

Interaktive Karten – adaptives Zoomen mit Scalable Vector Graphics

Diplomarbeit von Tobias Brühlmeier

Geographisches Institut der Universität Zürich, 2000

Betreuung durch Prof. Dr. Kurt Brassel und Dr. René Sieber

Vorwort

Computer können in der modernen Kartografie einerseits als Hilfsmittel in der Produktion eingesetzt werden. Im Zeitalter von World Wide Web und Multimedia findet andererseits der Computer immer häufiger auch seine Anwendung in der Ausgabe von Karten direkt auf dem Bildschirm. Der Computer wird zum Nachschlagewerk, die Bildschirmkarte zu einem neuen Informationsmedium.

Keinesfalls sollen aber Bildschirmkarten Papierkarten ersetzen. Papierkarten haben zu viele Vorteile, allen voran steht die hohe Ausgabequalität. Wichtigstes Ziel von Bildschirmkarten soll es sein, Papierkarten zu ergänzen und mit Hilfe multimedialer Methoden den totalen Informationsgehalt zu erhöhen. Der grösste Nachteil einer Papierkarte, nämlich die Beschränkung auf einen einzelnen Zustand, ist mit digitalen Karten aufgehoben. Es wird z. B. möglich, dass sich die Benutzerin einen groben Überblick über eine Region verschaffen, oder mit einem Zoomwerkzeug dieselbe Region stark vergrössert und detailliert betrachten kann.

In der vorliegenden Arbeit wird die Entwicklung von adaptiv zoombaren Karten beschrieben. Nicht die technische Seite wird dabei im Vordergrund stehen, sondern viel mehr die kartografischen Aspekte.

Ich möchte mich an dieser Stelle bei allen Personen bedanken, die durch konstruktive Kritik oder Beisteuerung guter Ideen zum Gelingen dieses Projektes beigetragen haben.

- ◆ Insbesondere zu erwähnen ist *Dr. René Sieber*, Projektleiter des «Atlas der Schweiz interaktiv» am Institut für Kartographie der ETH Zürich, der als mein Betreuer alle Manuskriptversionen kritisch durchgelesen und viele Verbesserungsvorschläge eingebracht hat. Im Verlaufe des Projektes hat er sich immer wieder viel Zeit für mich genommen.
- ◆ Ebenso bin ich *Prof. Dr. Kurt Brassel* vom Geografischen Institut der Universität Zürich Irchel für seine Anregungen und Korrekturen dankbar.
- ◆ *Robert Huber*, Geschäftsführer der Softwarefirma 7r gmbh, war mehr als ein verständnisvoller Arbeitgeber, er war für mich auch ein guter Lehrer im Umgang mit *Jackson Structured Programming* und vermochte es immer wieder, mich zu einer produktiven und effizienten Arbeitsweise zu motivieren.
- ◆ *Hannes Rellstab*, Geografiestudent an der Universität Zürich Irchel, hat das Manuskript kritisch nach typografischen Fehlern durchgesehen und auf Verständlichkeit geprüft.
- ◆ Dankbar bin ich auch allen, die der Aufforderung, meine interaktive Karte zu testen und zu kritisieren, gefolgt sind, vor allem *Anton Shevchenko* aus Russland und *Dr. Franz-Josef Behr* aus Deutschland.
- ◆ Für ihr Verständnis möchte ich mich abschliessend bei denjenigen Leuten bedanken, die meine Begeisterung für interaktive Karten und *Scalable Vector Graphics* in langen, meist unaufgeforderten Vorträgen zu spüren bekommen haben.

In der vorliegenden Arbeit werden öfters Personen in ihrer Tätigkeit oder Funktion (etwa «Benutzerin» oder «Kartenautor») genannt. Ich habe mich bemüht, etwa gleich oft wie die männliche auch die weibliche Form zu benutzen. Dies sei nicht in irgend einer Form wertend oder politisch zu verstehen.

Zürich, September 2000, Tobias Brühlmeier

Zusammenfassung

Die vorliegende Arbeit hat zum Ziel, eine Bildschirmkarte zu entwickeln, die von einem Benutzer ohne Expertenwissen interaktiv im Massstab verändert werden kann; mit der Änderung des Massstabs soll sich auch die Detailgenauigkeit der Karte anpassen. Gleichzeitig ist es die Absicht, die Entwicklung weitgehend unabhängig von der Form der Implementation zu halten; die Methoden und Konzepte, die hier entwickelt werden, sollen als Grundlage für andere, ähnliche Projekte dienen.

Im ersten Teil der Arbeit werden die Rahmenbedingungen der Bildschirmausgabe von Karten abgesteckt. Die Analyse beschränkt sich aber auf diejenigen Themen, die später für adaptives Zoomen nützlich sind. Die technischen Grundlagen, die Vor- und Nachteile der Bildschirmausgabe, die Regeln der Kartengestaltung und die Interaktion mit Bildschirmkarten werden untersucht.

