kette macht der Basisspline null-ter Ordnung, der folgendermaßen definiert ist:

$$N_{i0}(x) = \begin{cases} 1 & \dots x_i \leq x \leq x_{i+1} \\ 0 & \dots \text{sonst} \end{cases} \quad (6-5-3)$$

Die Rekursionsvorschrift für den Basisspline m -ter Ordnung lautet

$$N_{im}(x) = \frac{x-x_i}{x_{i+m}-x_i} N_{i,m-1}(x) + \frac{x_{i+m+1}-x}{x_{i+m+1}-x_{i+1}} N_{i+1,m-1}(x) \quad \dots (6-5-4)$$

Durch die Faltung zweier konstanter Anteile erhalten wir also die Dachfunktion für den Basisspline erster Ordnung (Abb. 6.15). Sie ist in zwei Intervallen von Null verschieden. Durch abermalige Faltung erhalten wir den quadratischen Basisspline, der in drei Intervallen von Null verschieden ist, usw. (Abb. 6.16).

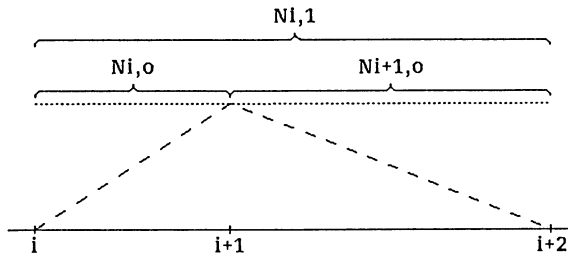


Abb. 6.15. Basissplines null-ter und erster Ordnung

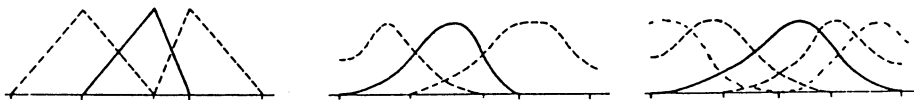


Abb. 6.16. Linearer, quadratischer und kubischer Basisspline

Für einen vorgegebenen Punkt (t_j, y_j) bleiben also von der obigen Summe nur $m+1$ Summanden übrig, die von Null verschieden sind. Fällt der Punkt etwa in das Intervall (x_i, x_{i+1}) , so erhalten wir eine Gleichung

$$a_i N_{im}(t_j) + a_{i-1} N_{i-1,m}(t_j) + \dots + a_{i-m} N_{i-m,m}(t_j) = y_j$$

... (6-5-5)

Setzen wir alle gegebenen Punkte in dieser Form an, so ergibt sich ein gebändertes Gleichungssystem für die Koeffizienten a_i . Wir erwähnen hier, daß es Einschränkungen für die Lage der gegebenen Punkte bezüglich der Stützintervalle (x_i, x_{i+1}) gibt; konzentrieren sich diese gegebenen Punkte i in $i+1$ bestimmten Bereichen, so bleiben einige Intervalle unterbestimmt, und das Gleichungssystem ist nicht mehr eindeutig lösbar. Außerdem ist anzumerken, daß man oft Stützintervalle gleicher Länge verwendet, was Vereinfachungen in den obigen Formeln bringt.

Bisher haben wir unsere Interpolationskurve explizit formuliert, das heißt, wir schließen von vorgegebenen x -Werten auf die jeweiligen y -Werte. Bei einer Schichtenlinie jedoch werden - je nach Wahl des Koordinatensystems - für einen x -Wert im allgemeinen mehrere y -Werte auftreten. Wir müssen daher zu einer **Parameterdarstellung** der Kurve übergehen: wir stellen sowohl x als auch y durch einen Kurvenparameter t dar; es ergeben sich also zwei Summen

$$x(t) = \sum_{i=-m}^{k-1} a_i N_{im}(t)$$

$$y(t) = \sum_{i=-m}^{k-1} b_i N_{im}(t)$$

(6-5-6)

Fassen wir die beiden Gleichungen zusammen, so ergibt sich

$$P(t) = \sum_{i=-m}^{k-1} P_i N_{im}(t)$$

(6-5-7)

wobei $P(t)$ und P_i Vektoren sind; $P(t)$ entspricht jeweils einem gegebenen i Punkt, und die Vektoren P_i sind zunächst Zusammenfassungen der zu schätzenden Koeffizienten (a_i, b_i) . Natürlich können wir sie ebenfalls als Punkte ansehen; sie werden als **Leitpunkte** ("guiding points") bezeichnet. Im Rahmen eines Interpolationsproblems ergeben sie sich also durch die Auflösung eines (gebänderten) Gleichungssystems.

Wie wir bereits im einleitenden Abschnitt festgestellt haben, geht die Interpolation in eine Approximation über, wenn wir eine Überbestimmung durch einen vermittelnden Ausgleich lösen; in diesem Fall liegen die Leitpunkte nicht mehr auf der Kurve, sondern in ihrer Nähe. Immer aber spielen sie die Rolle eines stenographischen Kürzels für die Kurve; aus ihnen läßt sich die Kurve jederzeit reproduzieren.

Wir haben uns bis jetzt auf die Interpolation bzw. Approximation durch Splines beschränkt. Die im einleitenden Abschnitt erhobenen Forderungen können aber ebensogut erfüllt werden, wenn wir in den obigen Formeln die Terme $N_{im}(t)$ durch andere - ebenso elementare - Bausteine ersetzen: eine solche Variante ergibt sich, wenn wir Legendre-Polynome verwenden:

$$N_{im}(t) = \binom{m}{i} t^i (1-t)^{m-i} \quad (6-5-8)$$

Die resultierende Kurve wird **Bezierkurve** genannt. Sie hat ähnliche Eigenschaften wie die Splinekurve: im allgemeinen unterdrückt sie Überschwingungen noch besser; daher eignet sie sich gut für CAD-Aufgaben; aber auch in unserem Anwendungsbereich wird sie erfolgreich verwendet, so zum Beispiel bei der Rückinterpolation von Schichtenlinien aus einem digitalen Geländemodell, wo es darauf ankommt, daß sich einzelne Linien nicht überschneiden (Kropatsch 1983).

Eine weitere Variante wurde von Akima (1970) entwickelt. Er läßt die stetige Differenzierbarkeit an den Nahtstellen außer acht und verwendet die so gewonnenen Freiheitsgrade ebenfalls zu einer Reduzierung der Überschwingungen.

Es gibt auch Anwendungen (etwa im CAD-Bereich), wo man Leitpunkte nicht aufgrund einer Interpolation bzw. Approximation gegebener (gemessener) Punkte ermittelt, sondern von den Leitpunkten selbst ausgeht, indem man sie von vornherein festsetzt. Durch eine geschickte Wahl dieser Leitpunkte kann man für die resultierende Kurve eine Reihe von Eigenschaften erzwingen:

- Eine Bezierkurve liegt in der konvexen Hülle von höchstens $m+1$ Leitpunkten (wobei m hier wieder der Grad des Basispolynoms ist). So kann man etwa für $m = 2$ erzwingen, daß die Kurve nicht aus einem vorgegebenen Dreieck von Leitpunkten "ausschwingen" kann (Abb. 6.17).
- Wir können gewissen Leitpunkten ein höheres Gewicht erteilen, indem wir sie mehrfach ansetzen. Mehrfachpunkte ziehen die Kurve enger an sich. Ein m -facher Punkt muß auf der resultierenden Kurve liegen. Damit kann man "Ecken" in einer ansonsten glatt verlaufenden Kurve erzeugen (etwa bei Bruchkanten).

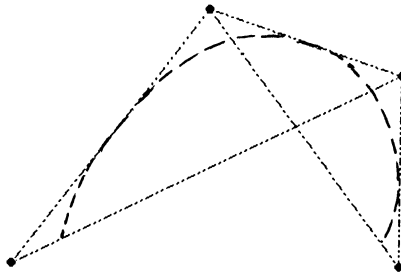


Abb. 6.17. Bezierkurve (Grad m) und konvexe Hüllen von je $m+1$ Leitpunkten

- Liegen $m+1$ aufeinander folgende Leitpunkte auf einer Geraden, so liegt auch die Kurve auf dieser Geraden; dadurch können wir Geradlinigkeitsbedingungen erfüllen.

Es gibt noch eine Reihe weiterer Aussagen, die für spezielle Basisfunktionen und spezielle Leitpunktanordnungen gelten. Wir verweisen für weitere Details auf Pavlidis (1982).

6.5.3. Approximation

Bis jetzt haben wir die Approximation als Verallgemeinerung der Interpolation betrachtet; wir ersetzen die Forderung, daß die resultierende Kurve durch eine Reihe von gegebenen Punkten hindurchgehen soll, durch eine "weniger strenge" Vorschrift, daß sie sich nicht "allzu weit" von diesen Punkten entfernt. Beim interaktiven Arbeiten ist der Unterschied auch nicht so wesentlich. Der Anwender modifiziert die Leitpunkte so lange, bis die Kurve seinen Ansprüchen genügt. Das mag bedeuten, daß sie durch alle Punkte geht, oder durch die meisten, oder in der Nähe aller Punkte verläuft usw.

Ganz anders ist der Sachverhalt bei automatisch ablaufenden Prozessen. Der Begriff "Nähe" muß so formuliert werden, daß er eindeutig und auch überprüfbar ist. Im allgemeinen fordern wir hier ein Minimum der Verbesserungsquadratsumme (Gauß'sches Minimumsprinzip); dabei gelten die Abweichungen der gegebenen Punkte von der resultierenden Kurve als Verbesserungen. Dieses Minimumsprinzip führt zur Aufstellung von Normalgleichungen, die als Lösung die ausgeglichenen Parameter liefern. Nähere Details zur Ausgleichung können Meissl (1982) entnommen werden.

Bisher haben wir die Wahl der Stützstellen nicht in unsere Optimalitätsuntersuchungen einbezogen, obwohl sie natürlich einen entscheidenden Einfluß auf das Interpolationsergebnis hat. Ein allgemeiner Ansatz, der die Anzahl, Lage und Länge der einzelnen Stützintervalle aus den gegebenen Punkten allein aufgrund des Minimumsprinzips abzuleiten versucht, führt auf nichtlineare Gleichungssysteme. Wir müssen uns daher mit **suboptimalen Teillösungen** zufrieden geben.

Als Beispiel nehmen wir eine Gruppe von Punkten, von denen wir annehmen, daß sie - zumindest lokal - einen geradlinigen Verlauf aufweisen; es geht also um die Ermittlung eines Polygons, das möglichst wenige Stützpunkte haben soll. Wir ziehen zwischen dem Anfangspunkt und dem Endpunkt eine Gerade und betrachten die Abweichungen von dieser Geraden. Überschreitet ein Punkt (oder mehrere Punkte) ein bestimmtes vorher festgesetztes Toleranzmaß, so teilen wir das Gesamtintervall an jener Stelle, wo wir die größte Abweichung festgestellt haben, in zwei Teile, und wiederholen die Untersuchung für jedes Teilintervall, so lange, bis alle Punkte die Genauigkeitsschranke erfüllen. Die entsprechende Unterteilung in Stützintervalle ist zwar nicht minimal im strengen Sinne, liefert jedoch im allgemeinen brauchbare Ergebnisse. Ein Extremfall ergibt sich, wenn wir die Schranke auf Null setzen; das resultierende Polygon geht dann durch **alle** gegebenen Punkte.

Zu dieser streng hierarchischen Vorgehensweise gibt es Alternativen: wir beginnen nicht beim Gesamtintervall, sondern bei einer "mittleren" Aufteilung in ca. 5-10 Intervalle. Dann müssen wir aber auch die Möglichkeit einbeziehen, daß die Geraden in zwei aneinandergrenzenden Intervallen dieselbe Neigung aufweisen (natürlich wieder innerhalb einer bestimmten Toleranz); diese sind dann zu vereinigen. Es gibt also sowohl Teilungs- als auch Vereinigungsoperationen. Dieser Vorgang wird deshalb **Split and Merge** genannt.

Eine weitere Alternative besteht darin, daß wir eine Anfangsrichtung, wie sie durch die ersten beiden Punkte gegeben ist, mit jedem weiteren Punkt vergleichen; sobald ein Punkt nicht mehr in den Toleranzbereich fällt, wird eine neue Stützstelle erzeugt, und man setzt mit der neuen Richtung fort (**scan along**). Wieder gibt es hier die Variante, daß man zunächst größere "Sprünge" nach vorn macht und dabei in Kauf nimmt, daß man gegebenenfalls einen Schritt zurück tun muß (**hop along**). Für nähere Erläuterungen zu diesen Verfahren sei auf Pavlidis (1982) verwiesen.

Wir können auch die Abweichung eines Punktes in einen **systematischen** und einen **zufälligen** Aspekt aufteilen und diese beiden Aspekte getrennt behandeln; ein Maß für die Zufälligkeit ist die Häufigkeit des Vorzeichenwechsels

(Abb. 6.18). Ergeben sich über längere Strecken Abweichungen nur nach einer Seite hin, so tritt der systematische Anteil (der Trend) hervor. Dieser Trend wird dann noch zusätzlich von einem rein zufällig oszillierenden Rauschen überlagert.

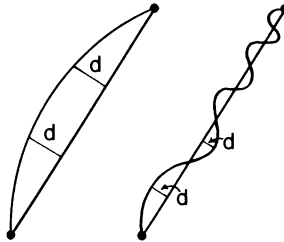


Abb. 6.18. Systematische und zufällige Aspekte bei der Approximation

6.6. Interpolation und Approximation von Flächen

6.6.1. Modellbildung (digitales Geländemodell)

Das Modell einer Fläche im dreidimensionalen Raum ist eine (mehr oder weniger) vereinfachte Abbildung eines Teiles der Wirklichkeit. Der Grad der Vereinfachung hängt vom Spektrum der Anwendungen ab, die auf diesem Modell aufbauen. Die Kugeloberfläche als Modell der Erdoberfläche ist ein Beispiel einer stark vereinfachenden Modellbildung, die in vielen Fällen durchaus brauchbare Ergebnisse liefert. Auch die Verfeinerung des Modelles in Richtung einer ellipsoidischen Oberfläche und das gröbere, punktförmige Erdmodell im interplanetaren Bereich sind uns wohlvertraut.

Im Rahmen der raumbezogenen Informationssysteme sind jedoch differenziertere Modelle für die Beschreibung eines Geländes vonnöten. Es wird im allgemeinen nicht möglich sein, eine einheitliche mathematische Beschreibung (Ebene, Kugel, Polynom etc.) zu finden, welche die Geländeformen wiedergibt. Man behilft sich daher so, daß man diese mathematische Beschreibung zumindest stückweise festsetzt; das Geländemodell gleicht also einem Fleckenteppich, der aus einzelnen wohldefinierten Bausteinen besteht, die nach bestimmten (Interpolations-) Vorschriften zusammengesetzt werden. Diese Interpolationsvorschriften sorgen dann dafür, daß die resultierende Gesamtfläche genügend glatt ist. (Ein anderer Ansatz, der nicht so wie das Bausteinprinzip von unten nach oben, sondern vom Großen ins Kleine geht, ist das

fraktale Modell; für nähere Erläuterungen zu den verschiedenen Möglichkeiten einer topologischen Modellbildung sei der Leser auf Kapitel 2 verwiesen.)

Das Modell einer Geländeoberfläche wird als **digitales Geländemodell** (DGM oder DTM nach dem englischen Ausdruck "digital terrain model") bezeichnet. Manche Autoren verwenden auch den Ausdruck **digitales Höhenmodell** (DHM oder DEM nach dem englischen Ausdruck "digital elevation model"), um damit zum Ausdruck zu bringen, daß es sich primär um eine Bestimmung der dritten Koordinate eines Punktes handelt; die Lage eines Punktes rechnen diese Autoren dem Geländemodell zu. Wir wollen aber nicht näher auf diese unterschiedlichen Auffassungen eingehen und im folgenden die Kurzbezeichnung "DGM" verwenden. Es sei hier aber auch erwähnt, daß in raumbezogenen Informationssystemen auch noch viele andere Themen mit denselben Werkzeugen modelliert werden können: für Temperaturverteilungen, Geoidhöhen, infrastrukturelle Gegebenheiten usw. lassen sich ebenfalls Modelle erzeugen, in denen die "Höhe" dem jeweiligen Thema entspricht.

Die Bausteine eines DGM können **Punkte, Linien** oder **Flächen** sein (Abb. 6.19). Im ersten Fall handelt es sich um Koordinaten x , y , z von gewissen ausgezeichneten Punkten. Die z -Koordinaten dazwischenliegender Punkte ergeben sich durch eine Interpolationsvorschrift. Im Linienmodell dienen Raumkurven als Grundbausteine; beispielsweise können dies Schichtenlinien sein; dazwischenliegende Punkte erhalten wir durch Interpolation dieser Linien. Dieses Konzept wird eher im CAD-Bereich angewendet; die Flächen, die man dort zwischen gegebene Kurven einhängt, werden als **lofted surfaces** bezeichnet. Schließlich ist es auch möglich, kleine Flächenstücke als Bausteine zu verwenden; sind die Parameter eines solchen Flächenstückes bekannt, so können wir einen beliebigen Flächenpunkt durch einfaches Einsetzen ermitteln; meistens wird die Fläche jedoch selbst wieder durch charakteristische Punkte repräsentiert, die dann durch eine vorgegebene Interpolationsvorschrift verknüpft werden.

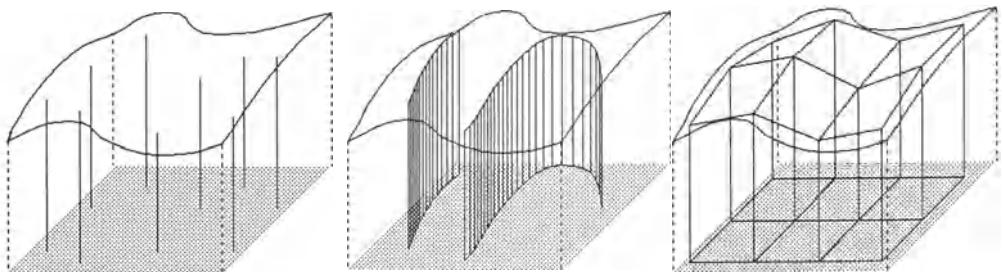


Abb. 6.19. Punkte, Linien und Flächen als Bausteine eines DGM

In der Praxis verschwimmen die Grenzen zwischen punktbezogenen, linienbezogenen und flächenbezogenen DGM. Sie verschwinden nahezu, wenn die Punkte in einem Gitter angeordnet sind. Aber auch sonst können wir jederzeit durch Interpolation bzw. Approximation von einer Kategorie zur anderen übergehen. Die dazu notwendigen Hilfsmittel untersuchen wir weiter unten.

Wie wir eben erwähnt haben, liegen die Punkte eines DGM oft in einer regelmäßigen Anordnung (Gitter) vor. Eine Alternative dazu bietet sich, wenn wir sie nach morphologischen Gesichtspunkten auswählen (Abb. 6.20).

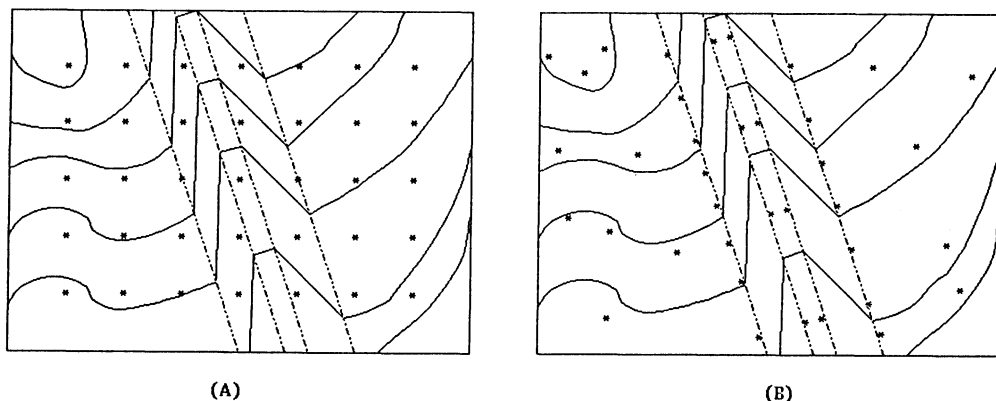


Abb. 6.20. Regelmäßige und morphologische Anordnung von DGM-Punkten

Die Anordnung hängt zunächst von der Erfassungsart ab. Wurden die Punkthöhen manuell bzw. tachymetrisch erfaßt, so sind sie meist unregelmäßig verteilt. Dasselbe gilt auch beim Nachfahren von Schichtenlinien in Stereomodellen. Wird der Erfassungsprozeß aber automatisiert, so geschieht dies meist in der Art, daß man im Stereomodell oder im Orthophoto entlang einer Koordinatenrichtung bzw. entlang eines Profiles in konstanten Abständen Werte ermittelt.

Im Kapitel 2 haben wir uns ausführlich mit dem Gegensatz zwischen Vektor- und Rastermodellen und der zugehörigen Methodik befaßt. Die regelmäßige Erfassung und Anordnung von DGM-Punkten entspringt dem Rastergedanken. Wir haben ein dichtes Netz von Punkten vorliegen; um gewisse charakteristische Eigenschaften des Geländes (wie etwa Bruchkanten, lokale Maxima und Minima) berücksichtigen zu können, müssen wir global eine sehr kleine Maschenweite wählen, die in den weniger stark fluktuierenden Bereichen einen unnötig hohen Speicherplatzbedarf nach sich zieht. Dafür werden aber viele Auswertungen einfach; Algorithmen,

die sich auf regelmäßig verteilte Daten stützen können, sind zahlreich und leicht durchschaubar.

Die morphologische Anordnung der Punkte hingegen entspricht dem Vektormodell; sie zielt auf eine Minimierung der Gesamtpunktezahl hin, wobei gleichzeitig eine möglichst gute Abbildung des Geländes und seiner kritischen Stellen - etwa Bruchkanten (Abb. 6.21) - erreicht werden soll; dafür zahlt man den Preis, daß die Daten unregelmäßig verteilt sind.

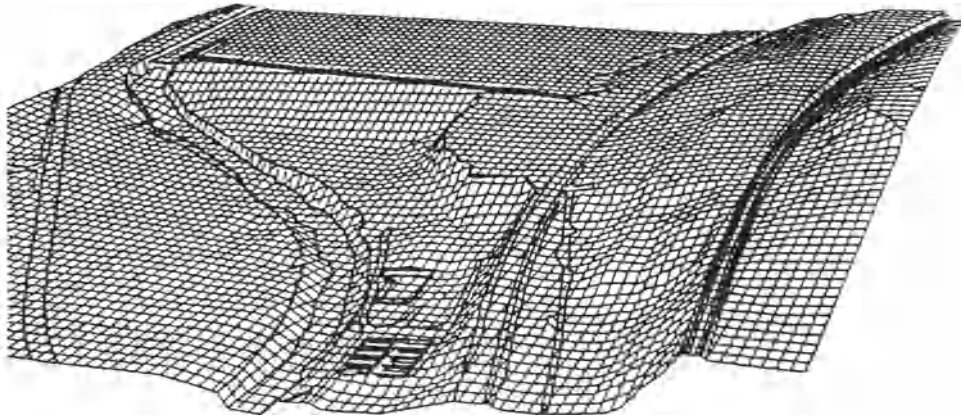


Abb. 6.21. Beispiel eines DGM mit Bruchkanten

Die Art der Erfassung bindet uns jedoch nicht an ein bestimmtes Modell; so können wir aus unregelmäßig verteilten Daten (**Primärdaten**) ein regelmäßiges, feinmaschigeres Gitter (**Sekundärdaten**) erzeugen; nach diesem Vorverarbeitungsschritt greifen alle Anwendungsprogramme dann auf diese Sekundärdaten zu. Für die Erzeugung des Sekundärdatenbestandes können wir die weiter unten beschriebenen Hilfsmittel der Interpolation verwenden. Haben wir umgekehrt ein regelmäßiges Gitter vorliegen, so können wir im nachhinein vektorhafte Informationen (Bruchlinien etc.) einbringen; die meisten DGM-Softwarepakete erlauben also eine solche hybride Verschmelzung der beiden Ideen.

Aus dem Linienmodell - sei es das primäre, durch Nachfahren von Schichtenlinien entstandene Modell, oder sei es ein durch Interpolation entstandenes sekundäres Modell - läßt sich eine interessante Modellvariante erzeugen, die dem Rastergedanken entspricht: das Höhenschichtenmodell (Abb. 6.22). Wenn wir allen jenen Rasterzellen, die zwischen zwei vorgegebenen Schichtenlinien liegen, die Höhe der jeweils niedrigeren Linie zuordnen, so ergeben sich Schichten: die unterste Schicht entspricht der minimalen Höhe im betrachteten Gebiet und überdeckt somit das gesamte

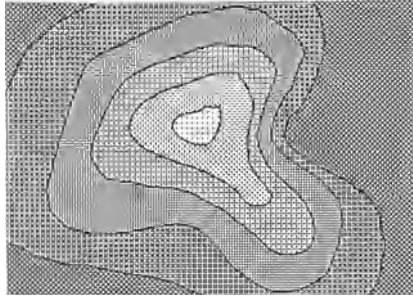


Abb. 6.22. Höhengschichtenmodell

Gebiet. Jede weitere Schicht überdeckt Teile der unter ihr liegenden Schicht. Alle Schichten haben eine konstante vertikale Dicke. Eine Schicht kann nie über die darunterliegende Schicht hinausragen (dann würden sich die entsprechenden Schichtenlinien schneiden); wohl aber kann sie durchlöchert sein, etwa im Fall einer Schottergrube; in dieser Grube kann selbst wieder eine "Insel" auftreten usw. Das Höhengschichtenmodell unterstützt (wie alle Rastermodelle) besonders einfache Methoden der geometrisch-topologischen Auswertung. Auch ist der Übergang zu einem Volumenmodell einfach; dieses ist aus quaderförmigen Zellen (sogenannten "voxels") aufgebaut und erlaubt auf einfache Weise Massenberechnungen und -bilanzen. Für nähere Erläuterungen dazu sei auf Niese u. Weber (1987) verwiesen.

6.6.2. Lineare und bilineare Interpolation

Wir ordnen jeder Punktlage x, y einen Wert z zu, der in erster Linie als Geländehöhe aufzufassen ist, aber auch verallgemeinerte Deutungen (Temperatur, Lärmpegel usw., siehe Abschnitt 6.6.1) zulässt. In einem punktbezogenen Modell sind die z -Werte gewisser charakteristischer Punkte - seien diese regelmäßig oder unregelmäßig verteilt - bekannt. Durch Interpolation ermitteln wir die z -Werte der dazwischenliegenden Punkte. Der einfachste Fall einer solchen Interpolation ergibt sich, wenn wir drei Punkte P, Q, R vorgegeben haben und dazwischen linear interpolieren (Abb. 6.23). Die Dreiecksfläche ist dann eine Ebene; sie wird durch folgende Formel repräsentiert:

$$T(u, v) = u * P + v * Q + (1-u-v) * R \quad (6-6-1)$$

P, Q und R sind dabei als Vektoren $(x_p, y_p, z_p), (x_q, y_q, z_q)$ und (x_r, y_r, z_r) zu interpretieren, deren

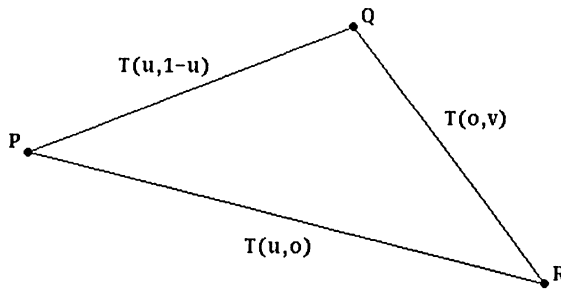


Abb. 6.23. Lineare Interpolation zwischen drei Punkten

Elemente bekannt sind. Die skalaren Größen u und v liegen im Bereich zwischen 0 und 1, wobei die Summe $u+v$ den Wert 1 nicht überschreiten darf. Variieren nun u und v im erlaubten Bereich, so überstreicht der Vektor T , bestehend aus den Werten (x_t, y_t, z_t) , genau jenen Teil der Ebene, der von den drei Punkten P, Q und R und deren Verbindungsgeraden eingeschlossen wird. Die Formel besteht also aus drei Gleichungen für die drei Komponenten des Vektors T , die von den zwei Parametern u und v abhängen.

$$\begin{aligned} x_t(u,v) &= u * x_p + v * x_q + (1-u-v) * x_r \\ y_t(u,v) &= u * y_p + v * y_q + (1-u-v) * y_r \\ z_t(u,v) &= u * z_p + v * z_q + (1-u-v) * z_r \end{aligned} \quad (6-6-2)$$

Man sieht sofort, daß die drei gegebenen Punkte P, Q, R durch eine spezielle Wahl der Parameter u, v reproduziert werden können:

$$\begin{aligned} P &= T(1,0) \\ Q &= T(0,1) \\ R &= T(0,0) \end{aligned} \quad (6-6-3)$$

Außerdem können wir die Gleichungen der das Ebenenstück begrenzenden Geraden bilden, indem wir jeweils einen Flächenparameter eliminieren:

$$\begin{aligned} \text{Gerade zwischen P und R} &\dots T(u, 0) \\ \text{Gerade zwischen Q und R} &\dots T(0, v) \\ \text{Gerade zwischen P und Q} &\dots T(u, 1-u) \end{aligned} \quad (6-6-4)$$

Wir sind nun am z -Wert eines konkreten Punktes T interessiert, dessen Koordinaten x_t und y_t wir vorgeben. Wir formen daher die Formel so um, daß die dritte Komponente z_t des Vektors T explizit durch die Komponenten x_t und y_t ausgedrückt wird und haben damit die Interpolationsvorschrift gefunden (sie ist zugleich die Gleichung der Ebene in expliziter Form):

$$z_t = a_0 + a_1 * x_t + a_2 * y_t \quad (6-6-5)$$

In den Konstanten a_0 , a_1 und a_2 stehen im wesentlichen Produkte von Koordinatendifferenzen der vorgegebenen Punkte P , Q und R . Da P , Q und R spezielle Punkte dieser Ebene sind, müssen auch ihre Koordinaten dieser Gleichung genügen; es gilt daher

$$\begin{aligned} a_0 + a_1 * x_p + a_2 * y_p &= z_p \\ a_0 + a_1 * x_q + a_2 * y_q &= z_q \\ a_0 + a_1 * x_r + a_2 * y_r &= z_r \end{aligned} \quad (6-6-6)$$

oder in Matrixschreibweise

$$AX = L, \quad (6-6-7)$$

wobei die Matrix A und der Vektor L im wesentlichen durch die Koordinatendifferenzen der Dreieckspunkte gegeben sind und X der Vektor der Koeffizienten a_0 , a_1 , a_2 der Ebene ist. Die oben beschriebenen Rechenoperationen sind also gleichwertig mit einer Inversion dieser Gleichung $AX = L$. Haben wir diese Koeffizienten berechnet, so können wir die z -Koordinate eines beliebigen, durch x und y gegebenen Punktes durch Einsetzen in die obige Ebenengleichung ermitteln.