Bildschirme haben offenkundig im Vergleich mit Papier grosse Nachteile als Ausgabemedium für Karten. Insbesondere sind die schlechtere Auflösung und die der Kartenautorin nicht bekannte Qualität des vom Benutzer verwendeten Computersystems zu erwähnen. Die reine kartografische Darstellung von Inhalten ist auf Papier also besser und übersichtlicher.

Eignung von Bildschirmen zur Ausgabe von Karten

Die auf Bildschirmen möglichen Mittel der Interaktion bieten aber der Kartenautorin ein neues Potenzial der Informationsvermittlung. Die Vermittlung komplexer räumlicher Daten ist mit interaktiven Anwendungen möglich, ohne dass vom Benutzer Expertenwissen verlangt werden muss.

Eine Aufstellung aller Anforderungen der Bildschirmkartografie an eine Entwicklungsumgebung ergibt, dass die neue, XML-basierte Technologie SVG für interaktive Karten wohl am besten geeignet ist: Nicht nur durch den grossen Funktionsumfang zeichnet sich SVG aus, sondern auch durch Unterstützung von Softwareherstellern auf breiter Front.

SVG als Entwicklungsumgebung für Bildschirmkarten

Die Möglichkeiten, die SVG der Kartografie zu bieten hat, sind riesig. Dynamische Kartenproduktion, mediumsorientierte Ausgabe, unterschiedliche Sprachversionen, dynamische Legendenfunktion und natürlich adaptive Karten sind einige Stichworte.

Die Grundsätze zur Kartengestaltung, wie sie für die Papierausgabe gelten, haben mit Einschränkungen auch für Bildschirmkarten ihre Gültigkeit. Im Hinblick auf adaptives Zoomen bedürfen aber insbesondere grafische Mindestgrössen und die Kriterien für die Schriftwahl der Modifikation.

Der zweite Teil der Arbeit beschäftigt sich mit den Grundlagen von interaktiver Massstabsänderung durch adaptives Zoomen, wobei «*adaptives Zoomen*» ist folgt definiert ist: «Ein Zoomprozess ist dann adaptiv, wenn sich die Darstellung einer Karte so an den Kartenmassstab respektive den Zoomfaktor anpasst, dass die Kartenqualität immer als gut empfunden wird. Dies gilt insbesondere für den Informationsgehalt, die Informationsdichte und die grafische Darstellung aller Signaturen.»

Das Konzept «adaptives Zoomen»

Eine adaptive Karte besteht immer aus der Kartengeometrie, die im adaptiven Datensatz enthalten ist, und einem Programm, das die Steuerung aller notwendigen Prozesse übernimmt.

Der adaptive Datensatz setzt sich aus mehreren, sich massstäblich unterscheidenden Geometriesätzen zusammen, die jeweils in bestimmten Grenzen skalierbar sind. Jeder dieser Geometriesätze deckt einen bestimmten Massstabsbereich ab; die Massstabsbereiche zweier aneinander anschliessender Geometriesätze überlappen sich im Optimalfall. Jeder Geometriesatz kann pro Objektgruppe verschiedene Dichtestufen haben.

Beim Zoomprozess wird, ausgehend von der äussersten Zoomstufe, ein Geometriesatz so lange vergrössert, bis sein Vergrösserungslimit erreicht ist. Gleichzeitig mit der Vergrösserung des Massstabs müssen neue Dichtestufen einzelner Objektgruppen hinzukommen, und zwar so, dass die Infor-

mationsdichte konstant bleibt. Wenn das Vergrößerungslimit erreicht ist, wird der Geometriesatz aus- und ein neuer, auf den anschliessenden Zoombereich optimierter Geometriesatz eingeblendet.

Um die Qualität der kartografischen Darstellung beibehalten zu können, müssen Signaturmasse bei der Vergrößerung das Massstabs in der Dimension relativ zu ihrer Umgebung kleiner gewählt werden können als im Vorläufermassstab. Die Signaturmasse müssen jeweils so bemessen sein, dass alle kartografischen Regeln eingehalten werden können.

Die verschiedenen Geometriesätze müssen gut aufeinander abgestimmt sein. Diese Bedingung ist besonders schwierig zu erfüllen, da sich die Geometriesätze, weil sie für verschiedene Massstäbe gelten, in der Generalisierung unterscheiden.

Die Entscheidung, welcher Geometriesatz welche Skalierungsgrenzwerte hat, kann nicht mit mathematisch erfassbaren Kriterien gefällt werden, da die Qualität der Generalisierung und der formale Ausdruck der Darstellungskomplexität nicht möglich sind. Die Skalierungsgrenzwerte liegen ungefähr in den Bereichen von 50% bzw. 200% des Originalmassstabs, können aber je nach Karte recht unterschiedlich sein.

Mit grösser werdendem Massstab vergrössert sich auch die zur Verfügung stehende Fläche, wenn die Signaturmasse angepasst werden. Diese Fläche kann mit neuen und zusätzlichen Informationen gefüllt werden. Der Informationsgehalt muss dabei in einem solchen Ausmass vergrössert werden, dass die Informationsdichte konstant bleibt.

Neue Dichtestufen kommen dann hinzu, wenn der Informationsgehalt-Grenzwert überschritten wird. Der Informationsgehalt-Grenzwert hängt vom Darstellungsgewicht einzelner Objekte und Objektgruppen ab, wobei kontrastreiche und massive Liniensignaturen eher schwer und kleine, feine Punktensignaturen leicht sind. Die Informationsgehalt-Grenzwerte sind mathematisch nicht erfassbar und müssen mittels Versuchen festgelegt werden.

Die Signaturmasse (sowohl Linien-, als auch Punktensignaturen und Schriftgrössen) können mit einer Funktion an den Massstab angepasst werden, die das ursprüngliche und das neue Signaturmass, den ursprünglichen und den neuen Massstab, den maximalen Skalierungsfaktor für das Signaturmass und die Annäherungsgeschwindigkeit an diesen Maximalwert berücksichtigt.

Durch die Veränderung der Schriftgrösse verschieben sich Labels von Punktensignaturen relativ zu ihrer ursprünglichen Position. Je nachdem, ob ein Label sich links, rechts, oberhalb oder unterhalb der zugehörigen Punktensignatur befindet, muss es anders behandelt werden.

Ein Label wird entsprechend seiner Position so in die richtige Richtung geschoben, dass bei der Massstabsänderung weder der horizontale, noch der vertikale Abstand zwischen dem Label und dem Rand der Punktensignatur verändert wird. Natürlich muss dabei auch die Grössenänderung der Punktensignatur berücksichtigt werden.

Zur eigentlichen Implementationsphase des Projektes gehören der Aufbau des adaptiven Datensatzes und das Steuerprogramm für die adaptive Karte. Beide Teile sind stark praktisch orientiert.

Es wird einerseits beschrieben, auf welche Systematiken beim Aufbau des adaptiven Datensatzes im Grafikprogramm Wert zu legen ist. Dies schliesst die Anpassung der Signaturmasse und der Labelpositionen in Abhängigkeit von den Skalierungsgrenzwerten einzelner Geometriesätze mit ein, aber auch die Bezeichnung der Ebenen, die später als Identifikation einzelner Objektgruppen verwendet werden. Ausserdem ist der Aufbau der SVG-Anwendung ein wichtiges Thema.

Andererseits wird die logische Abfolge des Steuerprogramms ausführlich geschildert. Das Steuerprogramm muss grundsätzlich zwei Abläufe steuern: hauptsächlich den Zoomprozess, daneben aber auch die Prozesse, die ablaufen, wenn die Karte geladen wird.

Informationsdichte und Informationsgehalt

Anpassung der Signaturmasse

Anpassung der Beschriftung von Punktensignaturen

Analyse der Implementationsphase

I Inhalt

I	Inhalt	v
II	Abbildungen	ix
III	Tabellen	xi
1	Zielsetzung und Überblick	1
1.1	Inhaltsüberblick	1
1.1.1	Allgemeiner Teil	1
1.1.2	Projektspezifischer Teil	2
1.2	Konventionen in der Layoutgestaltung	3
1.2.1	Die Randspalte	3
1.2.2	Konventionen in der Textauszeichnung	3
2	Der Bildschirm als Ausgabemedium für Grafik	5
2.1	Darstellung von Grafik auf dem Bildschirm	5
2.1.1	Darstellung von Pixeln	5
2.1.2	Darstellung von Farbe	5
2.1.3	Auflösung	6
2.1.4	Grafische Mindestgrößen	7
2.1.5	Antialiasing	7
2.2	Wahrnehmung von Grafik auf dem Bildschirm	7
2.2.1	Ergonomische Aspekte	7
2.2.2	Wahrnehmung von Farbe	8
2.2.3	Adaptionsfähigkeit des menschlichen Auges	8
2.3	Ausgabe von Grafik auf dem Bildschirm	9
2.3.1	Ausgabe von Grafik in Webbrowsern	9
2.3.2	Ausgabe von Grafik in Autorentools	9
3	Der Bildschirm als Ausgabemedium für Karten	11
3.1	Wichtigkeit der dynamischen Struktur von Karten	11
3.2	Unterschiede zwischen Papierkarten und Bildschirmkarten	11
3.2.1	Ausgabequalität und grafische Mindestgrößen	11
3.2.2	Konsistenz in der Ausgabe	12
3.2.3	Handlichkeit	12
3.2.4	Benutzerpotenzial	13
3.2.5	Entwicklungspotenzial	13
3.2.6	Interaktivität	13
3.2.7	Nachführung	13
3.3	Eine geeignete Entwicklungsumgebung für Bildschirmkarten	13
3.3.1	Unterschiedliche Dateiformate	14
3.3.2	Kartografische Anforderungen an ein Dateiformat	14
3.3.3	SVG als Dateiformat für Bildschirmkarten	15
4	Gestaltung von Bildschirmkarten	17
4.1	Qualität der Kartendarstellung	17
4.2	Grundsätze zur Kartengestaltung	17
4.2.1	Aufbau des kartografischen Zeichensystems	18
4.2.2	Grafische Variation der Zeichen	18
4.2.3	Kartenlogische Bedingungen für die Kartengrafik	18
4.2.4	Gestaltwahrnehmung	19