Haben wir nicht drei, sondern vier Punkte gegeben, zwischen denen wir interpolieren wollen - wie dies etwa bei einem regelmäßigen Gitter der Fall ist, so gibt es zunächst die Möglichkeit, aus dem Viereck zwei Dreiecke zu erzeugen, und dann so zu verfahren wie bisher. Als Alternative dazu kommt die **bilineare Interpolation** in Betracht (Abb. 6.24), die jedoch im allgemeinen keine Ebene mehr erzeugt, sondern eine Fläche 2. Ordnung, ein hyperbolisches Paraboloid. Vier Punkte P , Q , R , S geben in diesem Fall Anlaß zu einer Formel

$$B(u,v) = (1-u) * (1-v) * P + (1-u) * v * Q + u * (1-v) * R + u * v * S \quad (6-6-8)$$

wobei hier sowohl u als auch v zwischen 0 und 1 variieren dürfen. Wieder handelt es sich um drei Gleichungen, in denen zwei Parameter u und v auftreten, und wieder können wir durch eine spezielle Wahl dieser Parameter die gegebenen Punkte sowie deren Verbindungslinien reproduzieren:

$$\begin{aligned} P &= B(0,0) \\ Q &= B(0,1) \\ R &= B(1,0) \\ S &= B(1,1) \end{aligned} \quad (6-6-9)$$

$$\begin{aligned} \text{Gerade zwischen } P \text{ und } Q &\dots B(0,v) \\ \text{Gerade zwischen } P \text{ und } R &\dots B(u,0) \\ \text{Gerade zwischen } Q \text{ und } S &\dots B(u,1) \\ \text{Gerade zwischen } R \text{ und } S &\dots B(1,v) \end{aligned} \quad (6-6-10)$$

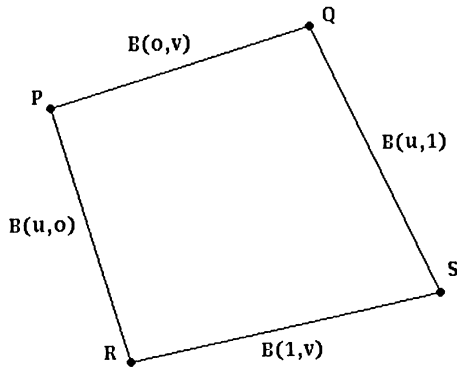


Abb. 6.24. Bilineare Interpolation zwischen vier Punkten

Um zu einer expliziten Darstellung von z_b zu kommen, schreiben wir die obige Interpolationsformel um:

$$B-P = (R-P) * u + (Q-P) * v + (P-Q-R+S) * uv \quad (6-6-11)$$

Wenn der Grundriß der Masche P, Q, R, S ein achsparalleles Rechteck mit Seitenlängen dx und dy ist, so vereinfachen sich die ersten beiden Gleichungen dieser Formel zu:

$$\begin{aligned} v &= (x_b - x_p) / dx \\ u &= (y_b - y_p) / dy \end{aligned} \quad (6-6-12)$$

Dies setzen wir in die dritte Gleichung ein und erhalten einen Ausdruck für z_b - und somit die gesuchte Interpolationsvorschrift - in der folgenden Form:

$$z_b = a_0 + a_1 * x_b + a_2 * y_b + a_3 * x_b * y_b \quad (6-6-13)$$

oder $AX = L$, wobei X der Vektor der Koeffizienten a_0, a_1, a_2, a_3 des Flächenstückes ist. Die Matrix A enthält im wesentlichen - so wie beim früheren Fall - Koordinatendifferenzen der bekannten Punkte P, Q, R, S.

Beide bisher beschriebenen Verfahren bergen den Nachteil in sich, daß zwei aneinanderstoßende Flächenstücke im allgemeinen nicht glatt ineinander übergehen. Um dies zu vermeiden, können wir die Umgebung, die zur Berechnung einer solchen elementaren Teilfläche herangezogen wird, vergrößern, und eine entsprechende stetige Fläche höherer Ordnung durch diese Punkte hindurchlegen. Der Wert eines vorgegebenen Punktes beeinflusst dann mehrere solcher

Teilflächen, und diese überlappen einander. Dadurch werden die Knicke zwischen aneinandergrenzenden Teilflächen gemildert. (Ein - wenn auch nur theoretisch interessanter - Extremfall ergibt sich, wenn wir die Umgebung auf das gesamte Modell ausdehnen.)

Anstatt eine Fläche entsprechend hoher Ordnung *exakt* durch n Punkte hindurchzulegen, können wir auch eine etwaige Überbestimmung im Rahmen einer vermittelnden Ausgleichung ausnützen. Unsere Formel $AX = L$ verallgemeinert sich dementsprechend zu den Beobachtungsgleichungen eines vermittelnden Ausgleichsproblems, aus denen wir durch Bildung der Normalgleichungen und deren Auflösung die "besten" Werte für die Flächenparameter a_0, a_1, a_2 usw. ermitteln. Für Einzelheiten zur vermittelnden Ausgleichung sei auf Meißl (1982) verwiesen.

Als zusätzliche "Beobachtungen" führt man manchmal auch noch die Tangenten an die Fläche in den gegebenen Punkten bzw. Nahtstellen sowie die dort auftretenden Krümmungen ein. Für die Werte dieser zusätzlichen Beobachtungen verwendet man Höhen- und Koordinatendifferenzen. Bruchlinien, d.h. solche Linien, an denen wir einen glatten Übergang explizit ausschließen möchten, können so leicht berücksichtigt werden, indem wir die entsprechenden Tangenten- und Krümmungsbeobachtungen weglassen.

6.6.3. Gleitende Mittelbildung

Eine andere Methode der Interpolation zwischen bestehenden Datenpunkten besteht darin, daß man erst gar nicht versucht, mathematisch beschreibbare Teilflächen zu erzeugen. Vielmehr berechnet man sie **dynamisch** aus den Informationen, die in einer gewissen Umgebung vorliegen. Man spricht vom **Gleitenden Mittel (moving average)**. Diese Interpolationsmethode benützt für die Berechnung eines Flächenpunktes die nächstliegenden n Datenpunkte (n umfaßt manchmal den gesamten Datenbestand), wobei ein gewichtetes Mittel der an diesen Datenpunkten vorliegenden Werte berechnet wird (Abb. 6.25):

$$z = (p_1 * x_1 + \dots + p_n * x_n) / (p_1 + \dots + p_n) \quad (6-6-14)$$

Das Gewicht p_i des Wertes für den i -ten Datenpunkt ergibt sich aus dessen reziproker Distanz von jenem Punkt, für den man den z -Wert berechnen will. Manchmal werden auch höhere Potenzen dieser Distanz verwendet. Siehe dazu auch Konecny (1981).

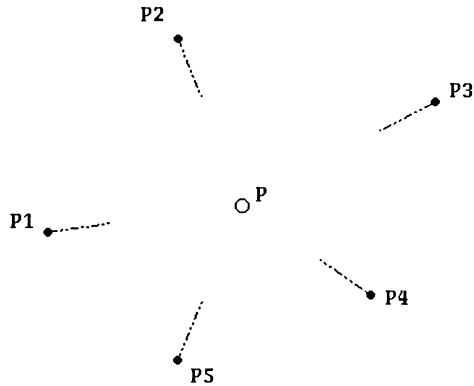


Abb. 6.25. Interpolation durch gleitende Mittelbildung

Eine andere Möglichkeit der Gewichtung ergibt sich, wenn wir eine **Kovarianzfunktion** zugrunde legen, die immer stärker abklingt, je weiter man sich vom fraglichen Punkt entfernt (Abb. 6.26). Die Kovarianzfunktion prägt natürlich das Interpolationsergebnis; bei ihrer Festsetzung müssen wir die Beschaffenheit der zu ermittelnden Fläche berücksichtigen. Oft wird sie empirisch anhand charakteristischer "Trainingsgebiete" festgesetzt; sie kann aber auch - falls die Anzahl vorhandener Datenpunkte sehr groß ist - aus den "Beobachtungen", also den gegebenen Werten bzw. deren Differenzen, berechnet werden. Sie ähnelt meist der Gauß'schen Glockenkurve; je breiter sie ist, desto weitreichender ist der Einfluß eines gegebenen Wertes, und desto stärker ist die resultierende Glättung. Es ist aber zu bedenken, daß eine Glättung nicht immer nur positive Aspekte hat. Es ist mitunter auch wünschenswert, gerade die Unregelmäßigkeiten eines Erscheinungsbildes stärker hervorzuheben; hier sei auf Abschnitt 6.4 und die dort näher beschriebenen Generalisierungsmethoden verwiesen.

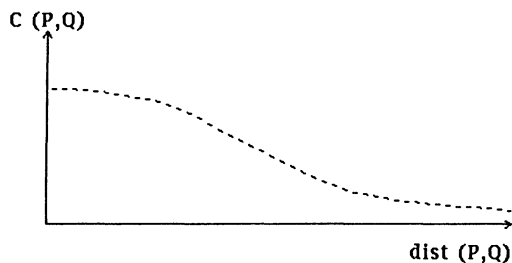


Abb. 6.26. Beispiel für eine Kovarianzfunktion

Schließlich können wir den Interpolationsansatz auch noch verfeinern, indem wir beispielsweise eine lineare **Trendfunktion** abspalten und nur mehr die verbleibenden Klaffungen ausgleichen. Damit sind wir schon beim Ansatz der **Kollokation** bzw. **linearen Prädiktion**, der eine Verallgemeinerung des vermittelnden Ausgleichs darstellt.

6.6.4. Lineare Prädiktion

Wir wollen diesen Ansatz hier kurz erläutern. Näheres kann man der Darstellung von Kraus u. Mikhail (1972) entnehmen.

Wir spalten die "Beobachtung" H - in unserem Fall den Vektor der an den Primär- bzw. Sekundärpunkten gegebenen Höhen - in zwei Bestandteile S und R auf: S bezeichnet das *Signal*, während R das *Rauschen* darstellt. R übernimmt die Rolle eines zufälligen Beobachtungsfehlers, ist also eine Zufallsvariable. S hingegen ist (im Rahmen unserer Anwendungen) meist deterministisch; wenn wir ein Gelände mehrmals "beobachten" und dabei keinen Meßfehler machen, so erhalten wir stets dasselbe Ergebnis. Trotzdem erweist es sich als nützlich, S (zumindest formal) ebenfalls statistisch zu behandeln. Eine fundierte Begründung für die Berechtigung einer solchen Vorgehensweise können wir hier nicht geben; es genügt uns, seine Plausibilität durch folgende Überlegung klarzulegen: wenn wir das Modell wiederholt an *verschiedenen, benachbarten Stellen* aufbauen, so ändert sich S "zufällig". Die S -Bestandteile verschiedener Messungen sind dabei "korreliert"; legt man durch sie eine ausgleichende Fläche, so befinden sich benachbarte Punkte mit großer Wahrscheinlichkeit auf derselben Seite dieser Fläche. Bei den R -Bestandteilen hingegen sind Punkte, auch wenn sie noch so nahe beisammenliegen, keiner solchen Einschränkung unterworfen.

Sowohl S als auch R werden also als **Zufallsvariable** aufgefaßt, von denen wir annehmen, daß sie voneinander unabhängig sind; ferner sei ihr Erwartungswert gleich Null. Dies bedeutet, daß wir unsere Meßwerte in die Fläche des Signals verschieben müssen; dazu können wir die vorhin beschriebenen Methoden benutzen; wir arbeiten also derzeit an einer Verfeinerung dieser Methoden.

Wollen wir nun die Höhe h an einem (Neu-)Punkt durch eine lineare Funktion der Höhen h_1, \dots, h_n an n gegebenen Punkten schätzen, so ergibt sich das Problem, die Koeffizienten a_1, \dots, a_n zu ermitteln.

$$h = a_1 * h_1 + a_2 * h_2 + \dots + a_n * h_n \quad (6-6-15)$$

Wir sprechen in diesem Fall auch von linearer Prädiktion.

Die beste Schätzung ergibt sich aus

$$h = c^T (C+D)^{-1} H \quad (6-6-16)$$

Dabei ist H der Vektor der vorgegebenen Höhen, C ist die Matrix der Kovarianzen der jeweiligen vorgegebenen Höhen, D ist die (im allgemeinen als diagonal angenommene) Kovarianzmatrix der R-Bestandteile, und c ist der Vektor der Kovarianzen zwischen dem Neupunkt und den bestehenden Punkten. Natürlich sind die Matrizen C und D bzw. der Vektor c zunächst noch gar nicht bekannt. Anstelle der Matrix C können wir auch eine Kovarianzfunktion $C(d)$ verwenden, die als Funktion der Distanz zwischen zwei Punkten auftritt (siehe auch Abb. 6.26). Diese wiederum können wir empirisch so ermitteln, daß wir für jedes Distanzintervall alle möglichen Kombinationen von Punktepaaren i, j bestimmen, deren Entfernung in dieses Intervall hineinfällt, und die entsprechenden Produkte $h_i * h_j$ mitteln. Auf ähnliche Weise können wir die Kovarianzmatrix C sowie den Vektor c bestimmen. Für die Diagonalmatrix D wird meist eine a priori-Wertung vorgenommen.

Da in diesem Ansatz für die Prädiktion eines Höhenwertes die Inversion einer großen vollbesetzten Matrix erforderlich ist, sucht man, das Verfahren zu modifizieren. Sünkel (1981) nützt den Umstand aus, daß ein regelmäßiges Gitter im allgemeinen viel feiner als das Netz der Primärdaten ist, und daß sich die Kovarianzmatrix lokal nur wenig ändert, wenn man von einem Gitterpunkt zum nächsten übergeht. Man muß also beim Übergang zum Nachbarpunkt nur einen kleinen Teil der Inversen neu berechnen. Dieser Vorgang wird als **Moving Inverse** bezeichnet.

6.6.5. Interpolation über finite Elemente

Diese Methode geht auf Ebner (1979) zurück. Man will - simultan - von den Werten an m beliebig angeordneten Primärpunkten auf die Werte an n gitterförmig angelegten Sekundärpunkten übergehen. Dabei ermittelt man für jeden Primärpunkt P die Masche, also die Sekundärpunkte (i, j) , in die er hineinfällt, und setzt seinen (gegebenen) Wert als bilineare Kombination der (unbekannten) Werte an den vier Sekundärpunkten an (Abb. 6.27):

$$\begin{aligned} vp &= (1 - dx) * (1 - dy) * z(i, j) \\ &+ dx * (1 - dy) * z(i+1, j) \\ &+ (1 - dx) * dy * z(i, j+1) \\ &+ dx * dy * z(i+1, j+1) - zp \end{aligned} \quad (6-6-17)$$

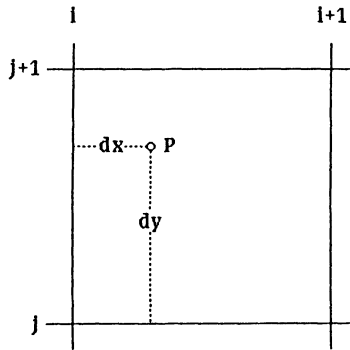


Abb. 6.27. Interpolation über finite Elemente

Diese Gleichungen bezeichnet man als "Beobachtungsgleichungen": die "beobachtete" (also gegebene) Höhe im Punkt z_p stimmt bis auf eine "Verbesserung" v_p mit dem Ergebnis der bilinearen Kombination überein. Das System von Beobachtungsgleichungen ist in dieser Form noch nicht lösbar, da es mehr Unbekannte als Messungen hat. Wir führen also noch für jeden Gitterpunkt zusätzliche Verbesserungsgleichungen ein, welche die Krümmungen in diesem Punkt betreffen:

$$\begin{aligned} v(1,i,j) &= z(i-1,j) - 2z(i,j) + z(i+1,j) - 0 \\ v(2,i,j) &= z(i,j-1) - 2z(i,j) + z(i,j+1) - 0 \end{aligned}$$

... (6-6-18)

Diesen zusätzlichen Verbesserungen ordnen wir Gewichte zu, mit denen wir die Krümmungseigenschaften der Fläche steuern können. Das resultierende System von Beobachtungsgleichungen lösen wir dann nach dem Schema eines vermittelnden Ausgleichsproblems. Da jede Beobachtung nur mit einer beschränkten Zahl von Unbekannten verbunden ist, hat das Normalgleichungssystem eine Bandstruktur, die sich günstig auf die numerische Effizienz des Verfahrens auswirkt.

Auch diese Methode läßt sich beliebig verallgemeinern; etwa, indem man anstatt des bilinearen Ansatzes bikubische Splines verwendet. Wir brauchen dazu für jede "Beobachtung" 16 angrenzende Maschen. Indem wir die Stetigkeit der ersten und zweiten Ableitung beim Übergang von einer Masche in die Nachbarmasche fordern, ergibt sich eine glatte Fläche.

6.6.6. Dreiecksvermaschung

In Abschnitt 6.6.1 haben wir punktbezogene, linienbezogene und flächenbezogene Modelle einander gegenübergestellt. Wir haben gesehen, daß die Unterschiede zwischen diesen Modellen verschwimmen, wenn wir regelmäßig verteilte Daten zugrunde legen; außerdem haben wir Methoden kennengelernt, die aus unregelmäßig verteilten Primärdaten Sekundärdaten in Gitterform erzeugen. Wir haben aber auch herausgefunden, daß Gitterdaten Nachteile haben; sie beanspruchen viel Speicherplatz und können lokal charakteristische Geländeformen nicht so gut modellieren wie Primärdaten.

Wenn wir also für gewisse Anwendungen auf die Einführung solcher regelmäßiger Sekundärdaten verzichten wollen, so ist es notwendig, die Primärdaten zu *vermaschen*. Wie kommen wir aber zu einer passenden Dreiecksvermaschung, das heißt, welche Punkte nehmen wir als benachbart an, und vor allem, wie können wir diesen Prozeß automatisieren? Während das Problem bei drei Ausgangspunkten trivial ist, können wir bei vier Ausgangspunkten bereits zwischen zwei Lösungen wählen, je nachdem, welche Diagonale des Viereckes wir als Dreiecksseite nehmen (Abb. 6.28).

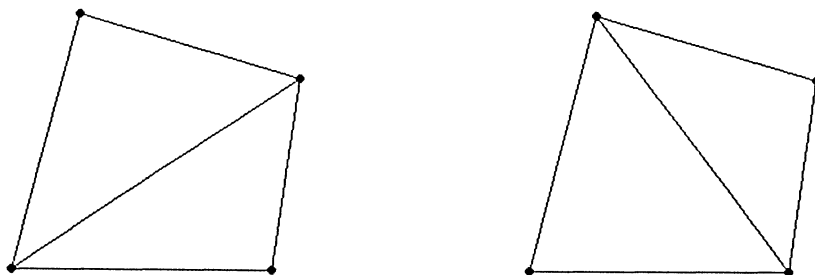


Abb. 6.28. Varianten bei der Dreiecksvermaschung von vier Punkten

Je mehr Punkte vorhanden sind, desto größer wird die Anzahl der Möglichkeiten. Man kann sich vorstellen, daß es auch darauf ankommt, wo wir mit der Vermaschung beginnen, und in welcher Reihenfolge wir die Punkte bearbeiten. Wir müssen also Strategien entwickeln, wie wir bei der Vermaschung vorgehen.

Wir beginnen mit einem Dreieck - also einem minimalen Dreiecksnetz - und fügen sukzessive neue Punkte ein. Dabei können wir am Rande des Bearbeitungsgebietes beginnen und immer weiter ins Innere vordringen; oder wir bilden ein umschreibendes Dreieck (bzw. ein Rechteck, das wir in zwei Dreiecke aufspalten), und gehen hierarchisch von großen Dreiecken zu kleineren über.

Eine Schlüsselstellung im Algorithmus hat die Beantwortung der Frage, ob ein neu hinzukommender Punkt innerhalb oder außerhalb des bestehenden Netzes liegt. Liegt er innerhalb, so kann das Dreieck, das ihn überdeckt, in drei Teildreiecke aufgespalten werden; liegt er außerhalb, muß ein Dreieck hinzugefügt werden. Liegt er "in der Nähe" eines bestehenden Punktes oder einer bestehenden Kante, so muß dies in geeigneter Weise berücksichtigt werden.

Wir können uns leicht vorstellen, daß bei einer ungünstigen Anordnung bzw. Reihenfolge der Punkte sehr viele schmale, langgezogene und spitzwinkelige Dreiecke entstehen. Man versucht daher, die Auswahl der Punkte zu steuern, indem man Nebenbedingungen angibt, die von den Dreiecken erfüllt werden müssen: so kann man etwa aus mehreren möglichen Varianten jene wählen, die am ehesten einem gleichseitigen Dreieck entspricht. Man erzeugt also nach Möglichkeit Winkel in der Nähe von 60 Grad; oder man zieht den Umfang oder den Flächeninhalt als Kriterium heran, indem man etwa das Einfügen neuer Dreiecke verbietet, deren Fläche ein bestimmtes Mindestmaß unterschreitet. Manchmal versucht man auch, die Gesamtlänge aller Kanten zu minimieren, die Streuung der Winkel in einem Dreieck möglichst gering zu halten, oder in allen Punkten denselben topologischen Knotengrad zu erzeugen.

Natürlich bleiben durch diese Auslese gewisse Punkte übrig, die nicht verarbeitet werden konnten. In einer weiteren Iterationsphase versucht man nun diese Punkte - gegebenenfalls unter gelockerten Bedingungen - zu integrieren, und wiederholt dies so lange, bis alle Punkte eingefügt sind.

Meist folgt auf die **Erstellungsphase** eine **Optimierungs-** bzw. **Editierungsphase**. So versucht man etwa, zwei benachbarte Dreiecke zu einem Viereck zusammenzufassen, und dieses Viereck jeweils entlang der anderen Diagonale wieder aufzutrennen. Durch dieses **Umlegen** von Kanten gelingt es oft, eine lokale Verbesserung der Netzeigenschaften zu erreichen. Wieder entscheiden hier die oben erwähnten Optimalitätskriterien mit. In dieser Phase des Prozesses kann man auch Bedingungen angeben, die eingehalten werden sollen, so etwa die Krümmung entlang einer Kante.

Eine gute Einführung in die Problematik der Dreiecksvermaschung bietet Kropatsch (1983).

6.6.7. Interpolation von Schichtenlinien in einem DGM

Wir haben in Abschnitt 6.6.1 angemerkt, daß der Verlauf von Schichtenlinien im Rahmen der Eingabe für ein Geländemodell verwendet werden kann. Umgekehrt wiederum besteht die Notwendigkeit, aus einem gegebenen Modell Schichtenlinien zu extrahieren. Wir müssen dabei zwei Probleme lösen:

- a) wir müssen (zumindest theoretisch) jede mögliche Schichtenlinie mit jeder Masche des Modelles vergleichen und gegebenenfalls die Schnittpunkte ermitteln.
- b) wir müssen die Schnittpunkte durch glatte Kurven miteinander verbinden; der Verlauf der Kurven ist noch durch den Umstand eingeschränkt, daß Schichtenlinien einander nicht schneiden dürfen, und daß etwaige Bruchkanten berücksichtigt werden müssen.

Die Beantwortung von a) ist zunächst einfach; wir kennen für jede Masche das Minimum und das Maximum der Höhen an den Eckpunkten. Nur jene Schichtenlinien kommen in Frage, die in dieses Intervall hineinfallen. Durch lineare Interpolation finden wir die Schnittpunkte an den Kanten (dies gilt allerdings nur dann streng, wenn wir eine Ebene oder bilineare Fläche in dieser Masche vorfinden). Auf der Umgrenzung einer solchen Masche muß es natürlich eine gerade Anzahl von Schnittpunkten geben: Schichtenlinien dürfen weder in dieser Masche "verschwinden" noch dort "entstehen". Wenn es mehr als zwei Schnittpunkte gibt, müssen wir das topologische Problem lösen, welche Schnittpunkte miteinander zu verbinden sind. Dies kann durch einfachen Vergleich des Vorzeichens der Differenzen zu den benachbarten Eckpunkten geschehen. Wir merken hier an, daß Schichtenlinien, die sich innerhalb einer Masche schließen, bei dieser Vorgehensweise unerkant bleiben.

Für die Beantwortung der Frage b) verweisen wir auf die Abschnitte 6.5.1 - 6.5.3, in denen die Interpolation von Kurven besprochen wird. Wir wollen hier nur erwähnen, daß der Umstand, daß Schichtenlinien einander nicht schneiden dürfen, in gewissen Fällen zum Problem werden kann. Man behilft sich dadurch, daß man einen Glättungsfaktor wählt, den man dann - im Falle, daß eine Überschneidung auftritt - in einem interaktiven Editierungsprozeß lokal vermindert. Demgegenüber steht ein theoretisch sauberer Ansatz über Bezier-Polynome, welche die Eigenschaft besitzen, daß die resultierende Kurve in der konvexen Hülle von Stütz- und Führungspunkten liegt (Kropatsch 1983).

6.7. Generalisierung

Die Generalisierung ist eine Maßnahme, die sowohl bei der erstmaligen Erstellung einer Karte wie auch bei der Überführung einer gegebenen Karte in einen kleineren Maßstab notwendig wird. Die Karte als Abbildung der uns umgebenden Realität kann nur die "wesentlichen" Teile dieser Realität wiedergeben. Wir haben auch in den Kapiteln 2 und 3 detailliert zur Modellbildung Stellung genommen. Das Modell des Antlitzes unserer Erde muß natürlich eine grobe Verallgemeinerung sein, auch wenn wir nur an einem winzig kleinen Teil dieses Antlitzes interessiert sind, und auch wenn wir hochpräzise Meßmittel zur Verfügung haben. Aus diesem Grund wird die Kartenerstellung *immer* von Generalisierungseffekten beeinflusst. Sogar die Darstellung der Grenzen amerikanischer Bundesstaaten baut auf einem Modell auf, das die tatsächliche Erdoberfläche nur unvollkommen wiedergeben kann.

Das Ziel einer Generalisierung ist es, den Informationsgehalt von darzustellenden Objekten möglichst wenig, aber doch so weit herabzusetzen, daß die Interpretierbarkeit der Karte gewährleistet ist. Die Begriffe, die wir in dieser Definition verwendet haben, sind natürlich sehr allgemein und können individuell sehr verschieden interpretiert werden. Tatsächlich gehört die Generalisierung in der Kartographie zu jenen Methoden, die dem Kartographen die größte Erfahrung abverlangen und ein gerüttelt Maß an Fingerspitzengefühl voraussetzen. Es ist daher nicht verwunderlich, daß sie im Bereich der digitalen Kartographie eine jener Methoden ist, die sich am erfolgreichsten gegen eine Automatisierung wehren.

Trotzdem dürfen wir nicht vor dem Versuch zurückscheuen, Teilaspekte der Generalisierung einer computerunterstützten Lösung zugänglich zu machen. Denn die Wirtschaftlichkeit eines Informationssystems ist erst dann gegeben, wenn digital gespeicherte Informationen in vielerlei Hinsicht kombinierbar sind und in verschiedensten Auswertungen spürbar werden. Methoden, die den Übergang von einem Maßstab in einen anderen, kleineren, Maßstab erlauben, sind daher notwendig, wenn wir nicht eine redundante Speicherung der Geometrie für verschiedene Maßstäbe anstreben.

Für gewisse Teilaspekte der Generalisierung gibt es sehr wohl Ansätze. In Abschnitt 6.4 haben wir in der Filterung von rastermäßig gegebenen Flächen ein Mittel entdeckt, das die Generalisierung solcher Flächen bewerkstelligt. Solche Ansätze lassen sich auch teilweise auf Vektordaten - etwa auf Schichtenlinien - übertragen (Schweinfurth 1984). Die Idee der Filterung steht aber auch hinter der gleitenden Mittelbildung bei Schichtenlinien, Gewässerlinien und ähnlichen linienhaften Strukturen: Jeder Punkt wird durch

ein gewichtetes Mittel seiner Umgebung ersetzt. Die Größe dieser Umgebung ergibt den Grad der Generalisierung.

In diesem Abschnitt wollen wir weitere Generalisierungsvarianten für Daten im Vektorformat behandeln. Gewöhnlich unterscheidet man vier Generalisierungseffekte:

- die **Auswahl** von Objekten, die in der generalisierten Karte nicht mehr aufscheinen sollen oder dort durch ein platzsparendes Symbol ersetzt werden sollen,
- die **Vereinfachung** von Objekten,
- die **Zusammenfassung** von ursprünglich selbständigen Objekten zu einem Objekt der generalisierten Karte, und schließlich
- die **Verdrängung** von Objekten, die aufgrund des verminderten Platzangebotes in der generalisierten Karte nicht mehr an ihrer geometrisch korrekten Position dargestellt werden können.

Für diese einzelnen Teilbereiche der Generalisierung gibt es viele Beispiele, ebenso für das Zusammenspiel aller dieser Bereiche: während man bei einer Gebäudedarstellung im ursprünglichen Maßstab auf sehr viele Details, wie Mauervorsprünge, Innenhöfe und dergleichen Rücksicht nehmen kann, müssen diese Gebäudekonturen im generalisierten Bild vereinfacht werden; im Extremfall wird jedes Gebäude durch ein Rechteck dargestellt. Kleine Gebäude verschwinden ganz; dies ist ein Ergebnis der Auswahl; Häuser, die in geringem Abstand zueinander stehen, verschmelzen zu Häuserblocks. Im Altstadtbereich würden dadurch schmale Gassen zuwachsen, so daß hier die Häuser nach beiden Seiten hin verdrängt werden müssen. Durch die Verdrängung entsteht eine neue Situation, die wieder eine der vorhin genannten Operationen anstößt. Genauso können durch die Zusammenfassung neue "Vorsprünge" an den zusammengefaßten Gebäudeblocks entstehen, die wiederum vereinfacht werden müßten usw. Es ist somit klar, daß all diese Operationen einander beeinflussen und daß die *Reihenfolge*, in welcher diese Operationen durchzuführen sind, das Endergebnis beeinflusst.

Die Kriterien für die Ermittlung jener Objekte, die zu generalisieren sind, können **geometrischer** Natur sein: so kann man etwa alle jene flächenhaften Objekte eliminieren, deren Flächenmaß in der generalisierten Karte ein bestimmtes Mindestmaß unterschreitet; man faßt beispielsweise auch jene Objekte zusammen, die so nahe aneinanderrücken, daß eine gewisse Toleranz unterschritten wird, oder man vereinfacht kleine Mauervorsprünge. Geometrische Abfragen können natürlich leicht automatisiert werden.

Sehr oft jedoch treten geometrische hinter **thematischen Kriterien** zurück. Man läßt alle Gemeindestraßen weg und erhält nur Landes- und Bundesstraßen. Man faßt nur Gebäude gleicher Nutzung zusammen; man vereinfacht nur jene Bereiche im Straßenverlauf, deren Informationsgehalt gering ist, während man gerade Haarnadelkurven besonders hervorhebt. Solche thematische Kriterien sind einer Automatisierbarkeit naturgemäß weniger zugänglich wie geometrische Abfragen; deswegen führt man **Rang-Deskriptoren** ein, welche die thematische Wertigkeit der Objekte besser beschreiben: die Wertigkeit einer Kleinstadt ergibt sich beispielsweise aus der Einwohnerzahl, der Anzahl der Schulen, der kulturellen Bedeutung usw. Man kann also durch eine entsprechende Gewichtung der einzelnen Komponenten den automatischen Ablauf sehr gut beeinflussen.

Für nähere Einzelheiten zu den einzelnen Generalisierungsansätzen verweisen wir auf Weber (1982a), Grünreich (1986), Klauer (1986) und McMaster (1987).

6.8. Weitere Werkzeuge

In den bisherigen Abschnitten haben wir einige Beispiele aus dem Instrumentarium eines raumbezogenen Informationssystems hervorgehoben. Nicht jedes System unterstützt die Gesamtheit der vorgestellten Methoden; andererseits konnten wir nur eine Auswahl treffen, und es würde den Rahmen dieses Buches sprengen, wenn wir versuchten, auf all die anderen denkbaren (und auch praktisch im Einsatz stehenden) Methoden einzugehen.

Wir wollen jedoch wenigstens in Kurzform einige weitere Themenkreise erwähnen:

6.8.1. Ausgleichungsprobleme

Paßpunkttransformationen mit Überbestimmungen dürfen in keinem raumbezogenen Informationssystem fehlen. Neben einer globalen Genauigkeit kommt es hier auch auf eine **Nachbarschaftsgenauigkeit** an. Die Auswahl der Paßpunkte, die Elimination grober Fehler und weitere Hilfestellungen bei der Erstellung einer optimalen Konfiguration sind wünschenswert.

Der automatische **Randausgleich** für zwei aneinanderstoßende (gegebenenfalls überlappende) Kartenblätter ist bei der Nachbearbeitung digitalisierter Vorlagen notwendig. Dabei kommt es zunächst auf eine entsprechende Mittelung der

divergierenden Geometrien an; aber auch thematische Aspekte müssen kontinuierlich ineinander übergehen: eine möglichst eigenständig ablaufende Objektbildung für jene Elemente, die die Blattränder kreuzen, bringt entscheidende Vorteile.

Die **Einhaltung geometrischer Bedingungen**, wie etwa die Rechtwinkeligkeit, die Parallelität, die Beachtung von Sollwerten für Distanz, Richtung, Winkel und Flächeninhalte, ist auch der Nachbearbeitung digitalisierter Vorlagen zu-rechenbar. Unterschiedliche Genauigkeiten in den Vorlagen, in der Erfassungsmethodik, in der verwendeten Hardware mögen die Ursache sein; aber auch bei der eben erwähnten Randanpassung müssen solche Bedingungen beachtet werden; letztendlich sind sie für viele Generalisierungsschritte ebenso unerlässlich. Solche Vorgaben können etwa durch den Ansatz eines vermittelnden Ausgleichs mit zusätzlichen Bedingungen realisiert werden (Haag u. Köhler 1985).

6.8.2. Pufferzonen und Freistellungsflächen

Die Grundlagen für einen Verschneidungsmodul wurden bereits in Abschnitt 6.3 erläutert. Darauf aufbauend, können Algorithmen zur Berechnung von **Pufferzonen** erstellt werden: Längs einer Straße soll beispielsweise ein Streifen von 100 m angelegt werden, und alle jene Grundstücke, die in diesen Streifen hineinfallen, sind zu ermitteln. Die Effizienz solcher Pufferzonenberechnungen hängt von der Form der zugrundeliegenden Linien ab: gerade oder wenig gekrümmte Linien sind leichter zu behandeln.

Auch die **Freistellung von Texten und Symbolen** kann durch Verschneidungsalgorithmen unterstützt werden. Man versteht darunter das teilweise Verdecken des geometrischen Hintergrundes durch Symbole und Beschriftungen im Rahmen einer Planaufbereitung. Soll dies weitgehend automatisch ablaufen, so hat das Programm eine Fülle von Bedingungen zu beachten, die sich aufgrund von Prioritäten zwischen einzelnen Kartenelementen ergeben. Oft wird unter diesem Titel auch die Problematik der **Plazierung von Texten und Symbolen** subsumiert, bei der eine Reihe von Entscheidungen, die ein menschlicher Bearbeiter treffen würde, von einem Programm simuliert werden müssen. Methoden aus der Rasterdatenverarbeitung können bei der Plazierung günstig eingesetzt werden (Giebels 1983).

6.8.3. Automatische Flächenbildung und Aggregieren

Automatische Flächenbildungen stellen einen weiteren umfangreichen Zweig von raumbezogenen Methoden dar. Es geht darum, eine zunächst unsortiert vorliegende Menge von Kanten, die als Ergebnis eines Digitalisiervorganges entsteht, aufgrund topologischer Kriterien zu Flächen zusammenzufassen. Thematische Gesichtspunkte (welche Kanten dürfen zu welchen Flächen zusammengefaßt werden?) müssen dabei ebenso beachtet werden wie Genauigkeitsfragen; denn Kantenfolgen schließen sich vielleicht nicht, sie "schießen über das Ziel hinaus" (*overshooting*), sie schneiden sich selbst usw. (Joham 1988). Mit den Algorithmen der Flächenbildung sind die Methoden des Aggregierens verwandt. Diese Verfahren legen benachbarte Teilflächen (etwa Nutzungsflächen eines Grundstückes) zu einer Gesamtfläche zusammen. Dabei sind wieder topologische und graphentheoretische Hilfsmittel vorteilhaft, ebenso wie eine Datenstruktur, die nach dem Kanten-Knoten-Prinzip aufgebaut ist, und die zu jeder Kante die linke und die rechte Nachbarfläche kennt.

6.8.4. Netz- und Flußoptimierung

Netzoptimierungs- und Flußalgorithmen spielen eine große Rolle im Leitungskataster. Wo liegen die Engstellen eines Leitungsnetzes? Welche Bereiche müssen verstärkt werden? Wo kann eingespart werden? Aber auch in der Regionalplanung sind Anwendungen denkbar: wo muß ein vorhandenes Straßen- und Eisenbahnnetz verbessert werden? Wo können vorhandene Anlagen besser genutzt werden?

6.8.5. Mustererkennung

Die *Mustererkennung* ist ein umfangreiches Spezialgebiet, das wir hier nur kurz erwähnen können. Der Engpaß in der Datenerfassung kann erst durch einen wirksamen Einsatz von Scannern beseitigt werden, und dies auch nur dann, wenn es gelingt, auch die Erfassung thematischer Informationen zu automatisieren: Eisenbahn- und Autobahnsignaturen müssen (aufgrund geometrisch-topologischer Kriterien) als solche erkannt werden, Gebäude sollen klassifiziert werden, der Baumbewuchs soll aufgrund von Infrarotbildern klassifiziert werden und ähnliches mehr.

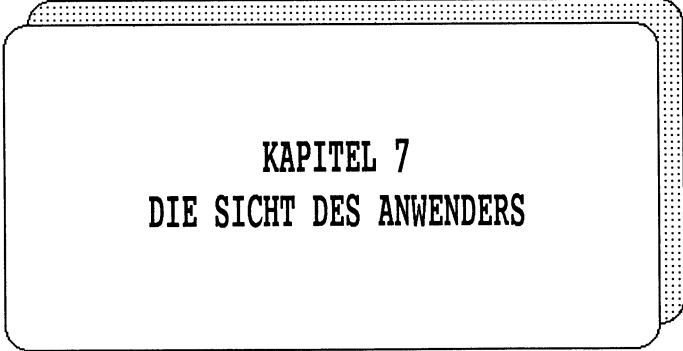
6.8.6. Karten- und Plangestaltung

Dazu zählen wir die Ausgestaltung des Kartenrandes, die Erstellung eines Plankopfes und die Unterstützung bei der Generierung von Legenden. Auch das Ablegen mehrerer verschiedener Karteninhalte in gesonderten Fenstern und die relative Plazierung der einzelnen Fenster gehört dazu, ebenso wie eine Reihe von Transformationen des Bildinhaltes, die einen optisch ansprechenden Gesamteindruck vermitteln.

Zu erwähnen ist auch noch die Bereitstellung eines Editors für die interaktive Ausformung von Symbol- und Linien-signaturen, von Schraffurmustern und von Textfonts.

6.8.7. Umwandlung von Datenformaten

Algorithmen für die Überführung von Vektor- in Rasterdaten und umgekehrt (Vektorisierung und Rasterung) müssen angeboten werden, ebenso wie Aufbereitungsprogramme für tachymetrische und photogrammetrische Rohdaten. Daneben sollten noch Schnittstellen zu Fremdsystemen bereitstehen. Die Datengewinnung und die Einbindung externer Datenquellen stellt eine der wichtigsten Kriterien für ein benutzerfreundliches System dar. Je flexibler und einfacher die Handhabung solcher Umwandlungsprogramme ist, desto besser wird die Akzeptanz des Systems sein.



KAPITEL 7
DIE SICHT DES ANWENDERS

7.1. Das raumbezogene Informationssystem als "black box"

für den Anwender

In den vorangegangenen Kapiteln standen die Konzeption eines raumbezogenen Informationssystems, die geometrischen, topologischen und thematischen Grundlagen, die Speicherung der Daten und das Angebot an Software-Werkzeugen im Vordergrund. Wir erkannten, daß ein RIS nur dann sinnvoll in Applikationen eingebunden werden kann, wenn die darin gespeicherten Daten redundanzfrei, konsistent und vollständig sind; eine Verwaltungsinstanz muß die Einhaltung dieser Bedingungen überwachen; die Daten und Bearbeitungsmethoden müssen weitgehend von den Applikationen abgekoppelt werden können. Das Informationssystem muß diesen Applikationen eine Betrachtungsweise zugestehen, in der es selbst als "black box" auftritt: die Daten fließen gebündelt über einige (wenige) Ein- und Ausgabekanäle, während ihre interne Verwaltung für den einzelnen Anwender unwichtig ist. Ähnliches gilt für die Werkzeuge, also die Algorithmen für die Transformation, die Interpolation, die Generalisierung, die Ausgleichung der Daten; hier muß es Schnittstellen in der Form von Programmaufrufen geben, die der Anwender in sein Programm einbauen kann.

Natürlich muß er sich ein Bild von dem Datenmodell und den Algorithmen machen können, die dem Informationssystem zugrunde gelegt wurden, aber die genaue Realisierung der Details interessiert ihn nicht. Sein Bild der Daten und Methoden ist vielleicht um vieles gröber als das Bild, das sich ein anderer Anwender von denselben Daten bzw. Methoden macht. Jede einzelne Applikation entwickelt eine Sicht, die unterschiedlich tief in das Informationssystem eingreift. Der Anwender sieht nur "die Spitze des Eisberges", während ihm der Großteil des Gesamtkonzeptes verborgen bleibt.

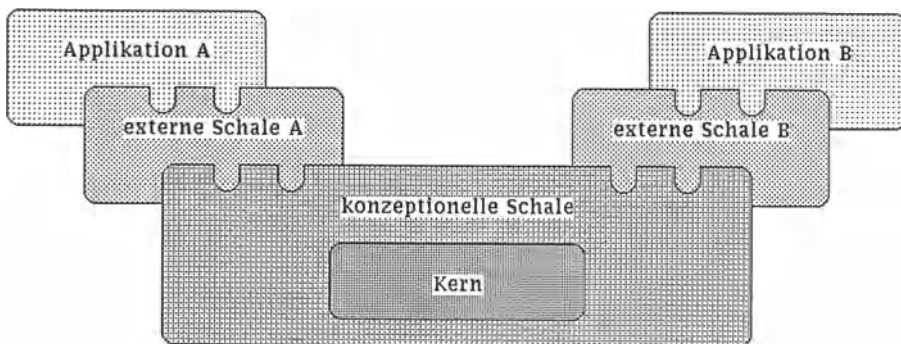


Abb. 7.1. Schalenweiser Aufbau eines Informationssystems

Es ist daher sinnvoll, wenn wir für die Erklärung des Konzeptes eines raumbezogenen Informationssystems einen schalenweisen Aufbau wählen (Späni u. Bartelme 1988). Jede Schale spricht eine andere Gruppe von Betrachtern an (Abb. 7.1): die **konzeptionelle Schale** legt die Datengruppen fest, die in einem Informationssystem auftreten, und ebenso die Werkzeuge, die zur Bearbeitung dieser Daten zur Verfügung stehen; sie wendet sich daher an den **Administrator** eines Informationssystems; an jenen Menschen, der - losgelöst von irgendwelchen Anwendungen - für das *logische Funktionieren* des Systems zuständig ist. Neben der konzeptionellen Schale gibt es mehrere **externe Schalen**, welche jeweils einen Extrakt der konzeptionellen Schale darstellen, der für eine bestimmte Applikation maßgeschneidert wurde und der unterschiedlich tief in die konzeptionelle Schale eingreift. Ganz im Inneren befindet sich dann noch der **Kern**, welcher die Aspekte der physischen Speicherung der Daten sowie den detaillierten Ablauf der zur Verfügung stehenden Algorithmen beleuchtet.

Diese drei Schichten treten auch in einem Datenbankmodell auf (siehe Kapitel 4). Im Sinne der Definition, die wir in der Einleitung für ein Informationssystem gefunden haben, ist ein solches System eine Datenbank, die mit einem Umfeld von Software-Werkzeugen umgeben ist. Damit ist es auch klar, daß wir das Schalenkonzept direkt verallgemeinern können.

Im vorliegenden Kapitel geht es uns also vor allem um die **externe Schale**. Wie kann ein Anwender eine Sicht der Daten und der damit verbundenen Werkzeuge entwickeln, die speziell auf seine Bedürfnisse abgestimmt ist? Wie können mehrere Anwender - unter Umständen gleichzeitig - solche Sichten verfolgen, ohne dabei miteinander in Konflikt zu kommen und ohne dabei das globale Konzept (die konzeptionelle Schale) zu verletzen? Diese Fragen wollen wir in den nächsten Abschnitten beantworten. Eine weitere Frage stellt sich zwangsläufig: in welchem Maße ist ein Informationssystem befähigt, **selbst** auf ein bestimmtes Anwenderprofil einzugehen? Die Beantwortung dieser Frage ist natürlich von eminenter Bedeutung für die Attraktivität eines solchen Systems; wir werden den Faden in diesem Kapitel aufgreifen; wir werden sehen, daß wir uns damit an das Gebiet der künstlichen Intelligenz und der Expertensysteme herantasten, und daher spinnen wir den Faden in Kapitel 8 weiter.

7.2. Anwenderprofil und Tabellen

Ein Informationssystem muß auf stark unterschiedliche Anwenderbedürfnisse eingehen; die externen Schalen (bzw. die externen Schemata) weichen voneinander ab. Das System

muß sich also dem *Anwenderprofil* anpassen. Jeder Anwender wird bestimmte *Themen* ansprechen; er hat jeweils unterschiedliche Vorstellungen von der Genauigkeit und von den damit verbundenen Konsistenzbedingungen; es werden nicht alle Anwender die gleiche Sprache sprechen: diese Aussage kann wiederum wörtlich aufgefaßt werden, indem Meldungen, Anweisungen und Hilfestellungen in unterschiedlichen Sprachen formuliert werden; sie kann aber auch bildlich verstanden werden, indem man auf "Anfänger" ausführlicher eingeht als auf "Fortgeschrittene". Soweit dies möglich ist, muß die Reaktion des Informationssystems der Vorgangsweise eines menschlichen Auskunftgebers angepaßt werden; dies kann man dadurch erreichen, daß man für jeden Anwender ein Profil anlegt, das die Art und den Umfang der Reaktionen des Systems festlegt. Dieses Profil kann natürlich immer wieder modifiziert werden, so etwa, wenn der einstmalige Anfänger geübter wird. Es kann in einem File abgelegt werden, der aktiviert wird, wenn der entsprechende Anwender identifiziert wird; damit hat es eine sehr große Ähnlichkeit mit den Benutzerprofilen und den damit verbundenen unterschiedlichen Privilegien in einer Rechnerumgebung. Der Anwender muß nicht unbedingt mit einer Person gleichgesetzt werden. Diese Person kann zu verschiedenen Zeiten in verschiedene Anwenderrollen schlüpfen. Das Anwenderprofil gehört somit ebensogut zum Themenkreis der **Projektbildung**, mit der wir uns in Abschnitt 7.3 beschäftigen werden. Wir wollen nun die einzelnen Komponenten des Anwenderprofils aufzählen (Haywood 1986).

7.2.1. Die Zugriffsberechtigung

Die Zugriffsberechtigung auf bestimmte Themen bzw. auf einzelne Datengruppen innerhalb dieser Themen ist ein wichtiges Kriterium zur Unterscheidung verschiedener Anwendergruppen, zur Entflechtung unterschiedlicher Interessen und zur Gewährleistung der **Datensicherheit** und des **Datenschutzes**. Die Berechtigung kann sich auf Lese-, Schreib- und Löschoptionen beziehen. So treten etwa in einem städtischen Leitungskataster die Ebenen "Gasleitungen", "Wasserleitungen", "Stromleitungen" usw. neben einer Ebene "Situation" auf. Die verschiedenen Leitungsbetreiber werden im allgemeinen für die ihnen zustehende Ebene eine Schreib- und Löschberechtigung haben. Alle Anwender werden jedoch die Ebene "Situation" zur Orientierung einblenden müssen; sie können diese Ebene aber nicht verändern, so daß sich ihr Zugriff auf Leseoperationen beschränkt. Aber auch innerhalb einer Ebene werden Zugriffsberechtigungen im allgemeinen differenzierter sein; so wird es privilegierte Benutzer geben, denen mehr Rechte zustehen als anderen; die Berechtigung wird sich unter Umständen auf einzelne Datengruppen beschränken; sie kann

aber auch für räumlich unterschiedliche Bereiche ausgesprochen werden: zwei Angestellte eines E-Werkes können gleichzeitig Änderungen in der Ebene "Stromleitungen" durchführen, aber nur in getrennten Stadtbezirken. Die Zugriffsberechtigung wird von einer zentralen Stelle, dem *Datenbankadministrator*, vergeben. Der Begriff "Datensicherheit" bezieht sich jedoch nicht nur auf die "Gefahr von außen"; um zu vermeiden, daß man durch Unachtsamkeit seine eigenen Daten gefährdet, wird sich jeder Anwender selbst von Zeit zu Zeit solche Beschränkungen auferlegen; dies dient auch einer besseren Übersichtlichkeit: man blendet temporär alle jene Dinge aus, die zur Zeit unwichtig sind, und konzentriert sich auf ein Teilgebiet.

7.2.2. Thematische Hierarchien

Thematische Hierarchien können von jedem Anwender definiert werden. Eine ausführliche Erläuterung zu diesen Hierarchien wird in Kapitel 3 gegeben. Kurz zusammengefaßt, geht es darum, daß man festlegt, welche Elemente man zu Objekten zusammenfassen darf, welche Elemente und Objekte einem gemeinsamen Thema angehören, und welche Attribute den Elementen bzw. Objekten zugeordnet werden können. Ein Beispiel dazu: Elemente der Kategorie "Gebäudelinie" dürfen zu Objekten des Themas "Gebäude" zusammengefaßt werden; in diesem Thema gibt es auch Objekte "Häuserblock", die aus (Teil-)Objekten "Gebäude" bestehen. Einem Gebäude können die Attribute "Eigentümername", "Adresse", "Versicherungsnummer", "Bauzustand" usw. zugeordnet werden. Die Festlegung einer solchen Hierarchie dient dazu, die Übersichtlichkeit der Datenstruktur zu erhöhen. Kurzzeitig mögen gewisse Einschränkungen als hinderlich angesehen werden; langfristig jedoch erhöhen solche klare Leitlinien den Anwenderkomfort.

7.2.3. Thematische Konsistenzvorgaben

Thematische Konsistenzvorgaben dienen ebenfalls dem Ziel eines möglichst einfachen und klaren Konzeptes, das sich der Anwender von den digitalen Datenbeständen machen kann. Wenn wir für ein Gebäude fordern, daß es eine zusammenhängende Fläche sein muß, daß es also nicht aus verschiedenen disjunkten Bereichen bestehen darf, so können sämtliche Anwenderprogramme auf dieser Voraussetzung aufbauen. Ein solches Anwenderprogramm, das den Flächeninhalt eines Gebäudes oder dessen Umfang berechnet, muß keine Verzweigungen beinhalten, die jenen Spezialfall abdecken. Ähnlich verhält es sich mit der Füllung einer Gebäudekontur mit einer Farbe oder einem Muster und mit der Verschneidung

von Gebäuden mit anderen topologischen Elementen. Wollen wir einen Bauernhof, der aus mehreren Gebäuden besteht, als eine Einheit behandeln, so ist es besser, einen gesonderten Objekttyp "Häusergruppe" einzurichten, der im Sinne der obigen Hierarchievorgaben aus Objekten des Typs "Gebäude" zusammengesetzt sein muß. Thematische Konsistenzvorgaben können sich aber auch auf ganze Themen ausdehnen (so etwa bei einem Gewässernetz, das eine baumartige topologische Struktur aufweisen muß) bzw. auf Kombinationen von Themen: wir können beispielsweise fordern, daß Gebäude und Verkehrsflächen einander nicht schneiden dürfen.

7.2.4. Wertebereichsvorgaben

Wertebereichsvorgaben können festlegen, daß Punktnummern numerisch sein müssen, daß die Gewässergüte durch eine der Zahlen 1-5 festgelegt wird, daß das Datum im Format TT-MMM-JJ einzugeben ist, und anderes mehr. Außerdem können wir zu jedem Attribut festlegen, ob und wann es mit einem Wert belegt werden muß: wenn wir etwa fordern, daß zu jedem Grundstück, das digitalisiert wird, sofort die Grundstücksnummer eingegeben werden muß, so können etwaige Anwendungsprogrammierer darauf vertrauen, daß ein Grundstück immer über seine Nummer aufgefunden werden kann. Wenn diese Überprüfung aber erst im Anschluß an den Digitalisiervorgang oder überhaupt erst beim Abschluß einer Projektsitzung pauschal durchgeführt wird, so gilt diese Regel natürlich nur beschränkt.

7.2.5. Genauigkeitsvorgaben

Genauigkeitsvorgaben beziehen sich auf die Erfassung geometrischer Einheiten, speziell von Punkten, und der dabei erzielbaren Genauigkeit. Denken wir an einen Digitalisiervorgang: die Genauigkeit der Vorlage, der aktuelle Maßstab, die Angaben des Herstellers bezüglich der Präzision des Digitalisiertisches, die Anzahl und die Verteilung der Paßpunkte ergeben in ihrer Gesamtheit ein Maß für die erreichbare Genauigkeit in den Punktkoordinaten. Diese Genauigkeit spielt eine tragende Rolle in einem raumbezogenen Informationssystem, bildet doch die Geometrie einen Grundpfeiler für alle in diesem System gespeicherten Informationen. Gerade die Angleichung von Daten, die aus verschiedensten Quellen stammen und daher stark unterschiedliche Genauigkeiten aufweisen, ist eine der wesentlichsten Herausforderungen, denen wir uns bei der Konzeption eines solchen Informationssystems stellen müssen (Muller 1987). Frank (1983a) spricht in diesem Zusammenhang

auch von **gemessener Geometrie** und **absoluter Geometrie** (Abb. 7.2). Während sich die gemessene Geometrie eines Punktes auf einen Erfassungsvorgang bezieht, muß die absolute Geometrie dieses Punktes als langfristig bedeutsamer gemeinsamer Nenner abgespeichert werden.



Abb. 7.2. Gemessene und absolute Geometrie von Punkten

Es handelt sich hier um ein **Datumsproblem**; streng genommen, müssen wir einen Ansatz wählen, der verschiedene Referenzsysteme miteinander in Beziehung bringt. In der Praxis behilft man sich jedoch mit der Vorgabe eines punktwise interpretierten Genauigkeitsintervalles und mit der Mittelung der dort hineinfallenden Koordinaten. Die Genauigkeit von Linien und Flächen läßt sich auf die Genauigkeit von Punkten zurückführen. Genauigkeitsvorgaben sind nicht nur während der Eingabe und erstmaligen Abspeicherung wichtig, sondern auch für nachfolgende Bearbeitungen, wie Transformationen, Ausgleichs- und Generalisierungsansätze.

7.2.6. Numerische Formate

Formate für die Ein- und Ausgabe stellen einen wichtigen Teil eines maßgefertigten Anwenderprofils dar. Zunächst handelt es sich um die verwendeten Maßeinheiten für die Lage, für Entfernungen und Richtungen. Während ein Anwender im Katasterbereich die Lage in Metern angibt und für Richtungen Neugrad verwendet, die sich auf die Nordrichtung beziehen, werden andere, großräumige Auswertungen in Kilometern und in einem anderen Winkelmaß durchgeführt. Das dabei verwendete Koordinatensystem kann jeweils ein ebenes System sein (z.B. Gauß-Krüger oder UTM); es kann sich aber auch um geographische Koordinaten (Länge und Breite) handeln. Auch die Anzahl der Stellen ist jeweils eine andere. Alle diese unterschiedlichen numerischen Formate müssen bei der Eingabe durch geeignete Transformationen in ein internes Referenzsystem übergeführt werden, das natürlich eindeutig sein muß.









7.2.7. Sprachen und Dialekte

Neben den numerischen Formaten müssen aber auch unterschiedliche alphanumerische Formate (*Sprachen und Dialekte*) berücksichtigt werden. Zu den internen Kennungen für Anweisungen und Antworten des Systems muß es daher eine Anzahl von extern manipulierbaren Files geben, von denen jeder einzelne in einer bestimmten Sprache abgefaßt ist. Das Anwenderprofil ordnet dann jeweils einen solchen File zu. Dieser kann demnach bei Bedarf auch im nachhinein erstellt werden. Der Begriff "Sprache" muß dabei nicht unbedingt im engeren Sinn verstanden werden; vielmehr beinhaltet er auch die Fähigkeit des Systems, auf den unterschiedlichen Grad an Erfahrung des jeweiligen Anwenders einzugehen. Im Bereich der Expertensysteme laufen aktuelle Untersuchungen zu dem Thema, wie weit es möglich ist, daß das System "intelligent" auf die Fähigkeiten eines Anwenders reagiert (siehe Abschnitt 8.7).

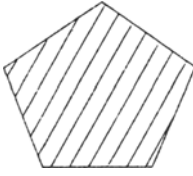
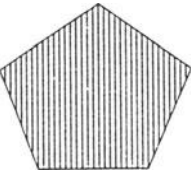
7.2.8. Graphische Formate

Graphische Formate sind ungemein wichtig, wenn wir das System als flexibles Instrument für vielfältige Darstellungsmöglichkeiten raumbezogener Phänomene nutzen wollen. Es gilt, den langfristig gespeicherten Daten eine aktuelle graphische Umsetzung gegenüberzustellen. Dies kann durch die Wahl der Farbe, des Symboltyps und der Symbolgröße, des Linientyps und der Linienbreite sowie der Schraffurmuster geschehen.

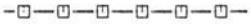

Die Art der graphischen Umsetzung hängt von den jeweils gewählten Themenkombinationen und von den Prioritäten ab, die man besonders hervorheben will. Blenden wir zum Thema "Bewaldung" auch noch das Thema "administrative Grenzen" ein, so dient dies nur zur räumlichen Orientierung. Die Grenzen werden sicherlich ganz anders darzustellen sein als im Fall einer Gemeindegemeinschaft, bei der ihnen eine primäre Bedeutung zukommt. Die Umsetzung hängt aber auch von der aktuellen Umgebung ab: die Hardware spielt hier ebenso eine Rolle wie der Umstand, ob man schnell eine Kontrollzeichnung in Händen haben will oder gewillt ist, längere Zeit auf eine Präzisionszeichnung zu warten. Wir können diesem Anspruch auf größtmögliche Vielfalt in der Darstellung durch die Bereitstellung von Tabellen begegnen, welche die Umsetzung der Daten in die aktuelle Situation beinhalten (Abb. 7.3). Jeweils eine solche Tabelle wird dann im konkreten Fall zugeordnet.

166	
167	
168	
169	
170	
171	
172	
173	

Symboltabelle

30	
35	

Schraffurtablette

1019	
1022	

Linientabelle

13	special-simplex
14	gothic-italian

Texttabelle

Abb. 7.3. Beispiele für graphische Umsetzungstabellen

7.2.9. Dialoggestaltung

Die Dialoggestaltung trägt wesentlich zu einer komfortablen Anwenderschnittstelle bei (Kuhn 1987). Jedem Anwender bleibt es überlassen, ob er über Kommandos oder über Menus mit dem System kommunizieren will. Er soll auch festlegen können, welche Abkürzungen er für Kommandos benutzen möchte. Hier ist ein Verweis auf die Möglichkeiten geboten, die ein leistungsfähiges Rechnerbetriebssystem anbietet: die eigentlichen Systemkommandos können durch anwenderspezifische Abkürzungen ersetzt werden. Genau dasselbe streben wir auch für den Benutzer eines raumbezogenen Informationssystems an. Aufgrund der Graphik müssen die hier angebotenen Möglichkeiten noch vielseitiger sein: wird der Dialog über Menus geführt, so muß festgelegt werden, ob es sich um eine fixe Menuleiste oder um ein Pop-up-Menu handelt, das erst dann eingeblendet wird, wenn es gebraucht wird, und sodann den Bildschirm wieder freigibt (Abb. 7.4). Größe, Form, Farbe und Schriftart dieser Menus sollen frei definierbar sein. Anwender, die an einem Digitalisierertisch arbeiten, wollen ein selbstgemachtes Soft-Menu verwenden, das neben der graphischen Vorlage auf den Tisch geklebt wird.

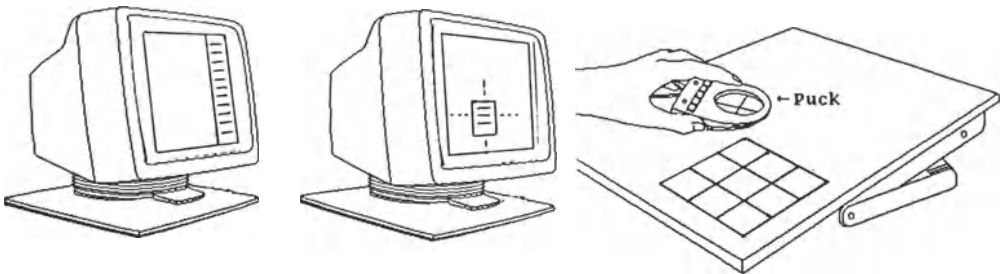


Abb. 7.4. Menüleiste (A) und Pop-up-Menü (B) am Bildschirm, Soft-Menu (C) am Digitalisierertisch

Wir haben hier nur die Äußerlichkeiten eines Menüs erwähnt, also Farbe, Form und Namensgebungen; weniger jedoch haben wir uns mit dem Inhalt auseinandergesetzt. In Abschnitt 7.7 befassen wir uns mit der Frage, wie der Anwender aus vorgegebenen Bausteinen selbst Benutzerfunktionen erstellen kann.

7.2.10. Standardeinstellungen

Standardeinstellungen erleichtern das Arbeiten am Terminal bzw. am Bildschirm. Es handelt sich dabei einerseits um Nuancen in der Systemkonfiguration, wie etwa spezielle Tastenbelegungen und die Bildschirmgestaltung (Farbe des Hintergrundes und des Vordergrundes, Zeichensatz und ähnliches). Aber auch Arbeitsabläufe können durch solche Einstellungen gesteuert werden: so kann der Anwender entscheiden, ob er bei der Erfassung eines Punktes interaktiv eine Punktnummer eingeben möchte, oder ob diese vom System nach einem fortlaufenden Numerierungsschema vergeben wird. Dies ist nur eines von vielen Beispielen für eine möglichst gute Anpassung des Systems an Anwendergewohnheiten. Es sind zwar zugegebenermaßen Details; beim praktischen Arbeiten erweisen sich aber gerade diese Fähigkeiten eines Systems als Wertmaßstab für seine Attraktivität und Benutzerfreundlichkeit. In diesem Zusammenhang gewinnt auch der Begriff **Ergonomie** eine Bedeutung, die über die physiologisch richtige Anordnung von Bildschirmen, Tastaturen und ähnlichem hinausgeht; nicht nur die Hardware trägt entscheidend zu einer benutzerfreundlichen Umgebung bei, sondern auch die **Benutzerschnittstelle (user interface)**.

7.3. Datenselektion und Projektbildung

Die Benutzer eines raumbezogenen Informationssystems sind nur an einem Teil der Daten interessiert, die in einem solchen System gespeichert sind. Sie definieren *Extrakte* aus der gesamten Datenmenge und bauen ihre Anwendungen, Auswertungen und Pläne auf diesen Extrakten auf. Da es sich dabei wahrscheinlich um umfangreiche Arbeiten handelt, werden sie in eine Reihe von Teileinheiten zerlegt, die unabhängig voneinander bearbeitet werden können. Eine solche Einheit nennen wir **Projekt**. Ein Projekt umfaßt demnach alle jene Daten des Gesamtsystems, die für ein bestimmtes Ziel - eine Auswertung - von Bedeutung sind. Innerhalb eines Projektes fallen im allgemeinen auch neue Daten an, und bestehende Daten werden verändert. Ist der Zweck eines Projektes erfüllt, so werden die veränderten und neu hinzugekommenen Daten in den Langzeitspeicher übernommen.

Es gibt mehrere Gründe, die für eine Aufgliederung in Teilprojekte sprechen:

- *Organisatorische Überlegungen* legen eine Aufteilung eines komplexen Gebietes in kleinere, überschaubare Einheiten nahe, die Hand in Hand mit einer lokalen Verantwortlichkeit geht.
- Haben mehrere Anwender *gleichzeitig* Zugang zum System, so kann ihre gegenseitige Behinderung auf ein Minimum reduziert werden.
- Handelt es sich um ein *verteiltes System* (siehe Kapitel 1), so werden die Daten bereits physisch in kleinere Blöcke zerteilt, die "vor Ort" gespeichert werden. Ein Nachvollziehen dieser Gruppierung auf logischer Ebene bringt sicher Vorteile.
- Kleinere Datenmengen haben gegenüber der gesamten Datenbank den Vorteil, daß sie die Antwort- und Verarbeitungszeiten begünstigen. Das Updating des Langzeitspeichers hingegen kann *en bloc* in betriebsschwachen Zeiten durchgeführt werden.
- Systemabsturz und "Irrläufe" können leichter abgefangen werden, wenn während lokaler Arbeiten der Langzeitspeicher unangetastet bleibt.
- Oft handelt es sich nur um das Durchspielen verschiedener *Varianten*, wie etwa bei der Projektierung eines Straßenverlaufes, und man möchte vermeiden, daß die dabei anfallenden Daten sofort in den Langzeitspeicher übernommen werden. Analoge Überlegungen sind bei der Planaufbereitung im Spiel, wo es hauptsächlich auf eine optisch ansprechende Gestaltung ankommt.

Es ergibt sich daraus, daß es nur dann sinnvoll ist, direkt an den originalen Daten zu arbeiten, wenn der Datenumfang gering ist, wenn nur ein Anwender vorhanden ist, oder wenn Zugriffszeiten keine Rolle spielen (wie etwa in der Batchverarbeitung). Allerdings fällt dann auch das Problem der Aufrechterhaltung der Konsistenz zwischen verschiedenen Teilprojekten weg. Bei der Projektbildung hingegen muß diese Konsistenz sehr wohl beachtet werden.

Die Auswahl der Daten, die zu einem Projekt gehören, nennen wir **Selektion**. Sie kann sowohl nach thematischen Kriterien wie auch nach geometrisch-topologischen Kriterien erfolgen. Wir wählen aus den Themen, die vom System angeboten werden, eine Teilmenge aus, und aus diesem Themenkreis interessieren uns jene Daten, die in einem bestimmten räumlich abgegrenzten Bereich liegen (Abb. 7.5). Dabei spielt natürlich die Zugriffsberechtigung eine Rolle. Die Berechtigung für das Lesen, Schreiben oder Löschen von Daten eines bestimmten Themas ist eine Sache des Anwenderprofils, welches im vorigen Abschnitt besprochen wurde.

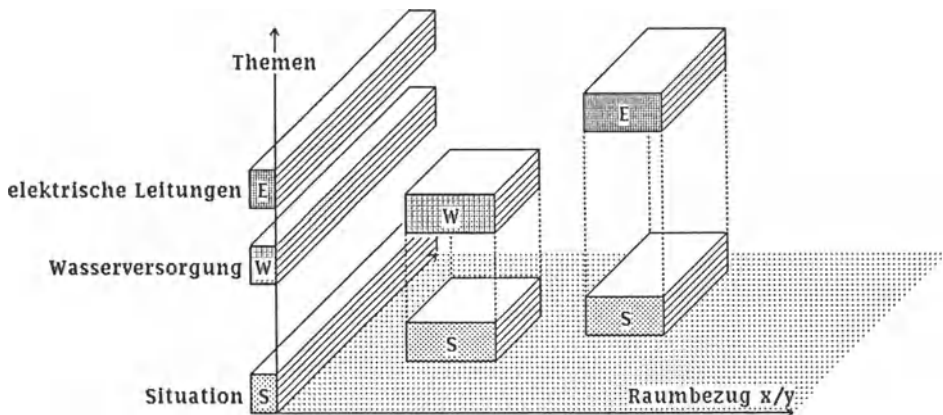


Abb. 7.5. Grobe thematische und räumliche Selektion

Die grobe Auswahl der Themen kann durch feinere Unterscheidungskriterien ergänzt werden. So können wir etwa die Ebene der Stromleitungen selektieren; wir können uns aber auch auf die 220V-Leitungen beschränken, auf die Freileitungen oder auf jene Leitungen, die schon älter als 10 Jahre sind; diese feine Selektion ergibt sich daher aus Attributwerten. Aber auch die räumliche Selektion kann aufgefächert werden: wir können grob ein Rechteck vorgeben und dies dann so verfeinern, daß wir die Grenzen eines Stadtbezirkes angeben und nur jene Leitungsteile selektieren wollen, die innerhalb

dieser Grenzen liegen. Es erweist sich als günstig, zuerst grobe Kriterien anzuwenden, weil wir hier mit relativ einfachen Mitteln den Großteil der Daten ausschließen können. Jene wenigen Daten, die dann noch übrig bleiben, können aufgrund der feineren Kriterien geprüft werden.

Nun ist die Frage zu lösen, was mit jenen Elementen und Objekten geschieht, welche die Grenzen des Stadtbezirkes überschreiten. Wir können keine allgemeingültige Antwort auf diese Frage geben. Stattdessen führen wir einige grundsätzliche Methoden an, die sich anbieten, und die man - gegebenenfalls in modifizierter Form - anwenden könnte:

A) Wir erweitern die Projektselektion dahingehend, daß diese Elemente bzw. Objekte ganz in das Projekt übernommen werden; die Grenzen werden also weiter gesteckt; natürlich können wir nicht einfach ein größeres Rechteck wählen, denn sonst würde das Problem erneut auftreten, nur mit anderen Elementen. Wir erweitern also gezielt, und zwar nur um die grenzüberschreitenden Leitungen (Abb. 7.6). Diesen müssen wir aber einen Sonderstatus zuweisen. Sie dürfen nicht verändert werden, denn sonst würde ihre (nicht mehr zum Projekt gehörende) Umgebung nicht konsistent mitgeführt werden. Wollen wir eine solche Leitung ändern, so müssen wir ein größeres Projekt selektieren.

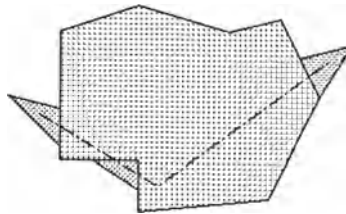


Abb. 7.6. Vermeidung von Verschneidungen durch Erweiterung der Projektgrenzen

B) Wir zerschneiden die Topologie an den Projektgrenzen. Der Einfachheit halber teilen wir das gesamte Gebiet, das von den Daten des Informationssystems überdeckt wird, in rechteckige Zellen auf und verschneiden die Topologie mit diesem Rechtecksgitter (Abb. 7.7).

Jedem Anwender werden - neben thematischen Zugriffsberechtigungen - auch räumliche Zugriffsberechtigungen erteilt, die sich auf die Zellen dieses Gitters beziehen. Bei der Projektselektion wird dann der von ihm angegebene Bereich automatisch auf die nächstgrößere(n) Zelle(n) erweitert. Selbstverständlich kann ein Anwender auch mehrere Zellen selektieren, und man kann die Anzahl der selektierten Zellen wieder mit den Privilegien einzelner

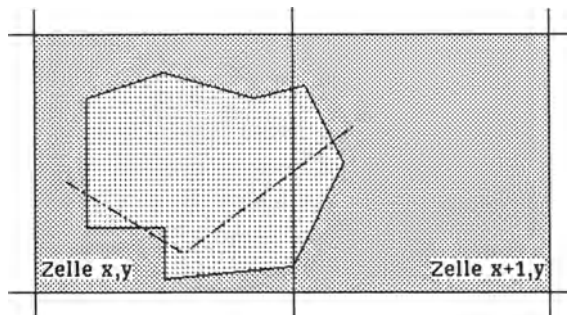


Abb. 7.7. Verschneidung der Topologie mit einem räumlichen Berechtigungsgitter. Erweiterung auf ganze Zellen

Anwender koppeln; so darf etwa nur der Administrator selbst alle Zellen selektieren. Diese Variante ist dann empfehlenswert, wenn der Datenbestand eher *statisch* ist (die Auswertung bestehender Daten steht im Vordergrund, während nur wenige Daten verändert werden oder neu hinzukommen), und wenn die Verteilung der Daten möglichst *gleichmäßig* ist.

C) Bei stark dynamischen und inhomogenen Datenbeständen ist eine Aufteilung des Datenraumes angebracht, wie wir sie in Kapitel 5 beschrieben haben: Baum- und Directory-Strukturen erfüllen am besten diese Voraussetzungen. Die Überlegung hinsichtlich der räumlichen Zugriffsberechtigung auf die (nun allerdings verschieden großen) Rechtecke bleiben dieselben; allerdings ist der Verwaltungsaufwand entsprechend höher.

Aus diesen drei Ansätzen lassen sich - wie immer - eine Reihe von Variationen ableiten. Ein grundsätzliches Problem besteht nach wie vor: Viele raumbezogene Informationssysteme nehmen für sich den Vorteil einer *blattschnittfreien* Speicherung der Daten in Anspruch; damit ist gemeint, daß es nicht notwendig ist, daß Projekte an vorgegebenen Linien "abgeschnitten" werden. In ihrer strengen Form läßt sich diese Forderung nur erfüllen, wenn jeweils nur *ein* Anwender das System verwendet und wenn er auf den *gesamten* Datenbestand zugreift. In allen anderen Fällen müssen wir diese Forderung dahingehend abschwächen, daß ein solches Abschneiden zwar letzten Endes unvermeidlich ist, daß sich jedoch die betroffenen Bereiche durch eine geeignete Wahl der Projektgrenzen bzw. der Berechtigungsmaschen auf den Rand des Projektes beschränken, wo sie - belegt mit einem Schreib- und Löschverbot - keine Probleme mehr verursachen.

7.4. Projektbeschreibung und Projektparameter

Durch die Selektion wird eine Teilmenge der Daten des Informationssystems zu einem Projekt zusammengefaßt. Dabei kann es sein, daß die Daten *physisch* in einen *Arbeitsbereich* (*work file*) kopiert werden, oder daß lediglich eine logische Zusammenfassung im Sinne eines *View* einer Datenbank erfolgt. Die verschiedenen Varianten wurden in Abschnitt 5.5 aufgezählt.

Die Daten stellen zwar - quantitativ - den Hauptanteil dessen dar, was ein Projekt ausmacht; es ist jedoch unerlässlich, daß dem Projekt eine Beschreibung (*dictionary*) beigelegt wird, welche die Daten und die Intentionen des jeweiligen Anwenders näher beschreibt. Wir können es in Anlehnung an Abschnitt 7.2 auch *Projektprofil* nennen. In der Tat ist es sehr eng mit dem Anwenderprofil verflochten. Die beiden Begriffe verschmelzen sogar, wenn Anwender immer mit bestimmten Projekten in Verbindung gebracht werden und umgekehrt, wenn also die Beziehung zwischen Anwendern und deren Projekten in beiden Richtungen *eindeutig* ist. In diesem Sinne fließen Teile der Charakteristika eines Anwenderprofiles auch in das Projektprofil ein. Der Anwender wird sein Profil als *Vorlage* ("*template*") verwenden, das er in einigen Details speziell an die jeweiligen Projektanforderungen anpassen wird. Dabei kann man sich als Beispiel das Ausfüllen eines Formulars vor Augen halten, in dem einige Eintragungen fest vorgegeben sind, für andere Eintragungen ein Vorschlag gemacht wird, der noch verändert werden kann, und zusätzliche Eintragungen hinzukommen.

Beispiele für solche zusätzliche Eintragungen wären etwa die Projektnummer, das Datum der Projektbildung bzw. der letzten Projektsitzung, der Name des Bearbeiters, der Name des Auftraggebers, Bemerkungen zum Fortgang der Arbeiten (*Journalfile*), Hinweise auf ähnliche Projekte usw. All diese Eintragungen sind *administrativer* Art. Daneben gibt es *technische* Eintragungen, wie etwa die Hardwareumgebung, die peripheren Geräte (Ein- und Ausgabegeräte), die im Rahmen des Projektes benötigt werden, ferner Parameter, die zur Steuerung des Programmablaufes benötigt werden (Radien von Ortungskreisen, Abbruchkriterien für bestimmte Algorithmen, a-priori-Fehler) und ähnliches.

Projektbeschreibungen tragen - ebenso wie Anwenderprofile - wesentlich zur Benutzerfreundlichkeit eines Informationssystems bei. Die Vielfalt der Verflechtungen, die im Laufe der Zeit in diesen Beschreibungs- und Profiltabellen entstehen, verlangen nach einer einheitlichen Behandlung mit Methoden der relationalen Algebra. Neueste Erkenntnisse auf dem Gebiet der künstlichen Intelligenz können hier ebenfalls praktisch erprobt werden (siehe Kapitel 8).

7.5. Multi-User-Betrieb

Nach der Selektion der Daten eines Projektes und dem Ausfüllen der Projektbeschreibung arbeitet der Anwender *lokal*: für ihn erscheinen die Verbindungen zum Langzeitspeicher kurzfristig unterbrochen zu sein. Dies trifft insbesondere dann zu, wenn die Daten auf eine lokale Datenstruktur kopiert werden; aber auch dann, wenn die Projektbildung nur *logisch* erfolgt (etwa im Sinne eines *View* einer relationalen Datenbank), so werden etwaige Änderungen im Langzeitspeicher vorerst nur in einem Journalfile vorgemerkt. Nach Abschluß der Projektarbeiten werden dann alle Änderungen pauschal im Langzeitspeicher nachgeführt. Beim Auftreten von Problemen in Form von Inkonsistenzen wird die gesamte Transaktion verweigert; der Anwender wird auf diese Probleme mittels eines mitgeführten Protokolles aufmerksam gemacht; die Transaktion kann erst dann durchgeführt werden, wenn er die Inkonsistenzen bereinigt hat.

Der Abschluß einer Transaktion muß nicht unbedingt mit dem Ende der Arbeiten in einem Projekt gleichgesetzt werden. Man kann sich auch vorstellen, daß ein Update auch zwischendurch angeregt wird. Dafür kann der Anwender in der Projektbeschreibung ein Zeitintervall angeben; der Programmablauf würde beispielsweise zu jeder vollen Stunde unterbrochen; der Abschluß der bestehenden Transaktion und der Beginn einer neuen kann aber auch explizit angestoßen werden.

Arbeiten mehrere Anwender gleichzeitig im System, so müssen die Spielregeln für einen solchen *Multi-User-Betrieb* definiert werden. Sie orientieren sich an der Vorgangsweise, die in einer Datenbankumgebung üblich ist; allerdings kommt bei einem raumbezogenen Informationssystem der Aspekt der relativen Lage zweier Anwendergebiete hinzu, der bei anderen Datenbanken nicht vorhanden ist. Zwei Anwender, deren Interessensgebiete sich nicht überschneiden, stören einander nicht, auch wenn sie dieselben Themen selektieren. Zwei Anwendungen mit verschiedenen Themen im selben räumlichen Ausschnitt jedoch beeinflussen einander sehr wohl: die Themen "Grundstücke" und "Verkehrswege" greifen auf dieselben Punkte und Kanten zu; werden Grundstücksgrenzen geändert, so beeinflußt dies auch die Geometrie der Verkehrswege. Wie wir bereits in Abschnitt 7.3 feststellten, führt man in diesem Fall ein Gitter von Berechtigungszonen ein (Abb. 7.7). Für die Vergabe der Update-Berechtigung in diesen Zonen stehen mehrere Strategien zur Auswahl:

- A) In jeder Zone darf nur eine Update-Berechtigung vergeben werden. Derjenige Anwender, der als erster ein Projekt selektiert, das diese Zone beinhaltet, ist in dieser Hinsicht privilegiert. Alle nach diesem Zeitpunkt gebildeten Projekte haben nur eine Leseberechtigung.
- B) Die Update-Berechtigung für eine Zone wird grundsätzlich mehreren Anwendern zugebilligt. Die jeweilige Zone wird aber erst dann gesperrt, wenn ein Anwender ein Update durchführen möchte.
- C) Alle konkurrierenden Updates werden synchronisiert, d.h. sie werden *simultan* zu betriebsschwachen Zeiten (z.B. nachts) durchgeführt; ergeben sich Konsistenzprobleme, so werden alle betroffenen Anwender verständigt.

Zu erwähnen ist noch, daß alle eben aufgezählten Alternativen sowie deren Kombinationsmöglichkeiten auch mit den oben erwähnten zwischendurch angestoßenen Transaktionen verträglich sind. In einer Datenbankumgebung kann man sehr wohl - etwa tagsüber - einen solchen Updateprozeß in Gang setzen. Dieser wird dann nur logisch durchgeführt, während die eigentlichen Änderungen im Langzeitspeicher erst später gemacht werden. Die Fähigkeiten eines Datenbankverwaltungssystems bezüglich *roll-back* und *roll-forward* werden hier ausgenutzt (siehe Kapitel 4). Wir können diese unterschiedliche Betrachtung auch auf unsere Überlegungen hinsichtlich des konzeptionellen Schemas einer Datenbank und der verschiedenen externen Schemata zurückführen: das Anstoßen eines Update-Vorganges gehört in das externe Schema, während die tatsächliche Anpassung der Daten dem konzeptionellen Schema zugerechnet wird.

Zusammenfassend können wir feststellen, daß für eine Transaktion keine allzu lange Zeitspanne angesetzt werden sollte, denn die Änderungen im Rahmen eines Projektes würden dann erst sehr spät zu anderen Anwendern durchdringen. Andererseits liegt es auch nicht im Sinne eines Projekt- und Transaktionskonzeptes, diese Zeitspanne auf Null zu reduzieren, etwa indem man jeden verschobenen Punkt sofort an den Langzeitspeicher rückmeldet; in diesem Fall würden die Vorteile einer Dezentralisierung und eines damit verbundenen effizienten lokalen Datendurchsatzes zunichte gemacht.

7.6. Netze und verteilte Datenbanken

Viele relationale Datenbanksysteme bieten heute bereits eine Netzwerkarchitektur an. Mehrere Arbeitsstationen und Computer unterschiedlichster Herkunft und Größenordnung - angefangen von Personalcomputern bis zu Großrechnern - sind

über leistungsfähige Leitungen zu einem Netzwerk verbunden (Abb. 7.8). Die Software ist auf all diesen Anlagen *funktional* identisch (obzwar sie natürlich intern auf die Erfordernisse des jeweiligen Rechners optimiert wird). Dies ermöglicht dem Benutzer den Zugriff auf beliebige Datenbanken im Netzwerk. Das bedeutet, daß die Grenzen zwischen heterogenen Hardwarekonfigurationen und Betriebssystemen verschwinden.

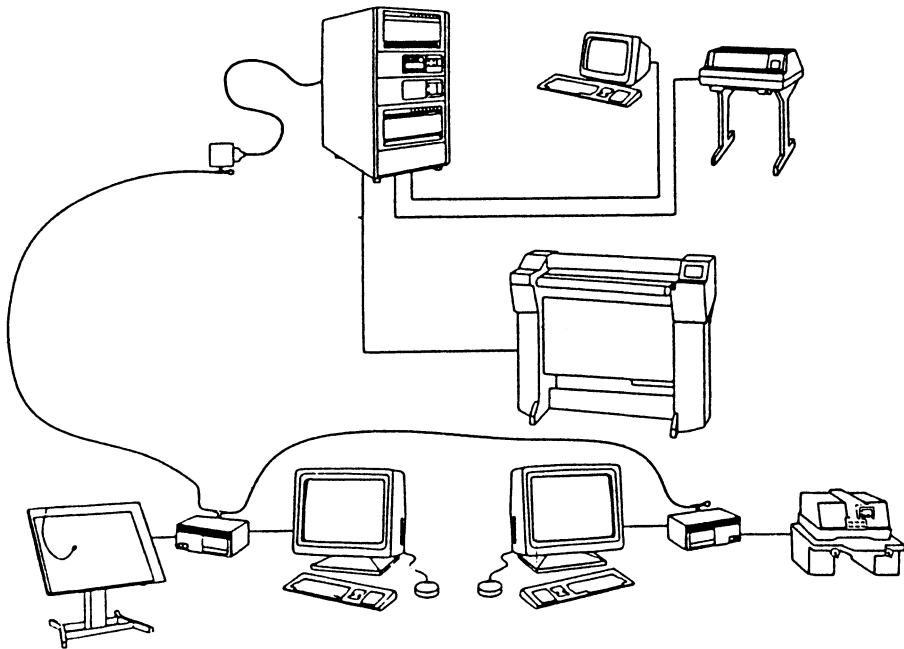


Abb. 7.8. Beispiel für eine Netzwerkarchitektur

Viele Systeme gehen sogar noch weiter und bieten die verteilte Datenverarbeitung von Datenbanken und Anwendungsprogrammen an. Informationen werden dort gespeichert, wo sie oft gebraucht werden. Der Anwender profitiert davon; er greift auf Daten und Programmpakete zu, ohne daß er über den Ort der *physischen* Speicherung dieser Informationen Bescheid wissen muß. Das Konzept der Projektselektion im Rahmen von Transaktionen kommt dieser Architektur natürlich sehr entgegen. Projekte und ihre Daten werden lokal gehalten. Die meisten Operationen sind also lokal durchführbar und belasten die Datenleitungen nur wenig. Trotzdem erlaubt es die Netzwerkumgebung, daß der Benutzer zwischendurch auch auf andere Bereiche zugreifen kann.

7.7. Benutzerspezifische Funktionen

In den vorangegangenen Abschnitten stand die Sicht der Daten eines raumbezogenen Informationssystems aus der Perspektive des Anwenders im Vordergrund. Wie wir im einleitenden Kapitel feststellten, ist ein Informationssystem jedoch mehr als eine Datenbank. Es beinhaltet noch eine Schale von Werkzeugen, die auf diese Daten zugreifen und dem Anwender nutzbar gemacht werden können. In diesem Abschnitt werden wir daher näher auf die Problematik eingehen, die sich für jenen Anwender auftut, der diese Werkzeuge und damit die zugrundeliegende Software in seine Programme einbauen will. Diese Vorgangsweise auf der Ebene der Anwendersoftware entspricht somit dem Bausteinprinzip, das wir auch erfolgreich beim Zusammenbau von anwenderspezifischen Datenstrukturen aus elementaren Datenelementen ausgenutzt haben (wir haben diese Views in Kapitel 4 besprochen).

Die Anforderungen, die eine solche Anwenderschnittstelle zu erfüllen hat, entspricht weitgehend den Anforderungen, die wir an ein - vielseitig zu verwendendes - Paket von Unterprogrammen stellen. (Viele - aber nicht alle - der Operationen, die wir in Kapitel 6 kennengelernt haben, lassen sich auch durch den Einbau von Unterprogrammaufrufen realisieren.) Sie können am besten mit dem Begriff **Ergonomie der Anwendersoftware** umrissen werden. So wie wir eine ergonomische Anordnung der Hardwarekomponenten fordern, so vehement müssen wir auch im Bereich der Software nach mehr Benutzerfreundlichkeit streben. Die EDV soll sich den Benutzerwünschen anpassen und nicht umgekehrt.

Zu dieser Software-Ergonomie gehören folgende Aspekte:

- Die **Einfachheit** der Operationen ist ein wichtiges Kriterium. Damit meinen wir nicht die Einfachheit der Details, denn diese können etwa bei geometrischen Ausgleichungen oder Generalisierungsvorgängen sogar sehr kompliziert werden. Vielmehr soll es für den Anwender einfach sein, ein gedankliches Modell für die Operation, deren Ausgangsposition und für die Resultate zu entwerfen.
- Die **Konsistenz** ist eine Forderung, die wir bis jetzt immer nur auf die Daten bezogen haben. Aber auch die Methoden, die von einem Informationssystem angeboten werden, sollen untereinander konsistent sein: die Art des Aufrufes, die Übermittlung der Eingabe- und Ausgabedaten, das Verhalten beim Auftreten von Fehlern soll immer gleich sein. Wir müssen es dem Anwender leicht machen, aus der Erfahrung zu lernen. Hat er einen Modul zu beherrschen gelernt, so soll er seine Erfahrungen auch bei anderen Modulen ausnützen können.

- Die **Robustheit** hängt eng mit der Konsistenz zusammen. Das System soll auf kleine "Unterlassungssünden" des Anwenders großzügig reagieren. Dazu gehört das Bereitstellen von Defaultwerten und die Formulierung von aussagekräftigen Fehlermeldungen, die bei der Suche nach der Fehlerursache behilflich sind.
- Die **Vollständigkeit** und die **Sparsamkeit** zielen auf ein System von möglichst elementaren Bausteinen ab, deren Grundgesamtheit ausreicht, um eine breite Palette von Anwendersituationen abdecken zu können; dabei sollen *möglichst wenige* Bausteine in *möglichst viele* Anwendungen einfließen.
- Die **Leistungsfähigkeit** schließlich beschreibt das Zeitverhalten eines Algorithmus. Natürlich sind viele Algorithmen, die im Rahmen raumbezogener Informationssysteme eingesetzt werden, extrem zeitaufwendig; daran wird sich in naher Zukunft auch nicht viel ändern. Unsere Intentionen gehen aber in die Richtung der Software-Ergonomie. Wir haben schon viel erreicht, wenn wir dem Anwender begreiflich machen können, *warum* eine Operation so lange dauert. Daß ein Generalisierungsvorgang Zeit kostet, wird man hinnehmen; der Bildaufbau hingegen soll rasch erfolgen, auch wenn dahinter eine riesige Datenbank steht. Hier können wir uns ein Beispiel aus dem Alltag vor Augen halten: Geschwindigkeitsbeschränkungen werden viel mehr beachtet, wenn der Grund (Lärmbelästigung, Schulweg und dergleichen) angegeben wird.

Zu einer **benutzerfreundlichen Schale** gehört also eine Menge von Subroutinen und Funktionen, die dem Anwender zur Verfügung gestellt werden. Die Unterscheidung zwischen diesen beiden Kategorien liegt in der Art der Einbindung: Subroutinen müssen in Unterprogrammbibliotheken abgelegt sein, die beim *Linkprozeß* zugeordnet werden (Abb. 7.9). Dies bedingt, daß dem Anwender ein **offenes System** zur Verfügung steht, das zumindest aus kompilierten Modulen besteht. Er muß auch eine gewisse EDV-Erfahrung mitbringen, die es ihm erlaubt, die Möglichkeiten eines Betriebssystems auszunutzen.

Wir können diese Einbindung aber auch auf einer anderen Ebene vorantreiben. Der Programmablauf wird durch einen File gesteuert, der dem Anwender zur Modifikation offensteht (Abb. 7.10). Benützerspezifische Besonderheiten schlagen sich in diesem File nieder, und deshalb können wir ihn dem Anwenderprofil (oder zumindest dem Projektprofil) zuordnen. Wir können auf diese Weise "von außen" in den Ablauf eingreifen und sogenannte **Makrofunktionen** definieren, die aus elementaren Funktionen zusammengesetzt sind. Die Programme sind in diesem Fall für sich allein voll lauffähig; es genügt also, wenn sie in der exekutierbaren Form vorliegen.

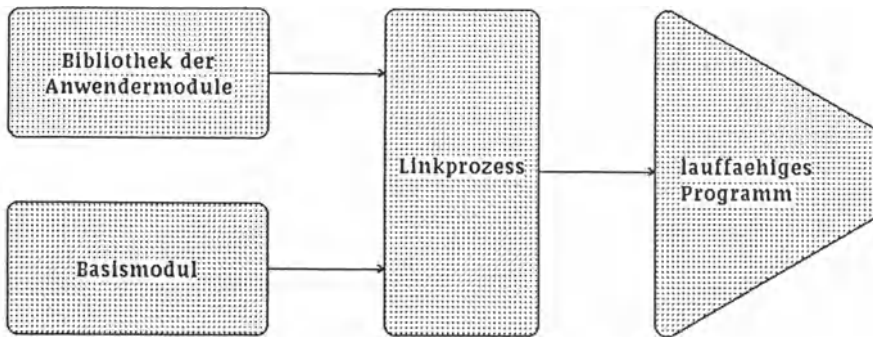


Abb. 7.9. Basismodul und Anwendermodule werden zu einem lauffähigen Programm zusammengebaut

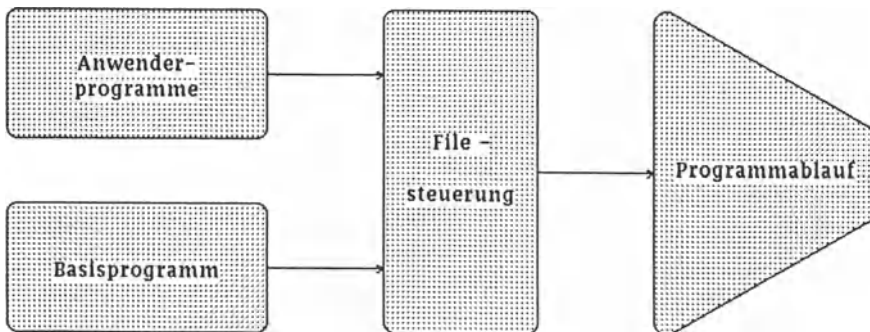


Abb. 7.10. Basisprogramm und Anwenderprogramme werden extern über einen File gesteuert

Auch der Datenaustausch vollzieht sich über Files. Wir können dieses Konzept noch verfeinern, wenn wir es ermöglichen, daß der Anwender selbst ein Menu mit selbstgemachten Makrofunktionen definieren kann, wenn er also selbst über die Anordnung einzelner Funktionen in einem Menu entscheiden kann. Eine andere Verfeinerung besteht darin, daß Steuerparameter zwischen den einzelnen Modulen ausgetauscht werden können.

Wir konnten in diesem Kapitel nur übersichtsweise auf verschiedene Ansätze bei der Konzeption einer komfortablen Anwenderschnittstelle eingehen. Viele der bestehenden Systeme gehen dem Anwender ein (mehr oder weniger großes) Stück Weges entgegen. Informationssysteme sind in dieser Hinsicht mit anderen Produkten gleichzusetzen, die dem "Konsumenten" angeboten werden: auf lange Sicht werden sich jene Systeme durchsetzen, die diesen Weg zum Anwender am weitesten zu gehen bereit - und imstande - sind. Nicht zuletzt deshalb laufen intensive Forschungen in dieser Richtung, die in den Bereich der künstlichen Intelligenz hineinführen. Wir wollen uns daher im nächsten Kapitel näher mit diesen neuesten Entwicklungen beschäftigen.



KAPITEL 8
AUSBLICKE

8.1. Künstliche Intelligenz:

von der Information zur Koordination

In diesem Buch haben wir viele Aspekte raumbezogener Informationssysteme beleuchtet: die Umgebung, die Daten und die Softwarewerkzeuge standen jeweils im Rampenlicht. In einigen Bereichen können wir bereits beachtliche Erfolge verbuchen, in anderen Bereichen steckt die Entwicklung noch in den Kinderschuhen, wobei hier meistens nicht die Hardware, sondern unzureichende oder noch nicht genügend ausgereifte Softwarekonzepte die Schuld tragen. Die Datenfülle, die vielfältigen Querverbindungen, der Widerspruch zwischen der individuellen Anpassung an den Anwender und einem hohen Maß an Allgemeingültigkeit - sowohl in der Sicht der Daten wie auch in der Auswahl der Methoden - kamen dabei ins Gespräch. Wir sind uns zwar einig, daß uns der Engpaß der Datenerfassung noch einige Jahre lang in Atem halten wird; dann aber stehen wir vor der Frage, wie wir uns davor schützen können, die Übersicht zu verlieren. Wie können wir das Datenangebot und das vorhandene Instrumentarium der Softwarewerkzeuge sinnvoll - und vor allem bequem - nützen?

In diesem letzten Kapitel wollen wir nun auch näher auf das eingehen, was im Bereich raumbezogener Informationssysteme in Zukunft möglich sein wird, welche Entwicklungen vorhersehbar sind, und wo die Grenzen zu ziehen sind. So wird hier, wie in anderen Bereichen auch, eine Steigerung der Kapazität von Hardware und Software nicht nur in quantitativer Hinsicht, sondern auch von der Qualität her zu erwarten sein. Schlagworte wie **künstliche Intelligenz** und **Expertensysteme** werden auch hier ihren Eingang finden; die sogenannte *5. Computergeneration* wird zur Lösung vieler Probleme beitragen. Sie wird aber auch neue Probleme schaffen.

Viele Bereiche der künstlichen Intelligenz sind noch weit von einem operationellen Einsatz entfernt. Uns geht es in diesem Kapitel jedoch nicht so sehr um die derzeitige *Machbarkeit*, sondern vielmehr um das *Konzept*. Wir wollen mit unserem Konzept nicht so lange warten, bis die dafür notwendige Hardware und Basis-Software angeboten wird. Wenn wir uns bereits vorher über unsere Bedürfnisse im klaren sind, so können wir frühzeitig Entwicklungstendenzen beeinflussen und dann - bei Verfügbarkeit der Basis - sofort "einsteigen".

Wir werden also die verschiedenen Bereiche der künstlichen Intelligenz näher betrachten und darlegen, welche Möglichkeiten sie für unsere speziellen Anwendungen im Bereich der RIS eröffnen. Unser Konzept soll aber mit ein wenig mehr Abstand vom rein technologischen Standpunkt

erstellt werden; vielmehr wollen wir dieses neue Spektrum von Möglichkeiten vor dem Hintergrund dessen beleuchten, was eigentlich der Ausgangspunkt war: der Mensch und seine Bedürfnisse.

Der Begriff **Künstliche Intelligenz** (Artificial Intelligence oder kurz AI - im Deutschen auch KI - genannt) ist eine Wortschöpfung der 50-er Jahre und entspricht demgemäß den Erwartungen, die man damals in die Zukunft der EDV setzte. Obwohl er nicht mehr ganz in unsere heutige Vorstellungswelt paßt, ist er inzwischen zu einem allgemein akzeptierten Sammelbegriff für eine Reihe von Forschungsschwerpunkten geworden, die auf vielen herkömmlichen Disziplinen aufbauen, angefangen von der Mathematik und ihrer verwandten Bereiche über die Datenverarbeitung bis in den Bereich der Psychologie und Soziologie.

Es gibt keine allgemein akzeptierte Definition dieses Begriffes. Selbst in anerkannten Fachbüchern wie etwa Nilsson (1982) findet sich keine Definition, sondern eher eine Umschreibung des Begriffes anhand von Beispielen. Viele sind sich aber einig, daß es darum geht, *in bestimmten Teilbereichen die menschliche Entscheidungsfindung zu simulieren*, wobei man daraus wieder in Form von Rückschlüssen Erkenntnisse über den Menschen selber gewinnen will. Andere Autoren wieder raten von einer Simulation ab und propagieren eigene Modelle für das Umsetzen von Wissen in Maschinen; als Beispiel führen sie an, daß der Mensch erst dann Erfolg beim Fliegen hatte, als er sich von der Simulation des Verhaltens von flugfähigen Tieren abwandte.

Daten allein repräsentieren noch kein Wissen. Ein Telefonbuch wäre für einen Menschen aus dem vorigen Jahrhundert wertlos; er wüßte nicht, was er mit den vielen Zahlen anfangen könnte. Erst eine Reihe von *Regeln*, die besagen, welchen Sachverhalt diese Daten beschreiben, welche Suchkriterien leicht erfüllbar sind (die Nummer zu einem Namen), welche schwieriger (den Namen zu einer Nummer), wie man die Antwort auf komplexe Probleme aus einfachen Bausteinen aufbaut, machen das Telefonbuch zu dem, was es uns bedeutet.

Wenn nun menschliches Wissen formalisiert in Form von Daten, Regeln, Netzstrukturen oder Situationsbeschreibungen abgelegt wird, so sind neben den daraus erwachsenden Vorteilen die Nachteile nicht zu übersehen. In einem Artikel von Sinding-Larsen (1987) wird die Problematik sehr treffend dargelegt: er spricht von einer *Externalisierung des Wissens*. Im Laufe der menschlichen Entwicklung haben bereits zwei solche Phasen stattgefunden, die entscheidende Veränderungen im menschlichen Selbstverständnis bewirkt haben: die Entwicklung der Sprache und das Entstehen der Schrift (siehe auch Abschnitt 1.1). Der Einsatz der künstlichen Intelligenz ist die dritte solche Phase, die auf uns zukommt. Das Problem besteht darin, daß man über extern

abgelegte Informationen die Kontrolle verliert. Dem Entstehen der Sprache folgte die "babylonische Sprachverwirrung", der Schrift folgte die Informationsexplosion. Die externe Archivierung des Wissens und die daraus resultierende Abhängigkeit von fixen Verwaltungsstrukturen und Normen bringt aber auch einen Verlust an Improvisationsfähigkeit.

Gerade im Bereich der RIS wird eine enorme Fülle von Daten zu erwarten sein, die noch dazu in vielerlei Weise kombiniert werden. Die Verwaltung der Daten könnte zum Selbstzweck werden. Die Fähigkeit zur "Improvisation" (in diesem Fall zum kreativen Kombinieren vorhandener Informationen) könnte verlorengehen, und ebenfalls der "Durchblick": eine zusammenfassende Sicht der Daten, ihres sinnvollen Einsatzes und der verwendbaren Kombinationen fehlt. Unsere Aufgabe wird es sein, diese Probleme der Stufe 2 (Datenexplosion) mit Methoden der Stufe 3 (künstliche Intelligenz) zu lösen. Dabei wissen wir, daß Stufe 3 unweigerlich neue Konflikte auf höherer Stufe heraufbeschwören wird.

Sollte dieser vorangegangene Exkurs beim Leser einen allzu fatalistischen Eindruck entstehen lassen, so widerspricht dies den Intentionen des Verfassers. Es ist ein wesentliches Merkmal des menschlichen Fortschrittes, Probleme zu lösen und danach neue Probleme anzupacken. Die Vision einer konfliktlosen stagnierenden Gesellschaft, in der keine ungelösten Probleme in Sicht sind, ist eher trist. In diesem Sinn kann man unsere Überlegungen mit durchaus positiven Untertönen belegen.

8.2. Formen der künstlichen Intelligenz

Wir finden die verschiedenen Zugänge zu Formen der künstlichen Intelligenz am besten, indem wir die jeweilige Formulierung des Zieles näher durchleuchten: oft läßt sich das Ziel einfach damit festlegen, daß wir a) "den Computer intelligent machen" wollen; wir können aber auch untersuchen, wie wir b) "menschliches Handeln in bestimmten Teilbereichen simulieren" können. Eine weitere Alternative besteht in c) "der Erstellung eines Modelles der menschlichen Intelligenz".

Den Zugang zur künstlichen Intelligenz, welcher der Formulierung a) am besten entspricht, finden wir durch das "Erstellen geeigneter Programme, Entscheidungskriterien und Lernmechanismen". Diese Aspekte sind natürlich auch für die Erreichung des Zieles b) wichtig, aber dort geht es zusätzlich noch um die maschinelle Umsetzung der Entscheidungen; die Forschungen, die in diese Richtung

gehen, werden mit dem Begriff **Robotics** zusammengefaßt. Das Ziel c) schließlich erreichen wir durch den "kognitiven Zugang"; dabei geht es zunächst hauptsächlich darum, daß man die Art und das Zusammenspiel der Prozesse ergründen will, die im Gehirn eines Menschen ablaufen, wenn er Eindrücke verarbeitet, Objekte wiedererkennt und Parallelen zu anderen Erfahrungen feststellt; die dabei gewonnenen Erkenntnisse können dann bei der Erstellung eines Modelles umgesetzt werden; es ergeben sich aber auch interessante Rückschlüsse in bezug auf das Verstehen menschlicher Verhaltensweisen.

8.3. Expertensysteme

Von jenen Bereichen der künstlichen Intelligenz, in denen man bisher die größten Fortschritte erzielt hat, sind sicherlich die *Expertensysteme* als erstes zu nennen. Ein Expertensystem ist auf ein eng abgegrenztes Wissensgebiet ausgelegt (Schnupp u. Leibbrandt 1986). Es legt hochspezialisiertes Wissen in einer Wissensbank (knowledge base) ab. Dieses Wissen besteht im wesentlichen aus elementaren Fakten, sowie aus Regeln bzw. Entscheidungskriterien, die in der Form "wenn A, dann B" vorliegen. Bezüglich des korrekten Gebrauches von Regeln gibt es selbst wieder Vorschriften, die man als **Metaregeln** bezeichnet ("Regeln über den Gebrauch von Regeln"). Fakten und Regeln werden zunächst in Zusammenarbeit mit einem (menschlichen) Spezialisten erarbeitet; aus diesen Grundregeln ermittelt ein **Ableitungsmechanismus** ein "komplexes" Wissen, das der Anwender nicht explizit speichern muß.

Ein Beispiel möge dies veranschaulichen. Regel 1 besagt: "jedes Grundstück gehört zu genau einer Katastralgemeinde". Regel 2 besagt: "jede Katastralgemeinde gehört zu genau einer Gemeinde". Damit ergibt sich, daß sich die Angabe der Regel 3 ("jedes Grundstück gehört zu genau einer Gemeinde") erübrigt, weil der Ableitungsmechanismus in der Lage ist, diese aus den Regeln 1 und 2 zu ermitteln. Es ergeben sich Parallelen zu einer Datenbank; dort werden zunächst elementare Daten gespeichert, etwa Linienstücke, die beim Digitalisieren entstehen. Diese Linienstücke können dann im nachhinein zu komplexen Objekten zusammengefaßt werden.

Der Ableitungsmechanismus wird auch **Problemlösungskomponente** ("inference engine") genannt. Es steht außer Frage, daß sehr viele Entscheidungen, die wir täglich treffen, das Resultat solcher elementarer Regeln darstellen, die wir (meist unbewußt) anwenden. In vielen Fällen bringen wir dann noch unsere Fähigkeit ins Spiel, *Analogien* mit ähnlichen Wissensgebieten zu berücksichtigen. Expertensysteme haben dafür kein Potential; hier ist eher der - oben

erwähnte - **kognitive Zugang** sinnvoller, auf den wir in Abschnitt 8.6 näher eingehen. Kommen dann noch Begriffe wie *Intuition* oder *Gefühl* hinzu, so muß unser Versuch einer Nachahmung natürlich scheitern. Die Vorstellung, daß der Mensch und seine Fähigkeiten in diesem Ausmaß ersetzbar sein könnten, ist unrealistisch (und auch abschreckend). Aus diesem Grund lehnen viele Autoren den Begriff "Künstliche Intelligenz" ab, weil er zu sehr in diese Richtung tendiert.

Die Problemlösungskomponente wird also sicher nicht alle unsere Probleme lösen; wie wir eingangs erwähnt haben, gibt es aber ganz bestimmte eng abgegrenzte Situationen, in denen ein solcher Mechanismus sehr wohl effizient eingesetzt wird. So wurden Expertensysteme bisher erfolgreich in folgenden Bereichen verwendet:

- medizinische Diagnostik (MYCIN),
- Interpretation geologischer Daten (PROSPECTOR),
- Spracherkennungs- und Antwortsysteme (HEARSAY),
- Konfiguration eines Rechner-Betriebssystems (R1),
- sowie in einer Reihe von weiteren Anwendungen im Bereich der *Prognose*, der *Planung* und *Überwachung*, aber auch des *Trainings*.

Dabei müssen wir uns über die Grenzen des Möglichen (und auch des Erstrebenswerten) im klaren sein; so muß etwa bei der medizinischen Diagnostik eine enorme Fülle von Symptomen zu einem konsistenten Krankheitsbild zusammengefügt werden, und die Gefahr ist groß, daß ein Mosaiksteinchen übersehen wird. Ärzte werden dadurch nicht ersetzbar, denn gerade sie brauchen ein hohes Maß an Intuition und Gefühl, um scheinbar identische Krankheitsbilder unterscheiden zu können; das Expertensystem kann ihnen die Entscheidung nicht abnehmen, wohl aber in einzelnen Teilbereichen Klarheit schaffen.

Neben der Problemlösungskomponente muß es - wie bereits erwähnt - einen Modul geben, der die Bereitstellung der Fakten und Grundregeln unterstützt und der auch abgeleitete Regeln aufnehmen kann (Lernmodul oder knowledge acquisition module).

Die Menge der Grundregeln muß

- *in sich konsistent*,
- *vollständig* und
- *minimal* sein.

Die Einhaltung dieser Bedingungen ist nicht einfach. Zunächst muß ein menschlicher Experte in die Lage gebracht werden, sein in langen Jahren angeeignetes Wissen in eine formalisierte Form von Regeln zu bringen, die sequentiell abgearbeitet werden können und die einander nicht widersprechen. Außerdem sollen sie vollständig sein; dies bedeutet, daß möglichst alle Anwendungsfälle bearbeitet

werden können; trotzdem sollen sie sich auf das Wesentliche konzentrieren, denn ein Zuviel würde Redundanzen in die Wissensbank bringen. Wieder ergeben sich Parallelen zur Datenbank, denn auch Datensammlungen müssen konsistent, vollständig und minimal sein.

Neben den Komponenten für die Problemlösung und die Wissensbereitstellung weist ein Expertensystem auch eine **Erklärungskomponente** ("why-how-utility") und eine **Dialogkomponente** (Anwender-Interface) auf. Die Schlußfolgerungen, die durch das Zusammenspiel vieler einzelner Regeln zustande kommen, werden erklärt und dokumentiert, so daß der Anwender die Möglichkeit hat, die Art der Problemlösung zu beurteilen und seine Entscheidungen dementsprechend zu fällen; auch können auf diese Weise Irrwege und inkonsistente Regeln erkannt werden. Eine Veranschaulichung der einzelnen Komponenten eines Expertensystems und ihres Zusammenspiels gibt Abb. 8.1.

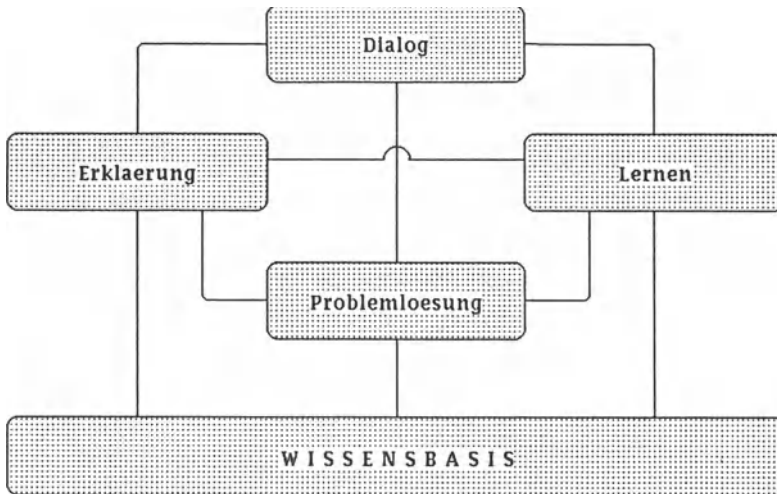


Abb. 8.1. Komponenten eines Expertensystems

Expertensysteme können als eine mögliche Konkretisierung eines **Ersetzungssystems** (production system) betrachtet werden (Appelrath 1985). Ersetzungssysteme stellen eine allgemeine Klasse von Methoden zur Wissensrepräsentierung und Wissensbereitstellung dar und bestehen aus einer Wissensbasis, aus Ersetzungsregeln und einem Kontrollsystem (Nilsson 1982). Ersetzungsregeln sind "wenn...dann..."-Formulierungen. Das Kontrollsystem wählt eine solche Regel aus, prüft das Erfülltsein des "wenn"-Teiles und führt gegebenenfalls den "dann"-Teil aus. Das Wissen wird induktiv bereitgestellt; man beginnt bei einzelnen Mosaiksteinchen und trägt sukzessive immer mehr Wissen

zusammen. Diese Strategie wird daher **forward chaining** oder **data-driven strategy** genannt.

Dazu gibt es als Alternative die **deduktive Strategie (backward chaining oder goal-driven strategy)**, die ein Problem solange in Teilprobleme zerlegt, bis man bei elementaren Fakten angelangt ist, die entweder wahr oder falsch sind und somit eine Beantwortung der Gesamtfrage ermöglichen.

Neben wahren und falschen Aussagen können Expertensysteme aber auch **vages Wissen** verarbeiten. Es handelt sich dabei um Anwendungsbereiche,

- in denen Aussagen nicht immer exakt machbar sind,
- in denen nicht immer alle Aussagen verfügbar sind, und
- in denen man aus Zeitgründen nur einen Teil der Regeln überprüft.

Solche Voraussetzungen treffen gerade auf die in diesem Buch behandelten Problemkreise sehr häufig zu, speziell wenn es sich um Themen wie Ökologie, Klima, Lagerstätten, Prognosen und Projektierungen handelt. Sehr viele Aussagen zu diesen Themen werden mit einer **Wahrscheinlichkeit** behaftet sein. Aussagen wie "das Waldgebiet X ist stark umweltbelastet", "der Einzugsbereich der Stadt Y ist relativ groß", "der Grundwasserspiegel im Gebiet Z sinkt" sind *subjektiv*, und man kann ihnen eine Wahrscheinlichkeit zuordnen. In diesem Fall spricht man auch von **fuzzy data** (ungenauen Daten).

Die anderen beiden Voraussetzungen treffen hier ebenfalls in besonderem Maße zu: aufgrund der vielfältigen Verflechtungen der Themen ist es nur möglich, eine Auswahl von Regeln zu überprüfen; auch wir selbst würden einige "wichtige" Regeln vorziehen, das heißt, wir würden die Regeln unterschiedlich stark gewichten, und wir würden die Überprüfung danach abbrechen. Expertensysteme verfahren nach demselben Schema.

Wahrscheinlichkeitsbewertete Aussagen und Regeln werfen allerdings eine Reihe von Fragen auf, deren Beantwortung zum heutigen Zeitpunkt nur zum Teil möglich ist: Wahrscheinlichkeiten pflanzen sich fort; da man auf unterschiedlichen Wegen zu neuen Aussagen und Regeln kommen kann, ergeben sich - je nach dem eingeschlagenen Weg - unterschiedliche Wahrscheinlichkeiten. Wie behandeln wir diese? Außerdem kann es - wenn nicht alle Aussagen und Regeln unumstößlich sind - zu Widersprüchen kommen; man spricht in diesem Fall auch von **nicht-monotoner Logik** und führt **Hypothesen** ein, die dann entweder *gerechtfertigt* oder *revidiert* werden (Appelrath 1985). Auch unsere eigenen Entscheidungen sind ja letzten Endes von einem ständigen Abwägen widersprüchlicher Hypothesen geprägt. Die Problemlösung wird - entgegen dem ersten Anschein - durch solche

Unsicherheiten und Widersprüchlichkeiten sogar einfacher (Cohen 1987).

Zusammenfassend können wir sagen, daß Expertensysteme gegenüber der Problemlösung mittels herkömmlicher EDV-Methoden eine Reihe von Vorteilen bieten:

- sie entsprechen eher der Denkweise des Anwenders, besonders wenn dieser über wenig EDV-Erfahrung verfügt; in diesem Sinn sind sie besser in der Anwenderschale einsetzbar als herkömmliche Programme (siehe auch Kapitel 7);
- sie können "von außen" (**nicht-prozedural**) definiert werden, brauchen also nicht in ein starres Ablaufschema gepreßt zu werden;
- sie bieten eine Vielfalt von Möglichkeiten, den Anwender beim "Erlernen" der Problematik zu unterstützen: dazu zählen Trainingsprogramme und die Möglichkeit, anhand von Beispielen die eigene Erfahrung zu testen und auszubauen. Auf diese Art können sie uns auch zu einem besseren Verständnis unseres eigenen Verhaltens und unserer Fähigkeiten und Entscheidungen verhelfen.

Wenn wir sie mit der Erfahrung menschlicher Experten vergleichen, so bleibt ihr rationeller Einsatz auf Erfahrungsbereiche vom Typ **deep and narrow** ("tief und schmal", also wenige Aspekte, diese aber detailliert) beschränkt, während sie in **wide and shallow**-Bereichen ("weit und seicht"; also Übersichten) nicht so gut abschneiden (Basden 1984). In diesen deep and narrow-Bereichen können sie jedoch einige Vorteile für sich buchen:

- sie sind *sicher*, wenn Entscheidungen unter Zeitdruck gemacht werden müssen,
- sie sind *konsistent* in den Details,
- sie spielen in kurzer Zeit viele *Varianten* durch,
- sie sind jederzeit *verfügbar* und *speichern* das Wissen, auch bei einem fluktuierenden Mitarbeiterstand, und
- sie erlauben eine einfache *Weitervermittlung* des Wissens.

8.4. Wissensgestützte Informationssysteme

Nach der im letzten Abschnitt gegebenen allgemeinen Definition eines Expertensystems und seiner Funktionalität wenden wir uns nun der Frage zu, wie es sich in den Themenkreis raumbezogener Informationssysteme einordnen

läßt. Ein Informationssystem kann als einfachste Ausprägung eines Expertensystems angesehen werden. Die Wissensbasis besteht aus den Daten und den geometrisch-topologischen sowie thematischen Strukturen, die den Daten auferlegt werden. Die Problemlösungskomponente wird durch die im System verfügbaren Algorithmen realisiert. Dasselbe gilt für die Komponente, die das Grundwissen und das daraus abgeleitete Wissen bereitstellt (Brauer u. Wahlster 1987).

Als Beispiel kann uns die Flächenbildung dienen, die auf eine Menge von digitalisierten Kanten angewendet wird und in vielen Systemen zum Großteil automatisierbar ist. Es gibt nun einige elementare Regeln, die etwa besagen,

- daß Flächen von geschlossenen Polygonen umrahmt werden,
- daß diese Polygone keine Schleifen haben dürfen,
- daß Flächen auch Inseln haben können,
- daß es Schranken für den Flächeninhalt und/oder den Durchmesser gibt, und daß wir damit extrem kleine, extrem langgezogene oder extrem schmale Flächen ausschließen möchten,
- daß es Toleranzbereiche für die Punktlage gibt, so daß Kantenpunkte, die "beinahe" übereinstimmen, zusammengelegt werden usw.

Wenn diese Regeln nun zur Flächenbildung herangezogen werden, so erhöht sich damit nach Anwendung der Regeln der "Wissensstand"; während zunächst nur Kanten vorlagen, gibt es nun komplexe Strukturen, eben Flächen. Nun kann man wieder Regeln definieren, die einzelne Flächen miteinander vergleichen und überprüfen, ob sie überlappend bzw. flächendeckend angeordnet sind. Ein Grundstücksplan etwa muß solche Bedingungen erfüllen. Wir haben in Kapitel 3 diese und ähnliche **Konsistenzbedingungen** aufgezählt. Ein raumbezogenes Informationssystem muß definitionsgemäß für ihre Einhaltung sorgen und hat somit einige wichtige Eigenschaften eines Expertensystems. Trotzdem unterscheidet es sich in wesentlichen Punkten:

a) der Grad der Komplexität für das Spektrum der zu beantwortenden Fragen ist deutlich geringer als bei "echten" Expertensystemen. Ein Informationssystem erlaubt es im allgemeinen nur, eine beschränkte Anzahl von Situationen zu definieren, die im Rahmen einer algorithmischen Behandlung erkannt und verändert werden können. Meist beziehen sich diese Situationen auf Daten der Datenbank oder auf Strukturen, die explizit als solche in der Datenbank abgelegt sind (allenfalls als logische Views).

b) die Regeln sind in einem Informationssystem herkömmlicher Bauart in Form von Algorithmen - also *prozedural* - abgelegt, während Expertensysteme oft eine *nicht-prozedurale* Speicherung der Regeln erlauben. Dies erleichtert die Anpassung an das Anwenderprofil ungemein, da man auf diese

Weise optimal auf Situationen reagieren kann, die bei der Erstellung des Systems im Detail noch nicht bekannt waren. Allerdings braucht man dazu auch die Elemente einer Sprache, die einen solchen nicht-prozeduralen Zugang ermöglicht, z.B. LISP, PROLOG oder C.

c) die Bereitstellung von neuem Wissen beschränkt sich auf das Einfügen oder Verändern von Daten in der Datenbank. Das Einfügen neuer Regeln jedoch ist nicht möglich, und schon gar nicht ihre automatische Ableitung aus bestehenden Regeln oder die Überprüfung auf ihre Verträglichkeit mit bestehenden Regeln. Wir wollen also nicht nur die Konsistenz von Daten, sondern auch die Konsistenz von Regeln untersuchen; dies ist auch deshalb wichtig, weil Regeln nicht immer eindeutig sind und sich auch gegebenenfalls widersprechen (siehe Abschnitt 8.3).

d) die Erklärungskomponente fehlt: das System begründet im allgemeinen nicht seine Aktionen; dies hängt mit der beschränkten Anzahl überprüfbarer Situationen und mit der Einfachheit der dort zu fällenden Entscheidungen zusammen: es braucht nicht näher begründet zu werden, warum eine Reihe von Kanten als Umriß einer Fläche anzusehen ist; ein Blick auf die Kontrollzeichnung genügt.

e) die Dialogkomponente fehlt ebenso: viele Expertensysteme erlauben nämlich die Anpassung der Anwenderschnittstelle an die **Natürliche Sprache** des Anwenders. Dabei meint man nicht so sehr die phonetische Erkennung des gesprochenen Wortes, sondern eine maßgeschneiderte Schnittstelle, die bezüglich ihrer Detailliertheit auf das Anwenderprofil ausgerichtet ist (siehe dazu Abschnitt 8.7).

f) Datenbestände sind in einem Informationssystem vollständig und präzise vorhanden. "Ungenau" Informationen fehlen, sofern es sich nicht um implizite topologische Nachbarschafts- und Überlappingsrelationen handelt. Expertensysteme hingegen können vages Wissen verarbeiten (siehe Abschnitt 8.3).

Zusammenfassend können wir festhalten, daß einige der Aspekte, die wir eben als charakteristisch für Expertensysteme erkannt haben, sehr wohl in unserer allgemeinen Definition eines raumbezogenen Informationssystems sichtbar werden; konkrete Systeme gehen unterschiedlich weit bei der Berücksichtigung dieser Aspekte. Trotzdem kann das Expertensystem als Erweiterung, als zusätzliche Schale eines Informationssystems gesehen werden (**expert system shell**), die vereinzelt sogar schon käuflich zu erwerben ist. Die Lage jener Grenzlinie, bei der das Informationssystem endet und das "echte" Expertensystem beginnt, kann natürlich nicht klar gezogen werden. Ein solches System, das die Aspekte eines raumbezogenen Informationssystems mit den Aspekten eines

Expertensystems vereinigt, bezeichnet man auch als wissensgestütztes RIS (knowledge-based geographical information system oder KBGIS; siehe dazu auch Peuquet 1984).

Viele Aufgaben eines KBGIS können "ohne Expertenwissen" bzw. mit einem "minimalen Expertenwissen" (etwa mit Werkzeugen, die wir in Kapitel 6 besprochen haben) durchgeführt werden; dazu gehören beispielsweise räumliche und topologische Abfragen und Verschneidungen. Das Erkennen und die Ausnützung thematischer Verflechtungen hingegen setzt eher ein höheres Expertenwissen voraus. In einem solchen Modell eines wissensgestützten Informationssystems zerfällt die Problemlösungskomponente in einen Teil, der für räumliche und topologische Aspekte verantwortlich ist, und in einen Teil, der thematische Belange abdeckt (Abb. 8.2).

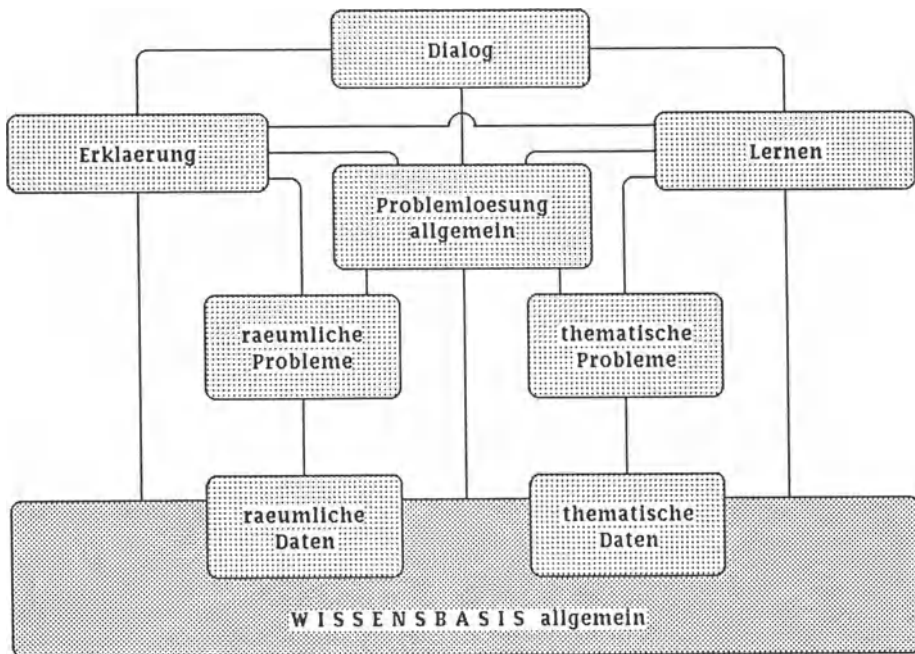


Abb. 8.2. Wissensgestütztes Informationssystem

8.5. Graphentheoretische Konzepte: UND-ODER-Graphen

Als Alternative - oder besser als Ergänzung - zu der in den Abschnitten 8.3 und 8.4 erläuterten regelgestützten Vorgangsweise bietet sich die Modellierung des Wissens mittels eines Graphen an. Wir können beispielsweise ein gesamtes Thema mit allen seinen zugehörigen Facetten als eine Baumstruktur darstellen (die Baumstruktur wurde in Kapitel 3 erklärt). Ein Beispiel für eine solche Baumstruktur ist in Abb. 8.3 für das Thema "Gewässer" dargestellt.

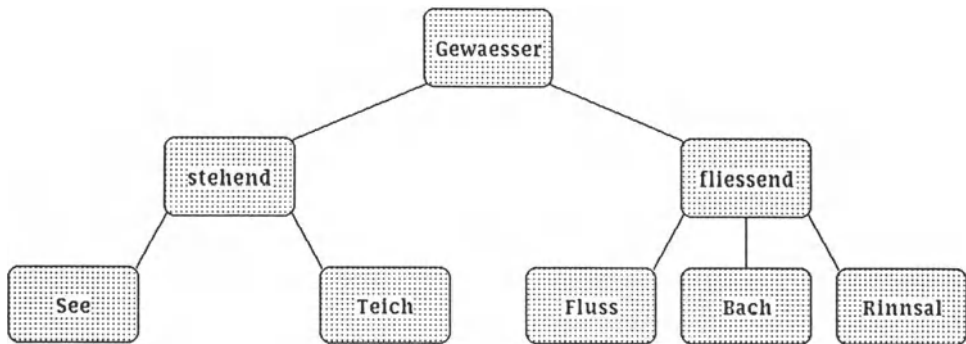


Abb. 8.3. Wissensrepräsentierung durch eine Baumstruktur (OR-TREE)

Dieser Art der Darstellung liegt eine sukzessive Verfeinerung der thematischen Bedeutung zugrunde. Für eine *Realisierung*, d.h. ein tatsächliches Objekt, wird dann ein bestimmter Zweig bis zu seiner letzten Verästelung durchschritten. Wir haben in dieser Abbildung ein besonders einfaches Beispiel gewählt, und zwar die Auffächerung eines Themas, wie wir sie schon in Kapitel 3 besprochen haben. Es können die Zwischenknoten eines solchen Baumes aber genauso gut komplexe Aussagen oder Regeln sein, z.B:

Ein Leitungsrohr ist erneuerungsbedürftig,
 WENN es vor mehr als 10 Jahren verlegt wurde,
 ODER WENN sein Durchmesser geringer als 3 cm ist.

Die strenge Baumstruktur bildet eher eine Ausnahme; im allgemeinen werden sich einzelne Zweige wieder verknoten, so daß ein allgemeiner Graph entsteht. Davon können wir uns leicht überzeugen, wenn wir die obige Formulierung umdrehen. Es kommt also darauf an, ob wir von der Tatsache ausgehen, daß das Rohr älter als 10 Jahre ist, und daraus schließen, daß es erneuert werden muß (*forward chaining*), oder ob wir die Hypothese aufstellen, daß eine Erneuerung fällig ist,

und prüfen, ob sie verworfen oder untermauert wird; das heißt, daß wir ein komplexes Problem in einfachere Teilprobleme aufspalten (*backward chaining*).

In diesem Beispiel stellt die Verzweigung an einem Zwischenknoten des Graphen eine Entscheidung vom Typ "ODER" dar; es gibt natürlich auch "UND"-Verknüpfungen: die Gewässergüte hängt von der Fließgeschwindigkeit UND von der Menge der eingeleiteten Abwässer ab. Man bezeichnet solche Graphen als **UND-ODER-Graphen**. Sie stellen eine Standardmethode zur Wissensrepräsentierung dar (Nilsson 1982).

8.6. Der kognitive Zugang: Semantische Netze und Frames

Wir haben bereits eingangs darauf hingewiesen, daß Expertensysteme nur in eng abgegrenzten Wissensgebieten sinnvoll eingesetzt werden können, und auch nur dort, wo man das Wissen in eine Menge von *Regeln* abbilden kann, die dann *sequentiell* durchlaufen und getestet werden. Dieser Vorgang weist formale Ähnlichkeiten zu der Ausführung eines Algorithmus auf, der mit herkömmlichen Methoden programmiert wurde - obzwar natürlich die Möglichkeit, die Regeln in nicht-prozeduraler Form angeben zu können, eine wesentliche Verallgemeinerung darstellt und auch andere Eigenschaften ein Expertensystem klar von einem gewöhnlichen Softwarepaket abheben (siehe Abschnitt 8.4).

Trotzdem schränkt uns die Notwendigkeit, eine Regel nach der anderen zu testen, auf eine *eindimensionale* Welt der Wissensbereitstellung und -veränderung ein. Ohne Zweifel verlaufen auch viele menschliche (körperliche wie geistige) Tätigkeiten sequentiell. Kochrezepte, Bedienungsanleitungen, geschäftliche Transaktionen, Kontobewegungen und vieles andere mehr wickeln wir nach dem "WENN...DANN..."-Muster ab. Auch beim Schreiben eines Sachbuches versucht der Autor mit mehr oder weniger Erfolg, sein Wissen in eine sequentielle Form abzubilden, die nur durch Querverweise, Literaturangaben und Stichwortverzeichnisse unterbrochen wird.

Wenn wir jedoch unsere Ansprüche etwas höher schrauben und komplexe menschliche Verhaltensmuster zum Vorbild nehmen, so sehen wir deutlich, daß die meisten gedanklichen Prozesse nicht eindimensional ablaufen. Gerade die Fähigkeit des Menschen, Assoziationen auszunützen, Analogien und Ähnlichkeiten mit früheren Erfahrungen zu beachten, zeichnet ihn als intelligentes Wesen aus. Man spricht in diesem Zusammenhang auch vom **kognitiven Zugang** zur künstlichen Intelligenz; er stellt den Versuch dar, den Wahrnehmungs- und Erkennungsprozeß besser verstehen zu lernen, und dafür ein **Modell** zu erstellen, das EDV-gerecht behandelt werden kann.

Eine Möglichkeit, ein Modell für die Darstellung dieses komplexen Wissens zu bilden, ist durch das **semantische Netz** gegeben. Alle Informationen bzw. Konzepte der Wissensbasis liegen in den **Knoten** ("concept nodes") des Netzes, die untereinander über gewichtete **Kanten** verbunden sind, so ähnlich wie wir unterschiedlich starke Assoziationen beim Hören eines Musikstückes, beim Wahrnehmen eines Geruches, bei optischen Eindrücken entwickeln. So ist beispielsweise der Knoten, der die Information "TIER" trägt, mit einem Knoten "HUND" verbunden. Je nach unseren früheren Erfahrungen kann nun "HUND" unterschiedlich stark mit den Knoten "TREUE", "FELL" oder "BEISSEN" verbunden sein. Die Kante kann also Beziehungsarten "IST EIN", "IST EIN BEISPIEL VON", "IST ÄHNLICH ZU" usw. ausdrücken. Die Beziehungen zwischen den einzelnen Knoten geben Anlaß zur Bezeichnung **Konnektionismus** (*connectionism*, siehe Partridge 1987).

Wenn nun die Beziehung zwischen zwei unterschiedlichen Eindrücken (etwa "ITALIEN" und "SONNE") untersucht werden soll, so beginnt man bei diesen beiden Knoten und verfolgt *simultan* die davon ausgehenden Kanten. Die sich dabei ausbreitenden Wellen klingen mit fortschreitender Zeit und mit der Weglänge ab, und die Gewichtung der durchlaufenen Kanten erzeugt eine zusätzliche Dämpfung. Kreuzen sich die beiden Wellen in einem oder in mehreren Knoten, so entstehen dort Assoziationen, deren Stärke sich aus der verbleibenden Kraft der beiden Wellen zusammensetzt. Natürlich bricht man den Vorgang nach einem endlichen Zeit- oder Wegintervall ab. Diejenigen Knoten, die von beiden Wellen erreicht wurden, werden markiert; sie ergeben ein *Muster*, das mit anderen Erfahrungswerten kombiniert werden kann. Damit kommen wir zum Problembereich der **Mustererkennung**, und auch Parallelen zu den in weiterer Folge behandelten *Frames* können gezogen werden.

Man kann sich vorstellen, daß mit den heutigen Mitteln an Hardware und Software nur eine sehr beschränkte Zahl von Beziehungsarten und nur eine geringe Reichweite bei der Wellenausbreitung berücksichtigt werden kann. Dieses Modell ist daher zur Zeit noch kaum im praktischen Einsatz. Die Entwicklung von immer leistungsfähigeren Parallelprozessoren läßt jedoch den Schluß zu, daß semantische Netze zur Wissensrepräsentierung in der Zukunft doch einiges an Bedeutung gewinnen werden.

Neben semantischen Netzen stellen **Frames** eine weitere anspruchsvolle Methode zur Wissensrepräsentierung dar. Der Name "Frame" (Rahmen) sagt bereits, daß es sich um eine Sammlung von "Rahmenbedingungen" handelt, die aus früheren Situationen bekannt sind und die man mit der gegenwärtigen Situation vergleicht. Ein Beispiel dafür - zugleich auch eine erfolgreiche Implementierung eines Rahmenmodelles - ist die (relativ kleine) Menge von Situationen, die am Buchungsschalter einer Fluggesellschaft auftreten können

(Trapp 1986): die Neubuchung, Umbuchung oder Stornierung eines Tickets von seiten des Kunden, die Mitteilungen von Verspätungen und Einschubflügen von seiten der Fluggesellschaft usw.

Auch der Mensch denkt oft in solchen Rahmen. Wir vergleichen unsere aktuelle Situation mit einer früheren ("das habe ich schon einmal erlebt"), wobei hier wieder exakte und vage Vergleiche möglich sind. Die Mächtigkeit einer solchen Vergleichsmöglichkeit kommt sehr gut zum Ausdruck, wenn wir uns wieder das Schreiben eines Sachbuches vor Augen halten: dies ist zwar ein sequentieller Vorgang; die Vermittlung der Wissensinhalte wird jedoch entscheidend durch das Einstreuen von *Beispielen* verbessert. Ein solches Beispiel spricht genau die Fähigkeit des menschlichen Gehirns an, das aktuelle Problem in einen von mehreren möglichen Rahmen einordnen zu können.

Ein solcher Rahmen besteht aus mehreren Slots, die zu gegebener Zeit mit Instanzen gefüllt werden können. Wir haben diese Ausdrücke bereits in Abschnitt 3.5 kennengelernt. Dort besprachen wir die Zuordnung von thematischen Attributen zu Objekten. Für das Objekt "Grundstück" gibt es einen Rahmen von möglichen Attributen (Slots), etwa den Eigentümernamen, die Grundstücksnummer usw. Für eine konkrete Realisierung aus der Gruppe der Grundstücke werden diese Slots zum Teil mit Instanzen gefüllt: "Hans Meier", "Nr. 171/3" usw.

Dieses Konzept können wir also auf Gesamtsituationen verallgemeinern. Bezeichnenderweise werden dafür auch Begriffe *Script* (Drehbuch), *Scene*, *Role* (Personen, die an der Szene beteiligt sind), *Prop* (Objekte, die an der Szene beteiligt sind), u.a.m. verwendet.

Slots und Frames können auch vernetzt werden, so daß sich hier eine Querverbindung zu den früher besprochenen semantischen Netzen ergibt. Für die derzeit gegebenen Möglichkeiten einer praktischen Umsetzung des Rahmenmodelles gelten dieselben Überlegungen, wie wir sie vorhin für Netze formuliert haben. Gegenwärtig ist eine Realisierung nur unter bestimmten eingeschränkten Voraussetzungen machbar. Die rasante Entwicklung auf dem Gebiet der Parallelprozessoren wird jedoch auch hier bald neue Perspektiven eröffnen, und wir sollten dann als Anwender mit unseren Konzepten so weit sein, daß wir die Möglichkeiten, die uns diese neue Hardware bietet, auch ausschöpfen können.

8.7. Derzeitige und künftige AI - Anwendungen

in raumbezogenen Informationssystemen

Von den derzeit bekannten Anwendungen von AI-Konzepten greifen wir einige heraus, die sich für den Einsatz in einem RIS eignen.

8.7.1. Erkennen von Strukturen

Dieser Überbegriff (im Englischen *feature analysis*, *feature extraction*) umspannt eine ganze Reihe von Problemen, die sowohl bei der Bearbeitung von Luft- und Satellitenbildern wie auch bei der Nachbehandlung von Scannerdaten auftreten. Solche Bilder liefern Rasterinformationen, und die dabei anfallenden geometrischen Daten können relativ problemlos übernommen werden. Schwieriger ist es, den Prozeß der Themenzuweisung und der thematischen Strukturierung zu automatisieren. Es gibt einige Ansätze dazu.

So kann man etwa in Satellitenbildern eine radiometrische Klassifikation der Landnutzung durchführen (Monmonier 1982). Man macht sich dabei den Umstand zunutze, daß Ackerland, Grünland oder Gewässer jeweils unterschiedliche Reflexionseigenschaften aufweisen. Wenn wir etwa, wie in Abb. 8.4 die Reflexion in Abhängigkeit von der Wellenlänge betrachten, so ergeben sich für verschiedene Landnutzungen jeweils verschiedene Kurven, die sich besonders im Infrarot-Bereich stark unterscheiden: dadurch wird es möglich, diese Kurve als Charakteristikum für die jeweilige Landnutzung anzusehen, oder, mit anderen Worten: jede Landnutzung hinterläßt ihre "Unterschrift" im Spektrum.

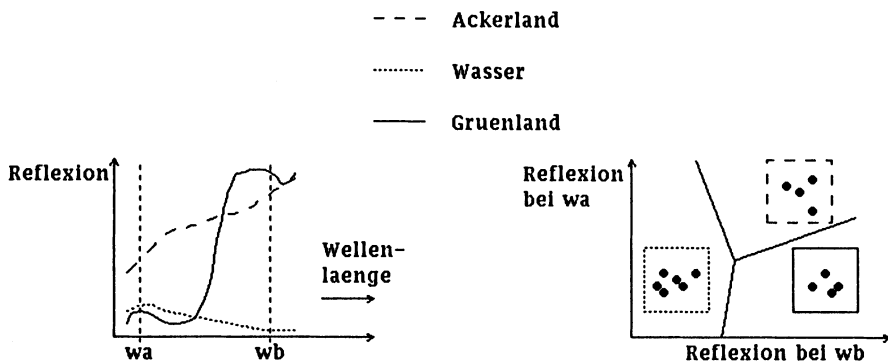


Abb. 8.4. Spektralunterschrift verschiedener Landnutzungen

Nun ist es notwendig, anhand von "Trainingsgebieten" diese Charakteristika festzulegen. In unserer Terminologie bedeutet dies, daß wir für jede Landnutzung einen Rahmen definieren; die aktuellen Daten werden dann mit den Rahmen verglichen, und jener Rahmen wird bestimmt, der am besten paßt. Natürlich passen die Rahmen im allgemeinen nicht exakt, so daß wir *vage* und *nicht-monotone* Techniken anwenden müssen, die wir in Abschnitt 8.3 kennengelernt haben.

Diese grobe Einteilung läßt sich hiernach noch verfeinern: wenn wir Brücken erkennen wollen, so müssen wir unser "Expertenwissen" formalisieren: eine Brücke verbindet immer zwei Landgebiete, die durch ein Gewässer getrennt sind, oder zwei Landgebiete über ein tiefer gelegenes Gelände hinweg. Sie muß gewisse geometrisch-topologische Bedingungen erfüllen; sie darf nicht als Vegetationszone klassifiziert sein. Ihr Auftreten ist in der Nähe von verbauten Gebieten viel wahrscheinlicher als in ländlichen Gebieten usw. Alle diese Regeln können von einem Expertensystem ausgewertet werden (siehe Lambird u. Lavine 1984); die obige frame-orientierte Methode der Klassifikation wird also durch ein Expertensystem unterstützt. Sie läßt sich auch für die Ermittlung von Vegetationsschäden anwenden. Geschädigte Wälder haben andere "Unterschriften" als gesunde Wälder (Reiner 1988); allerdings muß man auch beachten, daß der jahreszeitliche Zyklus das Reflexionsvermögen sehr stark beeinflußt, ebenso gelegentliche Dürre- und Regenperioden. Die dabei auftretende Fülle von Nebenbedingungen, die einander gegenseitig beeinflussen, eignet sich besonders gut für die Behandlung durch ein Expertensystem.

Die Strukturerkennung ist nicht nur für die Nachbehandlung von Rasterinformationen wichtig. Auch die Auswertung von Vektordaten kann durch automatische Strukturanalysen verbessert werden. So kann man etwa aus einer Karte von Gewässerlinien Informationen über Wasserscheiden, Täler und Gebirgskämme extrahieren (Palmer 1984); die Grundlage dafür bietet ein System von Regeln.

8.7.2. Datenkonsistenz

Wir haben in den vorangegangenen Kapiteln sehr viel über die Datenkonsistenz gesprochen; ihre Bedeutung für die Rolle eines raumbezogenen Informationssystems als Langzeitspeicher wurde immer wieder betont. Auch haben wir stets die Fülle von Bedingungen, die dabei zu beachten sind, hervorgehoben. Die Einbettung all dieser Regeln in ein Expertensystem entlastet die Datenbankverwaltung (DBMS) von all diesen Überprüfungen. Das DBMS sorgt nur mehr für die Konsistenz elementarer Daten (der Tupel, die physisch gespeichert oder allenfalls über einen View verbunden sind). Die Konsistenz

komplexer Zusammenhänge übernimmt das Expertensystem. Ein Beispiel möge dies veranschaulichen:

Typischerweise treten in einem raumbezogenen Informationssystem Themen auf, die eine administrative Aufteilung des Datenraumes in Teilflächen beschreiben; diese Aufteilung setzt sich meist hierarchisch über mehrere Stufen fort. Ein Bezirk zerfällt in Gemeinden, eine Gemeinde in Katastralgemeinden, eine Katastralgemeinde in Grundstücke. Jede einzelne Teilfläche muß dem geometrisch-topologischen Anspruch einer Fläche genügen; aber auch im hierarchischen Gesamtverbund muß sie bestimmten Regeln gehorchen: es dürfen keine Überschneidungen vorkommen, aber auch kein "Niemandland", keine Splitterflächen und ähnliches (Frank 1983).

Neben diesen geometrisch-topologischen Bedingungen müssen auch noch thematische Bedingungen eingehalten werden: so darf man zwar eine Menge von aneinandergrenzenden Grundstücken zu einer Gemeinde zusammenfassen, nicht aber aus mehreren Gemeinden ein Grundstück bilden.

Ein wissensgestütztes Informationssystem (KBGIS) ist in der Lage, solche auf Regeln basierende Bedingungen zu beachten. Gleichzeitig kann es auch noch eine *Lernfähigkeit* entwickeln: es führt eine Statistik über die Aktivitäten des Anwenders mit und baut besonders häufig benutzte Pfade aus, indem es für diese eigene Zugriffe schafft. So werden in jedem raumbezogenen Informationssystem zunächst geometrisch-topologische Zugriffspfade häufiger beschriftet werden; allmählich steigt die Zahl der thematischen Zugriffe an, und das System paßt sich daran an, indem es immer mehr thematische Konzepte speichert - es lernt also aus der Anwendung (Peuquet 1984).

8.7.3. Kartengestaltung

Zur optisch ansprechenden Gestaltung von Karten gehören (unter anderem) Schritte zur

- Text- und Symbolplatzierung,
- Text- und Symbolfreistellung,
- Generalisierung,
- Einhaltung geometrischer Bedingungen und
- Lösung von Verdrängungs- und Prioritätskonflikten.

Aus dieser Liste - die nur einen Auszug darstellt - wollen wir die (automatische) Textplatzierung als charakteristisches Beispiel näher durchleuchten. Dabei geht es darum, daß

- Texte der Geometrie eindeutig zugeordnet werden,
- Texte möglichst wenig Hintergrundinformation verdecken,
- Texte sich gegenseitig nicht verdecken.

Außerdem sollen Beschriftungen von Punkten - wenn möglich - nördlich, nordöstlich oder östlich der Punkte angebracht werden, und die Schrift soll in der Nähe des Punktes beginnen. Linienbeschriftungen (z.B. Flußnamen) sollen sich gegebenenfalls der Geometrie anpassen, aber nicht vertikal oder gar "auf dem Kopf" stehen. Flächenbeschriftungen sollen zentriert in die Fläche gesetzt werden. Die einzelnen Buchstaben sollen möglichst wenig durch die geometrische Hintergrundinformation getrennt werden, und die Beschriftung soll sich gleichmäßig auf die gesamte Karte verteilen. Wieder ergeben sich hier eine ganze Reihe von Regeln, und wieder gibt es wichtigere und unwichtigere, und sie können auch zu Konflikten führen. Der Einsatz eines Expertensystems lohnt sich daher (Freeman 1984).

8.7.4. Natürliche Sprache

Diesem Zweig der AI wird allgemein eine sehr große Bedeutung zugemessen (Habel 1984). Es geht dabei nicht (oder nicht nur) um die Interpretation des gesprochenen Wortes. In erster Linie ist das Problem zu lösen, wie einem Anwender, der ein Informationssystem benutzen möchte, die Kommunikation erleichtert werden kann, und wie man ihm das Erlernen einer formalen Sprache abnimmt. Dies ist eine enorme Hürde, die sich in einer oft-gehörten Formulierung äußert: "Computer sind ja ganz interessant, aber meine spezifischen Probleme können sie nicht lösen".

Ein wichtiger Schritt in diese Richtung wurde schon mit der Einführung der Expertensysteme getan. Es ist nicht mehr notwendig, einen Algorithmus *prozedural* vorzuplanen; vielmehr kann man zu jeder Zeit Regeln in der Wissensbasis ablegen und vorhandene Regeln erweitern oder umbauen. Regeln können wahrscheinlichkeitsbewertet werden; Widersprüche können erkannt, verarbeitet und ausgenutzt werden. In der Definition eines Expertensystems kommt auch die Dialogkomponente vor, und diese steht derzeit im Mittelpunkt unseres Interesses. Je besser es uns gelingt, diese Komponente dem jeweiligen Benutzer anzupassen, desto effizienter wird die Ausnutzung des Systems. In dieser Sparte sind die Erfordernisse eines raumbezogenen Informationssystems durchaus allgemeiner Natur.

Die *syntaktische und semantische Analyse* von Elementen der natürlichen Sprache soll den Kontext von Sätzen erfaßbar machen. Viele Wörter haben Mehrfachbedeutungen, und erst

aus dem Zusammenhang heraus verstehen wir die jeweils "richtige" Bedeutung. Außerdem geht es darum, subtile Unterschiede und Verbindungen zwischen Wörtern wie "fließen", "schütten", "träufeln" zu erkennen und auszunützen. Ansatzpunkte für solche Studien gibt es bereits, wenn auch wieder nur in eng abgegrenzten Bereichen: die automatische Übersetzung von Gebrauchsanweisungen oder Spezifikationen vom Englischen ins Deutsche wird - teilweise, und mit unterschiedlichem Erfolg - durch solche Methoden unterstützt (Trost et al. 1987).

8.7.5. Anwenderprofil

Da wir darauf Wert legen, daß sich die EDV den Bedürfnissen des Menschen unterordnet (und nicht umgekehrt), ist es wichtig, daß für jeden potentiellen Anwender eine Beschreibung angelegt wird, die auf seine Erfordernisse - aber auch auf seine Fähigkeiten - eingeht. Kapitel 7 war der Sicht gewidmet, die der Anwender von einem Informationssystem entwickelt. Hier geht es darum, daß das System eine Sicht des Anwenders entwickelt. So müssen etwa für einen "Anfänger" andere, ausführlichere Systemaufforderungen, Fehlermitteilungen und Hilfestellungen angeboten werden als für einen versierten Benutzer, der möglichst schnell zu Resultaten kommen will und gar nicht mehr auf die Systemreaktionen achtet.

Natürlich steigt der Anfänger mit der Zeit in den Status eines professionellen Anwenders auf, und das Anwenderprofil hat dem Rechnung zu tragen. Dazu gehört auch, daß das System die Aktionen des Anwenders *protokolliert*. Besonders häufige Fehler können dann die Einblendung der jeweils wichtigen Seite aus dem Benutzerhandbuch auslösen. Häufig benützte Zugriffspfade werden erkannt und gegebenenfalls durch die Bereitstellung besonderer Zugriffsmechanismen unterstützt. Ein erster - wenn auch bescheidener - Schritt in diese Richtung besteht bereits in dem Paging, das ein virtuelles Betriebssystem automatisch durchführt.

Eine weitere Anwendung des Protokolles (*Logfile*) besteht in der Bereitstellung eines *Logfile-Interpreters*, der es ermöglicht, die mitprotokollierten Aktionen eines Anwenders gegebenenfalls zu wiederholen. Dazu gehört auch die Möglichkeit, daß der Anwender eine exemplarische Transaktion an einem Objekt durchführt und daraufhin eine Reihe von weiteren Objekten spezifiziert, die derselben Transaktion unterworfen werden sollen. Das System "lernt" anhand des ersten Objektes, was zu tun ist, und nützt dieses Wissen bei den folgenden Objekten aus. Mit Ausdrücken der AI formuliert, bedeutet dies, daß wir ein *Script* (Drehbuch) vorgeben, das dann für alle weiteren Objekte bindend ist.

8.7.6. Training

Diesem Aspekt wird - zu Recht - eine besonders große Bedeutung beigemessen. Raumbezogene Informationssysteme sind vielfach erst in der Konzeptphase oder im Aufbau begriffen; aus diesem Grund sind sie derzeit nur für einen relativ kleinen Personenkreis von "Insidern" zugänglich. Andererseits werden sie in der Zukunft unerläßliche Instrumente in der Planung, in der Kontrolle und in der Entscheidung sein. Es ist daher wichtig, daß eine möglichst breite Gruppe von Anwendern schnell Zugang zu den Daten und Methoden eines solchen Informationssystems gewinnt. Dies geschieht am besten anhand von Beispielen, die man erarbeitet.

Die Verwendung von computergestützten Methoden (**computer-based training, advice-giving systems**) bringt dabei eine Reihe von Vorteilen (Green 1987):

- Die Grundkenntnisse können gleichzeitig an viele Interessenten weitervermittelt werden; dies bedeutet nicht, daß der Mensch als Lehrer ersetzt wird: er spürt vielmehr eine Entlastung von Standardaufgaben und kann sich somit besser auf Detailprobleme seiner Schüler konzentrieren.
- Jeder Schüler kann sein eigenes Tempo und seine eigenen Interessensgebiete bestimmen.
- Durch den Einsatz graphischer Methoden können sehr viele Sachverhalte in raumbezogenen Informationssystemen anschaulich dargestellt werden.
- Schwachstellen im Trainingsprogramm oder überhaupt im gesamten Systemkonzept (etwa zeitliche Engpässe) können systematisch erfaßt und an die Systemgruppe rückübermittelt werden.

8.7.7. Anwenderspezifische Techniken

Stellvertretend für ein ganzes Spektrum möglicher Anwendungen sei hier ein Einsatzgebiet aus dem Bereich des Umweltschutzes erwähnt (Uhlenbruck u. McLaughlin 1987): die Qualität des Wassers aus Grundwasserbrunnen ist vielerlei Gefahren durch Umwelteinflüsse ausgesetzt. Die Entscheidung, ob und in welchem Maße ein Brunnen geschützt werden muß, hängt von einer Reihe von Fragen ab: die Beschaffenheit des Bodens, die Entfernung des Brunnens von Deponien, die als gefährlich eingestuft sind, von der Wassermenge, die der Brunnen abgibt, sowie vom Ausmaß seiner Nutzung als Trinkwasserreservoir. Die dabei zu beachtenden Querverbindungen und Abhängigkeiten können wieder in eine Sammlung von Regeln abgebildet werden, die von einem Expertensystem ausgewertet werden.



ANHANG A
LITERATUR

- Akima, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. Journal of the Association for Computing Machinery, vol.17, no.4, Oktober 1970, pp.589-602.
- Andresson, S.: The Swedish Land Data Bank. Int. J. Geographical Information Systems, vol.1, no.3/1987.
- Appelrath, H.J.: Von Datenbanken zu Expertensystemen. Informatik-Fachberichte 102, Springer, Berlin Heidelberg New York Tokyo 1985.
- Appelt, G.: Das AdV-Vorhaben "Amtliches Topographisch-Kartographisches Informationssystem (ATKIS)" - Technische Konzeption. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 99/1987.
- Barnsley, M.F., A.D. Sloan: Chaotic Compression. Computer Graphics World, November 1987, pp. 107 ff.
- Bartelme, N., W. Kolb, K. Seiffter-Bartsch, B. Späni: IMAGE-Interaktive Manipulation von Geo-Elementen. Zeitschrift für Vermessungswesen, Heft 6, Juni 1986.
- Bartelme, N.: Raumbezogene Datenbanken und Interaktive Graphik. Beitrag veröffentlicht in: Folge 64 der Mitteilungen der geodätischen Institute der TU Graz, 1988.
- Bartelme, N., B. Späni: Spatial Access to Thematic and Topological Structures in Geo-Information Systems. Vortrag anlässlich des 16. ISPRS-Kongresses in Kyoto, Japan, 1.-10. Juli 1988.
- Basden, A.: On the application of expert systems. In: M. J. Coombs (ed.): Developments in expert systems. Academic Press, London New York 1984.
- Baumann, M., J. Friedl, M. Habarta, G. Otepka: Der Einsatz eines CAD-Systems bei einer landesweiten Waldzustandserhebung. In M. Schilcher (Hrsg.): CAD-Kartographie. Anwendungen in der Praxis. Wichmann, Karlsruhe 1985.
- Berger, A.: Entwicklung "Elektronischer Seekarten". Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 97/1986.
- Brassel, K., M. Heller, P. Jones: The Construction of Bisector Skeletons for Polygonal Networks. In: Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984. Geographisches Institut der Universität Zürich-Irchel.

- Brauer, W., W. Wahlster (Hrsg.): Wissensbasierte Systeme. Informatik-Fachberichte 155, Springer, Berlin Heidelberg New York Tokyo 1987.
- Brüggemann, H.: Geometrische Bedingungen bei der Generalisierung. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 85/1981.
- Charlwood, G., G. Moon, J. Tulip: Developing a DBMS for Geographic Information: A Review. Wild Heerbrugg, Schweiz, 1987.
- Cohen, P.R.: The Control of Reasoning under Uncertainty; a Discussion of some Programs. The Knowledge Engineering Review, Cambridge University Press, vol. 2, no. 1/1987.
- Conzett, R.: Landinformationssysteme. Vermessung, Photogrammetrie, Kulturtechnik, Heft 5/1983.
- Damerow, P., R. Englund, H. Nissen: Die ersten Zahldarstellungen und die Entwicklung des Zahlbegriffs. Spektrum der Wissenschaft, Heft 3/1988.
- Dangermond, J.: The Software Toolbox Approach to meeting the User's Needs for GIS Analysis. Proceedings of Geographic Information Systems Workshop, pp.66 ff, Atlanta/GA, 1986.
- Ebner, H.: Zwei neue Interpolationsverfahren und Beispiele für ihre Anwendung. Bildmessung und Luftbildwesen, Nr. 47/1979.
- Eidg. Justiz- und Polizeidepartement (Hrsg.): Detailkonzept Reform der amtlichen Vermessung RAV. Organisatorische und technische Maßnahmen zur Verbesserung der Bodeninformation. Eidg. Drucksachen- und Materialzentrale, Bern 1987.
- Fischer, A.: Thematische Kartographie im Umweltberichtswesen. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 97/1986.
- Frank, A.: Datenstrukturen für Landinformationssysteme - semantische, topologische und räumliche Beziehungen in Daten der Geo-Wissenschaften. Mitteilung Nr. 34 des Institutes für Geodäsie und Photogrammetrie an der ETH Zürich, 1983a.
- Frank, A.: Probleme der Realisierung von Landinformationssystemen. 2. Teil: Storage Methods for Space Related Data: The Field Tree. Institut für Geodäsie und Photogrammetrie, Bericht Nr. 71, ETH Zürich, 1983b.

- Freeman, H.: Autonap - an Expert System for Automatic Map Name Placement. In: Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984. Geographisches Institut der Universität Zürich-Irchel.
- Giebels, M.: Automatische Symbolerzeugung für topographische Karten durch digitale Rasterdatenverarbeitung am Beispiel der Topographischen Übersichtskarte 1:200 000. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 92/1983.
- Gleixner, G.: Raumbezogene Informationssysteme im praktischen Einsatz. Eine Fallstudie am Beispiel der landwirtschaftlichen Bodenkartierung in Österreich. Diplomarbeit, TU Graz 1987.
- Göpfert, W.: Raumbezogene Informationssysteme: Datenerfassung - Verarbeitung - Integration - Ausgabe auf der Grundlage digitaler Bild- und Kartenverarbeitung. Wichmann, Karlsruhe 1987.
- Green, N.: Teach yourself geographical information systems. The design, creation and use of demonstrators and tutors. Int. J. Geographical Information Systems, vol.1, no.3/1987.
- Grill, E.: Relationale Datenbanken. Ziele, Methoden, Lösungen. CW-Publikationen, München, 1982.
- Grünreich, D.: Ein Verfahren zur automatischen Generalisierung flächenhafter Diskreta. Zeitschrift für Vermessungswesen, Heft 4/1986.
- Haag, K., G. Köhler: Realisierung geometrischer Bedingungen bei der Digitalisierung von Katasterkarten. Allgemeine Vermessungsnachrichten, Heft 5/1985.
- Habel, C. (Hrsg.): Künstliche Intelligenz; Repräsentation von Wissen und natürlichsprachliche Systeme. Informatik-Fachberichte 93, Springer, Berlin Heidelberg New York Tokyo 1984.
- Harbeck, R.: Das AdV - Vorhaben "Amtliches Topographisch-Kartographisches Informationssystem (ATKIS)" - Inhaltliche Konzeption. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 99/1987.
- Hase, K., P. Vonderhorst, C. Hidber, M. Tchang: Informationsraster des ORL, ORL Disp. Nr. 24/1972.
- Haywood, P.: The Differing Requirements of Producers and Users of Data and the Design of Topographic / Cartographic Databases. Nachrichten aus dem Karten- und Vermessungswesen, Reihe II, Heft 44/1986.

- Hentschel, W.: Digitale Karten für einen Verkehrslotsen. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 99/1987.
- Hinrichs, K., J. Nievergelt: The GRID FILE: A Data Structure Designed To Support Proximity Queries On Spatial Objects. Institut für Informatik, ETH Zürich, Juli 1983.
- Höflinger, E.: Über die Möglichkeit der Einrichtung eines Mehrzweckkatasters. Österreichische Zeitschrift für Vermessungswesen und Photogrammetrie, Heft 3/1980.
- Hönig, H.: Sichtung und Bewertung der wichtigsten vorliegenden Naturraumpotential - Modellentwicklungen im In- und Ausland. Bericht des Instituts für Umweltforschung, Graz 1984.
- Howe, D.R.: Data Analysis for Data Base Design. Arnold, London 1983.
- Istituto Geografico Militare: Dall'Italia Immaginata all'Immagine dell'Italia. Dokument einer Ausstellung. Florenz 1986.
- Joham, A.: Bilden von flächenhaften Objekten. Diplomarbeit, TU Graz 1988.
- Kalcher, A.: Untersuchung von Datenstrukturen zur Abspeicherung raumbezogener Daten. Diplomarbeit, TU Graz 1985.
- Kazmierczak, H. (Hrsg.): Erfassung und maschinelle Bearbeitung von Bilddaten. Springer, Berlin Heidelberg New York 1980.
- Klauer, R.H.: Automatisierte Digitalisierung und Strukturierung von Strichvorlagen. Zeitschrift für Vermessungswesen, Heft 4/1986.
- Knuth, D.E.: The Art of Computer Programming. Addison-Wesley, London 1972.
- Konecny, G.: Digital Terrain Models and Orthophotography. An Overview. Paper presented at the Symposium on Digital Elevation Models and Orthophoto Technology, Dept. of Surveying, Queensland, Australia, 1981.
- Kraus, K., E.M. Mikhail: Linear Least-Squares Interpolation. Paper presented at 12th Congress of the International Society of Photogrammetry, Ottawa, Canada, 1972.

- Kropatsch, W.: Eine kartographische Datenbank für Experimente zur kartengestützten Analyse digitaler Bilder. Folge 33 der Mitteilungen der geodätischen Institute der TU Graz, 1980, S. 133-176.
- Kropatsch, W.: Dreiecksapproximationen für Flächen. DIBAG Bericht Nr. 8, Institut für digitale Bildverarbeitung und Graphik, Graz 1983.
- Kuhn, W.: Zur Entwicklung interaktiver Programme und Systeme. Vermessung, Photogrammetrie, Kulturtechnik, Heft 2/1985.
- Kuhn, W.: Interaktion: Dialog mit modernen graphischen und numerischen 'Werkzeugen'. Informationstagung an der ETH Zürich, 23.-24.Oktober 1987.
- Lambird, B., D. Lavine: Distributed Architecture and Parallel Non-directional Search for Knowledge-Based Cartographic Feature Extraction Systems. In: M. J. Coombs (ed.): Developments in expert systems. Academic Press, London New York 1984.
- Lay, G., W. Weber: Waldgeneralisierung durch digitale Rasterdatenverarbeitung. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 92/1983.
- Lorber, G., G. Strohmaier: Digitaler Stadtplan im Rahmen eines kommunalen ortsbezogenen Grafik - Informationssysteme. Stadtvermessungsamt Graz, 1986.
- Mandelbrot, B.: The Fractal Geometry of Nature. Freeman and Company, New York 1983.
- McMaster, R.B.: Automated Line Generalization. Cartographica, vol. 24, no. 2/1987.
- Meier, A., K. Durrer, G. Heiser, E. Petry, A. Wälchli, C.A. Zehnder: XRS: Ein erweitertes relationales Datenbanksystem zur Verwaltung von technischen Objekten und Versionen. Institut für Informatik, ETH Zürich 1987.
- Meissl, P.: Least squares adjustment - a modern approach. Folge 43 der Mitteilungen der geodätischen Institute der TU Graz, 1982.
- Messmer, W.: Leitungskataster in der Entwicklung zum Teil eines EDV - Landinformationssystems Basel - Stadt. Vermessung, Photogrammetrie, Kulturtechnik, Heft 3/1983.
- Mittelstraß, G.: Stand der ALK-Entwicklung (Oktober 1986). Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Heft 99/1987.

- Monmonier, M.S.: Computer-Assisted Cartography. Principles and Prospects. Prentice Hall, Englewood Cliffs 1982.
- Muller, J.C.: The Concept of Error in Cartography. Cartographica, vol. 24, no. 2/1987.
- Müller, G.: Informationsstrukturierung in Datenbanksystemen. Oldenbourg, München 1978.
- Niese, D., W. Weber: Vektorisierte Rasteralgorithmen zur Verarbeitung von Höhenschichten. Nachrichten aus dem Karten- und Vermessungswesen. Reihe I, Nr. 99/1987.
- Nilsson, N.J.: Principles of Artificial Intelligence. Springer, Berlin Heidelberg New York 1982.
- Nittinger, J.: Informationssysteme und Vermessungswesen im In- und Ausland - eine Chance für die Zukunft. Allgemeine Vermessungsnachrichten, Heft 1/1985.
- Olle, T.W.: Das Codasyl - Datenbankmodell. Springer, Berlin Heidelberg New York 1981.
- Palmer, B.: Symbolic Feature Analysis and Expert Systems. In: Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984. Geographisches Institut der Universität Zürich-Irchel.
- Papoulis, A.: Probability, Random Variables and Stochastic Processes. McGraw-Hill, London 1965.
- Partridge, D.: Human Decision Making & the Symbolic Search Space Paradigm in AI. AI & Society, vol. 1/1987, pp.93-101.
- Pavlidis, T.: Graphics and Image Processing. Springer, Berlin Heidelberg New York 1982.
- Petry, E.: Konzept und Realisierung einer Versionenverwaltung in einem relationalen Datenbankkern. Institut für Informatik, ETH Zürich 1987.
- Peuquet, D.J.: Data Structures for a Knowledge - Based Geographic Information System. In: Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984. Geographisches Institut der Universität Zürich-Irchel.
- Pfleger, J.: Erster Entwurf eines Informationssystems als Hilfe zur Strukturplanung im ländlichen Raum. Österreichische Zeitschrift für Vermessungswesen und Photogrammetrie, Heft 1/1982.

- Ponschab, A.: Graphische Datenverarbeitung in einem kommunalen Versorgungsunternehmen. In M. Schilcher (Hrsg.): CAD-Kartographie. Anwendungen in der Praxis. Wichmann, Karlsruhe 1985.
- Preparata, F., M. Shamos: Computational Geometry. Springer, Berlin Heidelberg New York 1985.
- Rätzsch, H.: Entwicklung eines kompakten Systems zur Digitalisierung komplexer graphischer Vorlagen. DIBAG-Bericht Nr. 20-II, Institut für digitale Bildverarbeitung und Graphik, Graz 1984.
- Ranzinger, M., W. Kainz, R. Hütter: DESBOD - a geographical information system. Institut für digitale Bildverarbeitung und Graphik, Graz 1984.
- Reiner, B.: Vegetationserfassung mit Hilfe von Infrarot-Luftbildern. Eine Pilotstudie für Graz. Diplomarbeit, TU Graz 1988.
- Richardson, L.F.: The problem of contiguity: an appendix of statistics of deadly quarrels. General Systems Yearbook 6/1961, pp.139-187.
- Rosenfeld, A., A.C. Kak: Digital Picture Processing. Academic Press, London New York 1976.
- Schweinfurth, G.: Höhenliniengeneralisierung mit Methoden der digitalen Bildverarbeitung. DGK München, Reihe C, Nr. 291/1984.
- Schmidt-Falkenberg, H.: Begriff, Einteilung und Stellung der Kartographie in heutiger Sicht. Kartographische Nachrichten 1964, S.52-63.
- Schnupp, P., U. Leibrandt: Expertensysteme. Nicht nur für Informatiker. Springer, Berlin Heidelberg New York 1986.
- Sinding-Larsen, H.: Information Technology and the Management of Knowledge. AI & Society, vol. 1/1987.
- Slocum, T., R. Hanisch, D. Thompson: The Design and Applications of a Cartographic Data Base for Redistricting. Geo - Processing, no. 2/1984, pp.151-176, Elsevier, Amsterdam 1984.
- Smith, T.: Requirements and principles for the implementation and construction of large-scale geographic information systems. Int. J. Geographical Information Systems, vol.1, no.1/1987.

- Smith, T., M. Pazner: Knowledge-Based Control of Search and Learning in a Large-Scale GIS. In: Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984. Geographisches Institut der Universität Zürich-Irchel.
- Späni B., N. Bartelme: Vom Datenmodell zur Anwenderschale eines raumbezogenen Informationssystems. Vortrag anlässlich des X. Internationalen Kurses für Ingenieurvermessung, München, 12.-17. September 1988.
- Sünkel, H.: GSPP - A General Surface Representation Module Designed for Geodesy. Bull. Geod. 55/1981, pp. 31-40.
- Tamminen, M.: The EXCELL method for efficient geometric access to data. Acta Polytechnica Scandinavica, Mathematics and Computer Science Series No. 34, Helsinki 1981.
- Tomlinson, R.: Current and potential uses of geographical information systems. The North American experience. Int. J. Geographical Information Systems, vol.1, no.3/1987.
- Trapp, R. (ed.): Impacts of Artificial Intelligence. Elsevier, Amsterdam 1986.
- Trost, H., E. Buchberger, W. Heinz, C. Hörtnagel, J. Matiassek: Datenbank-Dialog. A German Language Interface for Relational Databases. Applied Artificial Intelligence, vol. 1, no. 2/1987.
- Uhlenbruck, M., J.D. McLaughlin: Expert System Applications for Land Information Management. Proceedings of the Annual Fall Convention of the ACSM, Reno, Nev., 1987.
- Wagner, K.: Graphentheorie. BI, Mannheim 1970.
- Weber, W.: Automationsgestützte Generalisierung. Nachrichten aus dem Karten- und Vermessungswesen. Reihe I, Nr. 88/1982.
- Weber, W.: Raster - Datenverarbeitung in der Kartographie. Nachrichten aus dem Karten- und Vermessungswesen, Reihe I, Nr. 88/1982.
- Wiggins, J.C., R.P. Hartley, M.J. Higgins, R.J. Whittaker: Computing aspects of a large geographic information system for the European Community. Int. J. Geographical Information Systems, vol.1, no.1/1987.
- Willmersdorf, E.: Einsatz der graphischen Datenverarbeitung für geographische Informationssysteme. Vortrag an der TU Graz, Jänner 1986.
- Witt, W.: Thematische Kartographie. Jänecke, Hannover 1970.



ANHANG B
GLOSSAR

Einleitung

Die stürmische Entwicklung in jungen Fachgebieten - und dazu zählen das Geoinformationswesen und die digitale Kartographie - bringt es mit sich, daß eine Fülle von neuen Ausdrücken und von neuen, ungewohnten Inhalten für bekannte Bezeichnungen entsteht. Dabei ergibt es sich zwangsläufig, daß man in bezug auf eine Konsolidierung und Vereinheitlichung der Terminologie im Rückstand ist. Es gibt also durchaus unterschiedliche Auffassungen zur Bedeutung einzelner Begriffe. Umso wichtiger erscheint daher eine Standortbestimmung der Bezeichnungen, die in diesem Buch verwendet werden. Die nachfolgende Liste ist also nicht als Versuch zu sehen, die (persönlich gefärbte) Sicht des Autors als Dogma zu deklarieren; vielmehr soll sie dem Leser eine Hilfe dabei sein, die hier verwendete Terminologie in seine eigene Begriffswelt abzubilden.

Liste der Begriffe

Absolute Geometrie: Die Koordinaten topologischer Elemente, die sich nach Berücksichtigung unterschiedlicher Erfassungsergebnisse (der *gemessenen Geometrie*) ergeben.

Abstandstransformierte: Sekundärbild zu einer Rasterstruktur, in dem die Grauwerte eine *metrische* Bedeutung haben; hohe Grauwerte im Sekundärbild deuten auf große Abstände im Primärbild hin.

Adreßraum: Bezeichnung für ein Suchgitter, das für raumbezogen gespeicherte Daten (*EXCELL* oder *GRID FILE*) aufgebaut wird.

Aggregieren: Ermitteln einer Gesamtfläche aus vorgegebenen Teilflächen bzw. Ermitteln einer Linie aus Linienstücken (*Spaghetti*).

AI (Artificial Intelligence): Synonym für *Künstliche Intelligenz*.

Anwenderprofil: Jene Teile der *Benützerschnittstelle*, die den Anforderungen eines bestimmten Anwenders angepaßt werden können; die anderen Anwender bleiben davon unberührt.

Approximation: Ermittlung einer Kurve oder Fläche, die möglichst nahe an vorgegebene Punkte oder Flächenkurven herankommt und zusätzliche Nebenbedingungen (wie etwa einen möglichst glatten Verlauf) erfüllt.

Areales Modell: Synonym für *Rastermodell*.

Attribut: Beschreibendes Merkmal einer *Datengruppe*, das jedoch selbst im allgemeinen kein eindeutiges Zugriffskriterium (*Schlüssel*) darstellt.

Äußerer Ring: Folge von *Kanten*, die eine äußere Begrenzung einer Fläche darstellt.

Backward chaining: Im Bereich der *Künstlichen Intelligenz* versteht man darunter die deduktive Strategie, die durch Zerlegung von Problemen in Teilprobleme zum Ziel kommt ("goal-driven strategy", im Gegensatz zum *forward chaining* bzw. "data-driven strategy").

Baumstruktur: Spezielle Struktur eines Graphen; muß *zusammenhängend* sein und darf keine *Zyklen* aufweisen.

Bearbeitungsauszug: Der Anwender definiert einen räumlichen und / oder thematischen Ausschnitt des gesamten Datenbestandes als eine logische Einheit, die er mit einem Namen versehen kann und unter diesem Namen jederzeit abrufen kann (*View*).

Bemaßung: Aspekt der Karten- und Planausgestaltung: Metrische Informationen (Entfernungen, Flächenmaße usw.) werden in die Karte / den Plan eingeblendet.

Benützerschnittstelle: Gesamtheit der Möglichkeiten, die einem Anwender im Rahmen der interaktiven Ausnutzung der Software und der Daten eines EDV-Systems zur Verfügung stehen (*user interface*).

Beziehung vom Typ 1:1: Beziehung zwischen *Datengruppen*, die in beiden Richtungen eindeutig ist ("one to one").

Beziehung vom Typ 1:M: Beziehung zwischen *Datengruppen*, die einer *Realisierung* der ersten *Datengruppe* keine, eine oder mehrere *Realisierungen* der zweiten *Datengruppe* zuordnet. ("one to many").

Beziehung vom Typ M:N: Beziehung zwischen *Datengruppen*, die in keiner Richtung eindeutig ist ("many to many").

Bezierkurve: Findet im Rahmen der *Interpolation* und *Approximation* Verwendung; baut auf Polynomen auf, die eine Reihe von Bedingungen hinsichtlich eines möglichst glatten Verlaufes erfüllen.

Blattschnittfreie Speicherung: Forderung nach einer Speicheremethode, die sich nicht an rechteckigen Ausschnitten orientiert.

Block: Mehrere Bausteine eines *Vektormodells*, die vom Anwender aus Gründen einer besseren Manipulierbarkeit zu einer Einheit zusammengefaßt werden.

Blow-Shrink: Synonym für *Verdicken-Verdünnen*.

Bulk: Unstrukturierter Datensatz, der von der *Datenbank* als Ganzes verwaltet wird und nur von Anwenderprogrammen näher aufgeschlüsselt werden kann.

Chain code: Form der Datenkompression in *Rastermodellen*. Eine Linie (eine Kette von aneinanderstoßenden Rasterzellen) wird durch die Angabe einer Anfangszelle sowie durch eine Reihe von aufeinander folgenden Fortschreitungsrichtungen definiert.

Datenbank: Ein auf Dauer angelegter Datenbestand, gekoppelt mit einem Datenverwaltungssystem, das ihn schützt und ihn verschiedenen Benutzern zugänglich macht.

Datenbankmodell: Je nach der Art der Verkettung der Daten einer *Datenbank* unterscheidet man das *hierarchische Modell*, das *Netzwerkmodell* und das *relationale Modell*.

Datenbankschema: Eine *Datenbank* stellt sich für verschiedene Zielgruppen unterschiedlich dar; so unterscheidet man zwischen dem *internen*, dem *konzeptionellen* und dem *externen Schema* einer Datenbank.

Datengruppe: Zusammenfassung gleichartiger Datenelemente zu einer Einheit. Es gibt *Realisierungen* (aktuelle Werte) einer Datengruppe; Beispiel: "Donau" ist eine Realisierung der Datengruppe "Fluß".

DGM: Synonym für digitales *Geländemodell*.

DHM: Synonym für digitales *Höhenmodell*.

Dialogkomponente: Teil eines *Expertensystems*, der als Schnittstelle zum Anwender fungiert.

Digitale Karte: Synonym für *GIS*.

Digitaler Plan: Synonym für *LIS*.

DIME-Struktur (Dual Independent Map Encoding): Ausprägung eines *Vektormodells*, das *0-Zellen* (nulldimensionale Elemente bzw. *Knoten*), *1-Zellen* (eindimensionale Elemente bzw. *Kanten*) und *2-Zellen* (zweidimensionale Elemente bzw. *Flächen*) beinhaltet, wobei alle Kanten in *Zyklen* eingebunden sind. Die Flächen einer DIME-Struktur sind *Knoten* eines *dualen Graphen*.

Directory: Synonym für *Adreßraum*.

Domäne: Spalte einer Tabelle eines *relationalen Datenbankmodells*.

Drahtmodell: Dreidimensionales *Modell*, das auf Punkten und linienförmigen Verbindungen beruht.

Dualer Graph: Im Rahmen einer *DIME-Struktur* wird zu einer Kanten-Knoten-Struktur ein solcher Graph aufgebaut (*Thiessen-Polygon, Voronoi-Graph, RAG*).

Ergonomie: Im allgemeinen versteht man darunter eine auf die Bedürfnisse des Menschen optimal ausgerichtete Anordnung der Hardware. Hier wird dieser Begriff speziell auch auf die Software ausgedehnt.

Erklärungskomponente: Teil eines *Expertensystems*, der das Zustandekommen komplexer Schlußfolgerungen erklärt ("why-how-utility").

EXCELL: Methode zur räumlichen Speicherung; baut für den Zugriff einen regelmäßig strukturierten Adreßraum auf, der durch eine Matrix von Page-Adressen dargestellt werden kann.

Expertensystem: Wird der *Künstlichen Intelligenz* zugeordnet; für ein eng abgegrenztes Wissensgebiet wird hochspezialisiertes Wissen in einer *Wissensbank* abgelegt und kann zur Entscheidung herangezogen werden.

Externes Schema: *Datenbankschema*, das einen Auszug des *konzeptionellen Schemas* darstellt und auf einen bestimmten Benutzer bzw. eine konkrete Anwendung zugeschnitten ist.

Filterung: Abschwächung oder Unterdrückung bestimmter *Frequenzbereiche* einer durch periodische Schwingungen gekennzeichneten Erscheinung. In unserem Fall sind diese Schwingungen als Änderungen der Thematik in Abhängigkeit vom Ort aufzufassen. Man unterscheidet *Tiefpaß-* und *Hochpaßfilterung*.

Flächenbildung: Automatisches Zusammenfügen von Flächen aus Linienstücken (*Spaghetti*), indem *äußere* und *innere* Ränder bestimmt werden und *thematische* Kriterien berücksichtigt werden.

Folienprinzip: *Thematische Ebenen* können als beliebig miteinander kombinierbare Folien angesehen werden.

Formparameter: Gibt den geometrischen Aspekt einer Kante an (geradlinige Verbindung, bogenförmige Verbindung, usw.).

Forward chaining: Im Bereich der *Künstlichen Intelligenz* versteht man darunter die induktive Wissensbereitstellung, die von gegebenen Fakten ausgeht und sukzessive immer mehr Wissen akkumuliert ("data-driven strategy", im Gegensatz zum *backward chaining* bzw. "goal-driven strategy").

Fourier-Transformation: Allgemein ein Hilfsmittel zur Transformation vom *Ortsbereich* in den *Frequenzbereich*, wobei der Grund für einen solchen Übergang in der leichteren Manipulierbarkeit liegt. Hier ist im speziellen damit eine *Filterung* verbunden, die im Frequenzbereich einfacher zu bewerkstelligen ist.

Fraktale Dimension: Maß für die Komplexität eines *fraktalen Modells*.

Fraktales Modell: Form eines Datenmodells, das auf der *Selbst-Ähnlichkeit* beruht; globale geometrisch-topologische Eigenschaften werden im Kleinen reproduziert. Ein solches Modell unterdrückt daher nicht - wie die herkömmlichen Modelle - die Einzelheiten.

Frame: Synonym für *Rahmen*.

Freistellungsfläche: Spezielle Ausprägung einer *Pufferzone* für ein kartographisches Symbol oder eine Beschriftung, in der die Hintergrundinformation unterdrückt wird.

Fremdschlüssel: Teil einer *Datengruppe*, der als *Primärschlüssel* einer anderen Datengruppe auftritt.

Frequenzbereich: Bezeichnung im Rahmen der *Fourier-Transformation* und der damit verbundenen *Filterung*: es ist dies der Bereich der transformierten *Grauwerte*, in dem die Filterung durchgeführt wird.

Geländemodell: *Modell* eines Teiles der Erdoberfläche, das im allgemeinen von gegebenen Raumpunkten - manchmal auch von Raumkurven oder kleinen Flächenstücken im Raum - ausgeht und darauf eine Fläche aufbaut, die bestimmten vorgegebenen Kriterien genügt (*DGM*, *Höhenmodell* oder *DHM*).

Gemessene Geometrie: Die Koordinaten topologischer Elemente, die sich aus *einem* Datenerfassungsvorgang ergeben (im Gegensatz zur *absoluten Geometrie*).

Generalisierung: Unterdrückung "unwesentlicher" Einzelheiten im Rahmen der *Modellbildung*. Oft versteht man darunter speziell eine kartographische Maßnahme, die beim Übergang von größeren zu kleineren Maßstäben durch eine gezielte Verringerung des Informationsgehaltes die Lesbarkeit sicherstellt.

Generator: Operator, der auf den *Initiator* eines *fraktalen Modells* und auf die Folgezustände angewendet wird.

Geo-Informationssystem (GIS): Ausprägung eines *RIS* mit Schwerpunkt auf kleinmaßstäbliche Karten. Oft (vor allem im englischsprachigen Raum) als Synonym für ein *RIS* verwendet.

GIS: Abkürzung für *Geo-Informationssystem*.

Gleitendes Mittel: Im Rahmen von *Geländemodellen* bezeichnet man damit eine Methode, welche die Höhe an einem vorgegebenen Punkt durch eine gewichtete Mittelbildung der Höhen in einer bestimmten Nachbarschaft schätzt.

Graph: Menge von *Knoten* und *Kanten*; *Knoten* werden durch *Kanten* verbunden, und jede *Kante* hat genau einen Anfangsknoten und genau einen Endknoten.

Graphisch-interaktiver Arbeitsplatz (GIAP): Bezeichnung für eine Hardware- und Software-Umgebung, welche die Eingabe, interaktive Bearbeitung und Ausgabe von Daten eines *RIS* unterstützt.

Grauwert: Im engeren Sinn der Bildinhalt einer *Rasterzelle*; im weiteren Sinn die einer *Rasterzelle* überlagerte *Thematik*.

GRID FILE: Methode zur räumlichen Speicherung; baut für den Zugriff einen regelmäßig strukturierten Adreßraum auf, der durch eine Matrix von *Page-Adressen* dargestellt werden kann. Daneben werden *Vektoren* aufgebaut, welche die Ausdehnung der *Pages* in den jeweiligen Koordinatenrichtungen enthalten.

Hierarchie: Beziehung, in der jedes Element genau einen Vorgänger hat, während die Anzahl der Nachfolger beliebig ist (fallweise fehlt auch ein Nachfolger).

Hierarchisches Datenbankmodell: Form eines *Datenbankmodells*, in dem die *Datengruppen* in einer streng *hierarchischen* Beziehung zueinander stehen.

Hochpaßfilterung: Form der *Filterung*, welche die tiefen *Frequenzen* unterdrückt. Es sind dies jene Bereiche, wo sich die *Thematik* nicht oder nur in geringem Maß ändert (etwa im Inneren einer Fläche).

Höhenmodell: Synonym für *Geländemodell*.

Höhenschichtenmodell: Spielart eines *Geländemodells*, die sich auf Schichten konstanter Dicke stützt, welche sich nach oben hin verjüngen.

Homogene Koordinaten: Darstellungsform für Punkte und Gerade, die eine einfache Manipulierbarkeit gewährleistet.

Hybrides Modell: Mischform zwischen einem *Rastermodell*, das im Großen Anwendung findet, und einem *Vektormodell* im Detail.

Informationssystem: Dient der Erfassung, Speicherung, Verarbeitung und Auswertung von Daten, die einen bestimmten Ausschnitt der Realität beschreiben und zu einem organisatorischen Ganzen zusammengefaßt sind. Es besteht aus einer *Datenbank* und aus einer Reihe von Werkzeugen zur Bearbeitung der Daten.

Index: Eine vom Datenbankverwaltungssystem bereitgestellte Beschleunigung des Zugriffs auf bestimmte Teile einer Daten-*gruppe*.

Initiator: Anfangszustand einer *fraktalen Modellbildung*.

Innerer Ring: Folge von *Kanten*, welche eine Fläche von einer in ihr enthaltenen *Insel* trennt.

Insel: Bezeichnung für eine Fläche, die ganz in einer anderen Fläche enthalten ist, ohne daß sich die beiden Flächenbegrenzungen schneiden.

Instance (Instanz): Synonym für *Realisierung einer Daten-*gruppe**.

Internes Schema: *Datenbankschema*, das die interne Gruppierung der Daten und die Speicherplatzbelegung beschreibt.

Interpolation: Ermittlung einer Kurve oder Fläche, die durch vorgegebene Punkte oder Flächenkurven hindurchgeht und zusätzliche Nebenbedingungen (wie etwa einen möglichst glatten Verlauf) erfüllt.

Isomorphie von Graphen: Zwei *Graphen* sind isomorph, wenn zwischen ihnen eine umkehrbar eindeutige Abbildung existiert, welche die topologischen Beziehungen beibehält.

Journalfile: Protokoll aller Anwenderaktionen. Ähnlich dem *Logfile*, jedoch nicht speziell auf Datenbankzugriffe ausgerichtet.

Kante: Graphentheoretische Bezeichnung für die linienförmigen Elemente eines *Vektormodells*.

Kanten-Knoten-Struktur: Ausprägung eines *Vektormodells*. Punkte und Linien werden durch *Knoten* bzw. *Kanten* modelliert. Flächen werden durch die sie umgebenden *Kantenfolgen* (fallweise auch durch *innere Grenzkanten*) repräsentiert.

KD-Baum: Entspricht einer rekursiven Aufteilung eines Bereiches in jeweils zwei Teilbereiche, wobei die einzelnen Koordinatenrichtungen alternierend zum Zuge kommen.

Key: Synonym für *Schlüssel*.

KI: Synonym für *Künstliche Intelligenz*.

Knoten: Graphentheoretische Bezeichnung für die punktförmigen Elemente eines *Vektormodells*.

Kollokation: Allgemeiner *Interpolationsansatz*, der systematische Anteile (*Trend*) und zufällige Anteile (*Rauschen*) gesondert berücksichtigt.

Konsistenz: Der Datenbestand eines Informationssystems ist frei von Widersprüchen.

Konvexes Polygon: Die Verbindungslinie zweier beliebiger im Inneren des Polygons befindlicher Punkte liegt ganz innerhalb des Polygons; es gibt also keine "einspringenden Ecken".

Konvolution: Form der *Filterung*, welche jeden Grauwert durch ein gewogenes Mittel seiner Umgebung ersetzt.

Konzeptionelles Schema: *Datenbankschema*, das die logischen Zusammenhänge der Daten - unabhängig von einer konkreten Anwendung - beschreibt.

Kovarianzfunktion: Im Rahmen von *Geländemodellen* bezeichnet man damit eine Funktion, welche den Einfluß von vorgegebenen Höhen auf ihre Nachbarschaft in Abhängigkeit von der Entfernung darstellt.

Künstliche Intelligenz: Sammelbegriff für alle Bestrebungen, die ein besseres Verständnis der Funktionsweise intelligenten Verhaltens zum Ziel haben bzw. einer maschinengestützten Simulation bestimmter Teilbereiche der menschlichen Entscheidungsfindung und der Lernmechanismen gelten.

Landinformationssystem (LIS): Ausprägung eines *RIS* mit Schwerpunkt auf großmaßstäbliche Pläne (Präzisionszeichnungen).

Legende: Beschriftung und Erklärung einer Karte bzw. eines Plans und der dabei verwendeten Symbolik.

Leitpunkt: Im Rahmen der *Interpolation* und *Approximation* bezeichnet man damit einen Punkt, in dessen Nähe eine Kurve oder Fläche verlaufen soll.

Lernmodul: Teil eines *Expertensystems*, der die Bereitstellung von Fakten und Regeln unterstützt und der auch abgeleitete Regeln aufnehmen kann.

Lineales Modell: Synonym für *Vektormodell*.

LIS: Abkürzung für *Landinformationssystem*.

Logfile: Protokoll, das von einem Datenbankverwaltungssystem mitgeführt wird, und das es erlaubt, Operationen (etwa nach einem Systemabsturz) zu wiederholen (*Roll forward*) oder rückgängig zu machen (*Roll back*).

Look-up table: Übersetzungstabelle, etwa zur Umsetzung einer langfristig bedeutsamen *Thematik* in eine aktuelle graphische Ausprägung.

Makrofunktionen: Elementare Funktionen aus dem Software-Instrumentarium eines *Informationssystems* können im Sinne der *Makrotechnik* zu komplexen Operationen zusammengefaßt werden.

Makrotechnik: Baukastenprinzip; aus mehreren elementaren Bestandteilen baut der Anwender komplexe *Blöcke* zusammen; dies kann sich auf Daten, aber auch auf *Software-Werkzeuge* beziehen (*Makrofunktionen*).

Menu: Auswahl von Möglichkeiten für die Weiterführung des Programmes, die dem Anwender zur interaktiven Entscheidung angeboten wird.

Metrik: Definition eines Abstandsbegriffes, der bestimmten Regeln unterworfen sein muß.

Minimumsprinzip: Im Rahmen der *Approximation* wird damit die Forderung nach möglichst kleinen Klaffungen zwischen vorgegebenen Punkten und einer approximierenden Kurve oder Fläche bezeichnet.

Modell: Abstraktion eines Teiles der Realität; Ergebnis einer Abbildung, welche die für die geplanten Auswertungen relevanten Teile der Realität (mehr oder minder genau) übernimmt und den Rest unterdrückt.

Multi-User-Betrieb: Bezeichnung für die Möglichkeit, daß mehrere Benützer gleichzeitig auf die Ressourcen eines Informationssystems zugreifen können, ohne einander zu behindern.

Navigieren: Das Beschreiten eines Zugriffspfades im Rahmen der Identifikation einer *Realisierung* einer *Datengruppe*.

Netz: Allgemeine Form eines *Vektormodells*; *Bäume* und *Zyklen* sind spezielle Ausprägungen eines Netzes.

Netzwerkmodell: Form eines *Datenbankmodells*, in dem die *Datengruppen* in ein Netz von Beziehungen eingebettet sind, die durch Zeiger (*Pointer*) realisiert werden ("*CODASYL-Standard*").

Nicht-monotone Logik: Ausdruck der *Künstlichen Intelligenz* für Strategien, die im Fall von Regeln angewendet werden, die widersprüchlich sein können.

Normalform: Forderung nach einer möglichst einfachen, minimalen und redundanzfreien Speicherungsstrategie, die eine Garantie für die langfristige *Stabilität* einer Datenstruktur bietet. Es gibt Steigerungsstufen von der 1. bis zur 3. Normalform.

Objekt: Zusammenfassung von Elementen zu einer Einheit. Thematische Überlegungen geben dazu den Anstoß; das Ziel ist eine leichtere Manipulierbarkeit des Objektes in seiner Gesamtheit.

Ortsbereich: Bezeichnung im Rahmen der *Fourier-Transformation* und der damit verbundenen *Filterung*: es ist dies der Bereich der Grauwerte, von denen man ausgeht.

Page: Massenspeicherbereich mit physisch benachbart gespeicherten Daten, der in seiner Gesamtheit geladen bzw. archiviert wird.

Paßpunkttransformation: Koordinatentransformation, die durch eine Reihe von Paßpunkten bestimmt wird, deren Koordinaten in beiden Systemen bekannt sind.

Planarer Graph: Ist *isomorph* zu einem Graphen, in dem sich keine Kanten schneiden.

Pop-up-Menu: Spezielle Form für ein *Menu*, das bei Bedarf am Bildschirm eingeblendet wird.

Prädiktion: Allgemeiner *Interpolationsansatz*, der es erlaubt, von vorgegebenen Werten auf neue, zu schätzende Werte zu schließen, wobei man voraussetzt, daß der Nachbarschaftseinfluß statistisch bekannt ist.

Primärdaten: Im Rahmen von *Geländemodellen* versteht man darunter die (im allgemeinen unregelmäßig vorgegebenen) Ausgangsdaten, aus denen man aufgrund einer *Interpolationsvorschrift* (regelmäßig angeordnete) *Sekundärdaten* erzeugt.

Primärschlüssel: Teil einer *Datengruppe*, der eine eindeutige Identifikation einer *Realisierung* dieser Gruppe ermöglicht.

Problemlösungskomponente: Teil eines *Expertensystems*, der aus Grundregeln ein komplexes Wissen ermitteln kann ("Ableitungsmechanismus", engl. "inference engine").

Projekt: Gesamtheit aller jener Daten eines Informationssystems, die ein Anwender während eines beschränkten Zeitraumes für einen bestimmten Zweck benötigt.

Projektprofil: Jene Teile eines *Anwenderprofiles*, die projektbezogen sind.

Pseudokante: Verbindet den *äußeren Ring* einer Fläche mit einem *inneren Ring*, bzw. zwei innere Ringe miteinander; sie dient lediglich dem topologischen Zusammenhalt der äußeren und inneren Ringe.

Pufferzone: Umgebung eines Punktes, einer Linie, oder einer anderen *topologischen Struktur*, die meist einer *Verschneidung* mit dem Hintergrund unterworfen wird.

Quad Tree: *Baumstruktur*, in der jeder Knoten vier Nachfolger hat. Oft als Synonym für eine rekursive Aufteilung eines rechteckigen Bereiches in vier Quadranten verwendet.

Qualitative Karte: *Thematische Karte*, welche die Art eines Themas in den Vordergrund stellt.

Quantitative Karte: *Thematische Karte*, welche die mit einem Thema verbundenen Werte, Größen und Mengen in den Vordergrund stellt.

Radiometrische Transformation: Umwandlung der Grauwerte einer Rasterstruktur nach einer vorgegebenen Formel bzw. Funktion (*Transferfunktion*).

RAG: Synonym für *Region adjacency graph*.

Rahmen: Im Rahmen der *KI* bezeichnet man damit eine (bekannte) Situation, die man mit einer (aktuellen) Situation vergleichen möchte. Diese aktuelle *Instanz* soll also (mehr oder minder gut) mit bekannten Rahmenbedingungen zur Deckung gebracht werden. Ein Rahmen besteht im allgemeinen aus mehreren *Slots* (Schlitzen). Oft wird auch die Gesamtheit der *Attribute*, die einer *Datengruppe* zugeordnet werden, als Rahmen bezeichnet.

Randausgleich: Anpassung zweier getrennt aufgenommener und sich teilweise überlappenden Datenbestände.

Rang-Deskriptor: Thematische Wertigkeit von Objekten, die es erlaubt, bei der *Generalisierung* wichtige von unwichtigen Aspekten zu trennen.

Rastermodell: *Modell*, das auf regelmäßig angeordneten Gitterzellen beruht.

Rasterung: Umwandlung eines *Vektormodells* in ein *Rastermodell*.

Rasterzelle: Elementarbaustein eines *Rastermodells*; kleinstes - nicht mehr unterteilbares - rechteckiges Gebiet mit homogener Bedeutung.

Raumbezogenes Informationssystem (RIS): *Informationssystem*, dessen Daten einen mehr oder minder starken Bezug zu Koordinaten aufweisen.

Rauschen: Der zufällige Anteil einer *Geländeinterpolation*, also jene Aspekte, die sich rasch ändern (im Gegensatz zum *Trend*).

Realisierung einer Datengruppe: "Momentaufnahme", die ein Element einer Datengruppe zeigt (*Instanzen, Instance*).

Regeln: Entscheidungskriterien, die in der Form "wenn A, dann B" vorliegen und in der *Wissensbank* eines *Expertensystems* gespeichert sind.

Region adjacency graph (RAG): Dualer Graph einer *DIME-Struktur*.

Relationale Algebra: Beschreibt die Verknüpfungen und Operationen, die auf Tabellen und Tabellenteile eines *relationalen Datenbankmodells* angewendet werden können.

Relationaler Kalkül: Mathematisch präzise Definition von Anfragen an eine *relationale Datenbank*.

Relationales Datenbankmodell: Form eines *Datenbankmodells*, in dem die *Datengruppen* in Tabellen angeordnet sind und bei dem die Verweise zwischen einzelnen Datengruppen nicht durch physische Adressen, sondern implizit über die gespeicherten Werte realisiert werden.

RIS: Abkürzung für *raumbezogenes Informationssystem*.

Roll Back: Operationen einer *Datenbank* können rückgängig gemacht werden.

Roll Forward: Nachvollziehen von Operationen einer *Datenbank* aufgrund eines *Logfiles*.

Run length encoding: Form der Datenkompression in *Rastermodellen*, welche den Umstand ausnützt, daß sich der Bildinhalt beim zeilenweisen Durchlauf der Rasterzellen nur selten (bezogen auf die Gesamtzahl der Zellen) ändert.

Schlüssel: Teil einer *Datengruppe*, der einen *eindeutigen Zugriff* auf diese Datengruppe gewährleistet (im Gegensatz zum *Attribut*).

Schwellwertbildung: Spezielle *Transferfunktion*, welche die Grauwerte unterhalb einer bestimmten Schranke unterdrückt.

Sekundärdaten: Im Rahmen von *Geländemodellen* versteht man darunter jene (regelmäßig angeordneten) Daten, die aufgrund einer *Interpolationsvorschrift* aus (unregelmäßigen) *Primärdaten* erzeugt werden.

Sekundärschlüssel: Teil einer *Datengruppe*, der für den Zugriff verwendet wird, aber im allgemeinen keine eindeutige Identifikation ermöglicht (im Gegensatz zum *Primärschlüssel*).

Selbst-Ähnlichkeit: Aspekt eines *fraktalen Modells*; globale Eigenschaften reproduzieren sich im Detail (engl. "self-similarity").

Selektion: Wichtigster Bestandteil einer relationalen Zugriffssprache. Aus einer Tabelle werden jene Teile in eine neue Tabelle übernommen, die einem bestimmten Kriterium genügen.

Semantisches Netz: Anspruchsvolle Form der Wissensrepräsentation, welche die einzelnen Begriffe als *Knoten* eines Netzes auffaßt, während die verschieden starken Assoziationen zwischen den Begriffen durch gewichtete *Kanten* repräsentiert werden.

Skelett: Vereinfachte Repräsentation einer Fläche; in *Vektormodellen* ist es ein Graph, der als Grenzfall von simultan von den *äußeren* und *inneren Ringen* aufeinander zugehenden Wellenfronten entsteht; in *Rastermodellen* ist es eine Anordnung von *Rasterzellen*, die im Innersten der Fläche liegen.

Slot (Schlitz): Teil eines *Rahmens*, etwa ein *Attribut* einer *Datengruppe*; beispielsweise ist das Attribut "Flußname" eines von vielen Attributen der Datengruppe "Fluß". Ein Slot kann für eine konkrete Realisierung der Datengruppe einen Wert (*Instanz*) annehmen, z.B. "Donau".

Soft-Menu: Spezielle Form für ein Menu, das der Anwender (mit Unterstützung des Systems) selbst erstellt und das am Bildschirm eingeblendet wird oder auf den Digitalisierertisch geklebt wird.

Software-Werkzeuge: Instrumentarium von Algorithmen, Methoden und Verfahren zur Bearbeitung von Daten eines *Informationssystem*s, die dem Anwender zur Verfügung gestellt werden.

Spaghetti: Linienstücke, die bei der (halb-)automatischen Erfassung anfallen und dann einem thematischen *Flächenbildungsprozeß* oder einer Vernetzung unterworfen werden.

Splinekurve: Findet im Rahmen der *Interpolation* und *Approximation* Verwendung; sie besteht aus stückweise definierten Polynomen, die eine Reihe von Bedingungen hinsichtlich eines möglichst glatten Verlaufes erfüllen.

Stabile Datenstruktur: Änderungen am Datenbestand sollen sich nur lokal auswirken und keine Inkonsistenzen erzeugen.

Stochastischer Prozeß: Ergebnis eines Experimentes, das vom Zufall gesteuert ist und außerdem zeitabhängig ist.

Surrogat: Ein vom Datenbankverwaltungssystem intern verbogener eindeutiger *Schlüssel*.

Symbolfreistellung: Aspekt der Karten- und Planausgestaltung: Punkt-, Linien- und Flächensymbole verdrängen oder ersetzen (aus optischen Gründen) den Hintergrund.

Teragon: Grenzfall eines Polygones, der sich durch fortgesetzte Rekursion im Rahmen einer *fraktalen Modellbildung* ergibt.

Textfreistellung: Aspekt der Karten- und Planausgestaltung: Punkt-, Linien- und Flächenbeschriftungen verdrängen oder ersetzen (aus optischen Gründen) den Hintergrund.

Tiefpaßfilterung: Form der *Filterung*, welche die hohen *Frequenzen* unterdrückt. Es sind dies jene Bereiche, wo sich die *Thematik* sehr stark ändert (etwa an Flächenrändern).

Thema, Thematik, thematische Karte: Auf der Grundlage von geodätischen Plänen oder geographischen Karten wird ein bestimmter Aspekt (eine abgegrenzte Gruppe von Aspekten) im Vergleich zu den anderen vorrangig behandelt und besonders hervorgehoben.

Thematische Ebene: Die Ausprägungen eines Themas sind für ein bestimmtes Gebiet definiert und bilden in ihrer Gesamtheit eine thematische Ebene. Das gesamte *RIS* besteht aus einem Stapel solcher Ebenen. Für einen konkreten Zweck können einzelne Ebenen aus diesem Stapel selektiert werden (*Folienprinzip*).

Thiessen-Polygon: Dualer Graph einer *DIME-Struktur*, der auch noch eine metrische Bedeutung hat: jede seiner Kanten ist äquidistant zu den beiden jeweils benachbarten *DIME-Knoten* (auch *Voronoi-Graph*).

Topologie: Sammelbegriff für jene Eigenschaften, die bei einer stetigen Abbildung nicht verloren gehen, wie etwa "Verbunden-Sein", "Berühren", "Benachbart-Sein", "Überdecken", "Beinhalten".

Transaktion: Zusammenfassung von einzelnen Operationen einer Datenbank. Eine Transaktion führt die Datenbank immer von einem konsistenten Zustand wieder in einen konsistenten Zustand über. Sie ist somit die kleinste Einheit für die Datensicherung, für ein Roll back oder ein Roll forward.

Transferfunktion: Vorschrift, die im Rahmen einer Radiometrischen Transformation den Grauwerten eines (Ur-)Bildes die Grauwerte eines anderen (Ergebnis-)Bildes zuordnet.

Trend: Der systematische Anteil einer Geländeinterpolation, also jene Aspekte, die sich nur langsam ändern (im Gegensatz zum Rauschen).

Tupel: Zeile einer Tabelle eines relationalen Datenbankmodells.

UND-ODER-Graph: Form der Wissensrepräsentation, welche sich eines Graphen bedient (im Gegensatz zu regelgestützten Expertensystemen).

User interface: Synonym für Benützerschnittstelle.

Vages Wissen: Ausdruck der Künstlichen Intelligenz für Fakten bzw. Regeln, die nicht immer exakt, verfügbar oder vollständig sind.

Vektorisierung: Umwandlung eines Rastermodells in ein Vektormodell.

Vektormodell: Modell, das auf Punkten und Linien beruht.

Verdicken und Verdünnen: Operation, die auf eine Rasterstruktur angewendet wird; die Rasterstruktur wird dadurch geglättet und vereinfacht (Blow-Shrink).

Verschneidung von thematischen Ebenen: Aus zwei oder mehreren thematischen Ebenen wird eine weitere Ebene gebildet; dabei sind geometrisch-topologische Schnittalgorithmen und thematische Zusammenhänge auszuwerten.

Verteiltes System: Architektur einer Datenbank oder eines Informationssystems, das einen dezentralen Zugang erlaubt.

View: Synonym für Bearbeitungsauszug.

Volumenmodell: Dreidimensionales Modell, das auf elementaren Bausteinen wie Würfel, Quader, Zylinder, Kegel usw. beruht.

Voronoi-Graph: Dualer Graph einer DIME-Struktur, der auch noch eine metrische Bedeutung hat: jede seiner Kanten ist äquidistant zu den beiden jeweils benachbarten DIME-Knoten (auch Thiessen-Polygon).

Weg: Zusammenhängende Folge von Kanten, die von einem Knoten zu einem anderen Knoten führt.

Wissensbank: Teil eines *Expertensystems*; speichert elementare Fakten und Regeln.

Wissensgestütztes Informationssystem: Es ist dies ein *Informationssystem*, das um bestimmte Teilaspekte eines *Expertensystems* erweitert ist ("knowledge-based information system").

Zentroid: Ein Punkt (eine Rasterzelle), der (die) an einer markanten Stelle (etwa dem Zentrum) einer geometrisch-topologischen Struktur liegt und somit zur eindeutigen Kennzeichnung dieser Struktur herangezogen werden kann.

Zufallsvariable: Funktion, deren Werte nicht von einer deterministischen Vorschrift, sondern "vom Zufall" bestimmt werden.

Zugriffspfad: Ein vom Datenbankverwaltungssystem bevorzugt behandelter Weg zur Identifikation von *Realisierungen* einer *Datengruppe*.

Zusammenhängender Graph: Zu je zwei beliebigen Knoten dieses Graphen muß es mindestens einen Weg geben.

Zyklus: Man kann von einem Knoten ausgehen und auf einem anderen Weg zu diesem Knoten zurückkehren.



ANHANG C
SACHVERZEICHNIS

A

Ableitungsmechanismus 227
 Abstandsbegriff *s. Metrik*
 Abstandstransformierte 163, 167
 Abstraktionsprozeß *s. Modellbildung*
 Adreßraum 132-134, 137, 138
 Advice-giving systems 244
 Aggregieren 198
 AI *s. Artificial Intelligence*
 Anwenderprofil 203-204, 208, 212, 215, 220, 233, 243
 Anwenderprogramm 5-6, 13-14, 91, 95, 102, 103, 115-116, 120-121, 140, 170, 206, 218, 219
 Anwenderschnittstelle 209, 210, 219, 233
 Anwenderspezifische Techniken 244
 Anwendersprache 208
 Approximation
 Flächen 179-193
 Kurven 171-173, 177-179
 Arbeitsbereich *s. work file*
 Arbeitsplatz 6, 8, 217
 Konzept 6, 8
 Arbeitsstation *s. Arbeitsplatz*
 Areales Modell 21
 Arrayprozessor *s. Parallelprozessor*
 Artificial Intelligence 215, 221, 224-244
 Anwendungen 239-244
 Formen 226
 Attribute 95-99, 105
 Thematische *s. Thematik-Attribute*
 Ausgleichung 177, 187, 191, 197-198, 207, 219

B

Backward chaining 230, 236
 Baumstruktur 46, 68, 76-79, 99, 124, 128-131, 133, 136, 142, 206, 235
 Bearbeitungsauszug *s. View*
 Bemaßung 43, 81
 Benützerschnittstelle *s. Anwenderschnittstelle*

Beschriftung *s. Text*
 1:1-Beziehung 94-95, 96
 1:M-Beziehung 73, 76, 94-95, 96, 99, 104-105, 141
 M:1-Beziehung *s. 1:M-Beziehung*
 M:N-Beziehung 73, 94-95, 100-105
 Beziehung, hierarchische 76
 Bezierkurve 176, 194
 Bildschirmgestaltung 210
 Bildverarbeitung 14, 158, 164
 Bilineare Interpolation 185, 190
 Bit plane 69
 Bit slicing 69
 Blattschnittproblematik 48, 135-139, 214
 Blow-shrink 160-162, 164
 Bruchkante 182, 187, 194
 Bulk 122, 141, 144

C

CAD 42, 176, 180
 Chain code 45, 68, 125
 CODASYL-Datenbank 100
 5. Computergeneration 224

D

Darstellungsvielfalt 208
 Datenbank 5, 90-118, 120-146, 205, 216, 219
 Relationale *s. Datenbankmodell, relationales*
 Verteilte *s. System, verteiltes*
 Datenbankmodell 97, 99-103
 Hierarchisches 99-100
 Relationales 80, 101-117, 120, 141, 216
 Datenbankschemata 92-93
 Datenbankverwaltungssystem 5, 91, 145, 217, 240
 Datenerfassung 8, 14, 171, 181, 182, 199, 206, 224
 Datengruppe 77, 94-99, 104-105
 Datenkonsistenz *s. Konsistenz*

Datenmodell 10, 18-88
 Datenraum 132-134, 137
 Datenschutz 91, 93, 110,
 115, 116, 120, 121, 204
 Datenstruktur
 Raumbezogene s. *raumbezo-*
 gene Datenstruktur
 Stabile 103-108
 Datumsproblem 207
 DBMS s. *Datenbankverwal-*
 tungssystem
 Deep & narrow 231
 DGM s. *Geländemodell*
 DHM s. *Höhenmodell*
 Dialoggestaltung 208
 Dialogkomponente 229, 233
 Digitale Karte 2, 5
 Digitaler Plan 2, 5
 Digitales Geländemodell
 s. *Geländemodell*
 Digitales Höhenmodell
 s. *Höhenmodell*
 DIME-Struktur 37-38, 140
 Directory 132-134, 137, 138,
 139
 Diskzugriff 127, 133
 Domäne 101, 103
 Drahtmodell 22
 Dreiecksvermaschung 36, 140,
 192-193
 Dualität 38, 140, 155

E

Ebene, thematische s. *thema-*
 tische Ebene
 Entscheidungsfindung, mensche-
 liche 225, 226
 Erfassung von Daten s. *Daten-*
 erfassung
 Ergonomie 210, 219-220
 Erklärungskomponente 229,
 233
 Ersetzungssystem 229
 EXCELL 132-133, 138, 142
 Expertensystem 85, 208,
 227-231, 240, 241, 242
 Anwendungen 228
 Externes Schema
 Datenbank 20, 79, 92-93
 Informationssystem 217

F

Fakten 227
 Falschfarbenbild 68
 Faltung 174
 Feature extraction 239
 Fehler
 Systematischer 178-179
 Zufälliger 178-179
 Filterung 161, 164-168, 195
 Finite Elemente 190-191
 Fläche 31-36, 71-72, 74-76,
 90, 135, 162
 Flächenbildung, automatische
 199, 232
 Folienkonzept, thematisches
 s. *thematisches Folien-*
 konzept
 Format
 Graphisches 208
 Numerisches 207
 Formparameter 27, 36, 90
 Forward chaining 230, 235
 Fouriertransformation 168,
 169
 Fraktale Dimension 52
 Fraktale Geometrie 50
 Fraktales Modell 49-58, 180
 Frame 81, 94, 236, 237-238,
 240
 Frequenz eines Grauwertbildes
 158, 164, 166, 167
 Füllen von Flächen 161,
 162-163
 Füllungskeim 162-163
 Fuzzy data 230

G

Gauß'sches Minimumsprinzip
 177
 Geländemodell 148, 153,
 179-194
 Genauigkeit 206
 Generalisierung 148, 161,
 166, 171, 188, 195-197,
 207, 219, 241
 Generator 50
 Geoinformationssystem 2, 5
 Geometrie 11, 12, 18-58,
 60-61, 90, 131, 141,
 144, 148, 167
 Absolute 207
 Gemessene 207

Geometrisch-topologisches
Modell 18-58
Geometrische Aspekte s. *Geometrie*
Geometrische Bedingungen
198, 241
Gewichtung 165, 176, 187,
191, 196
GIAP s. *Graphisch-interaktiver Arbeitsplatz*
GIS s. *Geoinformationssystem*
GKS 6
Glättung 161, 188, 194
Gleitendes Mittel 187-188,
190
Graph 38, 79, 85-88, 99,
122, 124, 128, 199, 235
Graphentheoretische Konzepte
der KI 235-236
Graphisch-interaktiver
Arbeitsplatz 6-7
Graphisch-interaktives Anwen-
derprogramm 120-121, 140
Grauwert 68, 83, 84, 123,
129, 158-171
Parallelverschiebung 159,
160, 163
Selektion 159
Arithmetische Kombination
159
Logische Kombination 159,
160, 163
Grid file 133, 138, 142

H

Hashing 133-134
Hierarchie, thematische
s. *thematische Hierarchie*
Hochpaßfilterung 164, 166
Höhenmodell 36, 140, 153,
180-194
Höhenschichtenmodell 182-183
Homogene Koordinaten
150-152, 154
Hybride Methoden 48-49, 129
Hybrides Modell 47, 82-84
Hypothese 230

I

Index 98-99, 121, 131, 142
Inference engine s. *Problem-
lösungskomponente*
Informationsgehalt 4, 167,
195
Initiator 50
Insel 21, 24, 31-35, 74-75,
90, 135, 162, 166, 183
Instanz 77, 81, 94, 235, 238
Intelligenz, menschliche 236
Modell 226-227, 236, 237
Internes Schema 19, 92-93
Interpolation
Flächen 179-193
Kurven 171-177
Isolinien 65
Isomorphie von Graphen 86

J

Join von Relationen 111
Journalfile 215, 216

K

Kanonische Synthese 108
Kante 27-43, 74-76, 85-88,
90, 141, 155, 193
Kanten eines semantischen
Netzes 237
Kanten-Knoten-Struktur
29-35, 85-88, 141, 199
Karte 1, 2, 4, 60-65, 195,
200
Kartengestaltung 241
Kartesisches Produkt von
Relationen 110
Kataster 15, 60, 76, 207
KD-Baum 124, 128, 132
Kettencode s. *Chain code*
Key s. *Schlüssel*
KI s. *Künstliche Intelligenz*
Klassifikation, radiometri-
sche 239
Knoten 27-43, 74-76, 85-88,
122, 140, 141
in semantischen Netzen 237
Kognitiv 227, 228, 236
Kollinearität 154
Kollokation 189
Kommando 209

Konnektionismus 237
 Konsistenz
 Daten 12, 77, 82, 84-88,
 90-92, 120, 121, 140,
 144, 145, 204, 205, 212,
 216, 217, 219, 232, 240,
 241
 Regeln 233
 Konsistenzbedingung s. Kon-
 sistenz
 Konsistenzüberprüfung s. Kon-
 sistenz
 Kontrollsystem 229
 Konvexes Polygon 156, 157
 Konvolution 164-166
 Konzeptionelles Schema
 Datenbank 19, 79, 92-93
 Informationssystem 217
 Kovarianz 190
 Künstliche Intelligenz 215,
 221, 224-244
 Formen 226

L

Landinformationssystem 2, 5
 Langzeitspeicherung 90, 103,
 120, 144, 146, 211, 216,
 217, 240
 Lauflängencodierung s. Run
 length encoding
 Legende 200
 Leitpunkte 175-176
 Lernfähigkeit 241, 243
 Lernmodul 228
 Line adjacency graph 124,
 140
 Lineales Modell 21
 Lineare Interpolation
 183-184
 Lineare Prädiktion 189-190
 LIS s. *Landinformationssystem*
 Log-file 115, 243
 Interpreter 243
 Lookup-Table 65, 69, 208
 Lückenfüllung 160-161, 166

M

Makrofunktion 220
 Makrooperationen mit Raster-
 daten 160-169

Mehrzweckkataster 15
 Menu 209, 221
 Pop-up 209
 Soft 209
 Meßfehler 172
 Metadaten 117
 Metaregeln 227
 Methodenbank 5
 Metrik 30, 45, 85, 125, 159,
 162, 163
 Modellbildung 3, 10, 18-88,
 171-173, 179-183, 195
 Morphologische Anordnung
 181-182
 Multi-User-Betrieb 91, 144,
 211, 216-217
 Mustererkennung 199, 237

N

Natürliche Sprache 233, 242
 Navigieren 102
 Netz
 Topologisches 36
 Zusammenhängendes 37
 Netzwerk
 Arbeitsstationen 91, 116,
 120, 146, 217-218
 Datenbankmodell 100-101
 Nicht-monoton 230, 240
 Nicht-prozedural 102, 115,
 231, 232
 Normalformen 104-108
 Normalgleichungen 177, 187,
 191

O

Objekt 77-79, 81, 84,
 134-140, 141, 205, 206,
 213, 238, 243
 Zerschneidung 135-139,
 213

P

Page 127-139, 142
 Parallelprozessor 157,
 170-171, 237
 Paßpunkt 13, 169, 197
 Peano-Kurve 53

Perspektive 152-153
 Physische Speicherung 126,
 130, 218
 Pixel 21, 44, 161, 172
 Plan 1, 2, 4, 5
 Aufbereitung 121, 146, 200
 Pointer 100, 101
 Prädikatenlogik 112
 Pre-compiler 116
 Primärdaten 182, 190, 192
 Problemlösungskomponente
 227, 230, 232, 234
 Produktraum 67
 Projekt 144-146, 204, 211,
 212, 213, 214, 215, 216,
 218
 Projektbeschreibung 215, 216
 Projektierung, Varianten 211
 Projektion von Relationen
 109-110
 Projektive Geometrie 152
 Projektprofil 215, 220
 Prop 238
 Prozedural 232, 242
 Pseudokante 32
 Pufferzonen 148, 198
 Punktinformation 26, 92

Q

Quad tree 124, 128

R

Rahmen *s. Frame*
 Randausgleichung 169, 197
 Rang-Deskriptor 197
 Raster
 Daten *s. Rastermodell*
 Modell 21, 44-47, 66-70,
 82-84, 122, 123-125, 134,
 157, 181
 Operationen 157-171
 Strukturen *s. Rastermodell*
 Zelle 44, 66, 123, 125,
 158-167, 170-171
 Rasterung 200
 Raumbezogene Datenstruktur
 93, 120-146
 Raumbezogener Zugriff
 120-146

Raumbezogenes Informations-
 system
 Aspekte 5, 18, 60, 79, 83,
 93, 100, 103, 108, 126,
 137, 140, 143, 148, 171,
 179, 197, 203, 206, 216,
 224, 232
 Beispiele 15-16
 Daten 10-12
 Definition 5-8
 Konsumenten 221
 Schalen 203, 219, 220, 233
 Werkzeuge 13-14
 Raumbezug 120, 139, 140,
 142, 146
 Räumliche Auflösung 44
 Rauschen 179, 189
 Realisierung einer Daten-
 gruppe *s. Instanz*
 Redundanz 105-107
 Regeln 225, 227, 228, 230,
 232, 233, 235, 236, 240,
 241, 242
 Region adjacency graph 38
 Relationale Algebra 108-112,
 140, 215
 Relationale Datenbank
s. Datenbankmodell,
relationales
 Relationaler Kalkül 102, 112
 Relationen, Elementaroperati-
 onen 108-111
 Resampling 169
 Ring, äußerer 32
 Ring, innerer 32
 RIS *s. raumbezogenes Informa-*
tionssystem
 Robotics 227
 Role 238
 Roll back 115, 146, 217
 Roll forward 115, 146, 217
 Run length encoding 45, 68,
 123

S

Sampling 44
 Scene 238
 Schätzung, beste 190
 Schema einer Datenbank *s. Da-*
tenbankschemata
 Schichtenlinien 171-172,
 181-182, 194-195

Schlüssel 22, 80, 95-99,
104-110, 143
Schnittberechnung 25, 48,
148, 154-157
Schraffur 33, 162, 200, 208
Schwellwert 158, 160, 165
Script 238, 243
Sekundärdaten 182, 190, 192
Selbst-Ähnlichkeit 50
Selektion
 Daten 144-146, 211, 212
 Relationen 110
Semantische Sprachanalyse
 242
Semantisches Netz 236-238
Signal 189
Signatur 71, 161, 199, 200
Simulationsstudien 58
Skelett 34-35, 38, 46, 68,
163
Slot 81, 238
Software-Werkzeuge 13-14,
148-200, 219, 224
Spektrum von Grauwerten 158,
239
Spline 173-176, 191
Split-and-Merge-Prinzip 178
SQL 112-116
Standardeinstellung 210
Stochastischer Prozeß 55
Surrogat 105
Symbol
 Freistellung 40, 121, 148,
198
 Kartographisches 39, 148,
161, 163, 208
 Plazierung 40, 121, 148,
198, 241
Syntaktische Sprachanalyse
 242
System
 Attraktivität 210
 Benutzerfreundlichkeit 210
 Logisches Konzept 6
 Offenes 10, 220
 Verteiltes 6, 116, 211,
217

T

Tabelle 80, 101-115, 141
Tastenbelegung 210
Teragon 50

Text

Freistellung 42, 141, 148,
198
Kartographischer 41, 148
Plazierung 42, 141, 148,
198, 241
Thema s. *Thematik*
Thematik 1-4, 60-66
 Aspekte 1-4, 11, 12, 25,
26, 59-88, 131, 141, 144,
148
 Attribute 80-82, 83, 205,
206, 212, 238
 Dimension 67
 Ebene 63-64, 73, 83, 204
 Elemente 74-88, 205
 Folienkonzept 64, 78
 Hierarchie 73, 76-79, 81,
205, 241
 Karte 61
 Kartographie 60, 158
 Modell 59-88
Themenzuweisung, automatische
 239
Thiessen-Polygon 38, 140
Tiefpaßfilterung 164, 166
Topographie 11, 16, 61
Topologie 11, 12, 18-58,
60-61, 90, 131, 139, 140,
141, 144, 148, 167
Topologische Äquivalenz 27
Training, computer-based 244
Trainingsgebiet 188, 240
Transaktion 82, 87-88, 121,
145-146, 216, 217, 218,
243
Transferfunktion 158
Transformation
 Geometrische 148, 149-152,
169, 207
 Radiometrische 158
Trend 179, 189
Treppeneffekt 164, 166
Tupel 101, 104, 110, 240

U

Umweltkataster 15
Umweltschutz-Anwendungen 244
Und-oder-Graph 235-236
Update-Berechtigung 217
Update-Synchronisation 217

V

Vages Wissen 230, 233, 238
 Vektor
 Daten s. *Vektormodell*
 Modell 20-43, 70-76,
 82-84, 122, 126, 149,
 154, 181
 Strukturen s. *Vektormodell*
 Vektorisierung 161, 200
 Verdicken-Verdünnen s. *Blow-Shrink*
 Verdrängungsproblematik 40,
 42, 167, 196, 241-242
 Verkehrsleitsysteme 16
 Verschneidung thematischer
 Ebenen 64, 73
 View 43, 103, 113, 115, 145,
 216, 219, 232, 240
 Volumenmodell 23, 183
 Voronoi-Graph 38, 140

Z

0-Zelle 37
 1-Zelle 37
 2-Zelle 37
 Zentroid 84, 136-138
 Zufallsvariable 189
 Zugriffsberechtigung 204,
 212, 213, 214, 216, 217
 Zyklen in Graphen 86-87, 99

W

Wahrscheinlichkeit 230, 240
 Wahrscheinlichkeitsbewertete
 Aussagen 230, 242
 Wertebereich 206
 Wide & shallow 231
 Wissen 225, 229, 237, 238
 Externalisierung 225
 Netzstrukturen 225
 Vages s. *vages Wissen*
 Wissensbank 227, 229, 232,
 242
 Wissensbasis s. *Wissensbank*
 Wissensbereitstellung 229
 Wissensgestütztes Informa-
 tionssystem 231-234, 241
 Work file 215