4.2.5	Kartografische Gestaltungsmittel	19
4.2.6	Grafische Mindestgrößen	19
4.2.7	Wahl der Kartenschrift	20
4.2.8	Anordnung der Kartenbeschriftung	20
4.3	Technische Qualität der Kartendarstellung auf Bildschirmen	21
4.3.1	Einfluss des Antialiasing	22
4.3.2	Einfluss der Bildschirmauflösung	22
4.4	Bildschirmspezifische Modifizierungen der Grundsätze zur Kartengestaltung	22
4.4.1	Grafische Mindestgrößen	23
4.4.2	Wahl der Schrift für Bildschirmkarten	24
4.5	Bildschirmspezifische Vergrößerung der Kartengeometrie	24
5	Interaktion mit Bildschirmkarten	27
5.1	Mentale Schnittstelle: Metamodell für die Repräsentation	27
5.1.1	Eintauchen	28
5.1.2	Interaktivität	28
5.1.3	Informationsintensität	28
5.1.4	Intelligenz der Anzeigeobjekte	28
5.2	Visuelle Schnittstelle: Bedeutung von grafischen Benutzeroberflächen	28
5.3	Möglichkeiten der Interaktion	29
5.3.1	Funktionen zur thematischen Navigation	29
5.3.2	Funktionen zur räumlichen Navigation und Orientierung	29
5.3.3	Funktionen zur Visualisierung	29
5.3.4	GIS-Funktionen	30
6	Technische Grundlagen von SVG	31
6.1	XML im Überblick	31
6.1.1	Die Konzepte von XML	32
6.1.2	XML Document Markup	33
6.1.3	Physische Struktur einer XML-Anwendung	33
6.1.4	Logische Struktur einer XML-Anwendung	34
6.1.5	Dokumentformatierung mit CSS	34
6.1.6	Das Document Object Model	35
6.1.7	Interaktivität mit JavaScript	36
6.2	SVG im Überblick	36
6.2.1	Die Konzepte von SVG	37
6.2.2	Das Rendering-Modell	38
6.2.3	Struktur eines SVG-Dokumentes	38
6.2.4	Styling Properties	39
6.2.5	Koordinatensysteme und Transformationen	39
6.2.6	Path-Elemente	39
6.2.7	Text-Elemente	39
6.2.8	Kontrolle über die Sichtbarkeit	40
6.2.9	Interaktivität	40
6.3	SVG für Bildschirmkarten	40
6.3.1	Dynamische Kartenproduktion	40
6.3.2	Mediumsorientierte Ausgabe	40
6.3.3	Internationalisierung	41
6.3.4	Dynamische Legende	41
6.3.5	Interaktive und dynamische Stilveränderungen	41

7	Interaktive Massstabsveränderungen durch adaptives Zoomen	43
7.1	Hintergrund	43
7.1.1	Abgrenzung von grafischen GIS-Produkten und interaktiven Karten	43
7.1.2	Massstabsveränderungen in grafischen GIS-Produkten	44
7.1.3	Massstabsveränderungen in kartografischen Produkten	44
7.2	Konzept und Definitionen	45
7.3	Projektspezifische Ziele	46
7.3.1	Kartografische Ziele	46
7.3.2	Funktionelle Ziele	46
7.3.3	Inhaltliche Ziele	47
7.3.4	Technische Ziele	47
7.4	Der adaptive Datensatz	48
7.4.1	Kartografische Qualität	48
7.4.2	Technische Qualität	48
7.4.3	Auswahl des richtigen Geometriesatzes	48
7.4.4	Zeitlich versetztes Auswechseln von Objektgruppen	49
7.5	Informationsdichte und Informationsgehalt	49
7.5.1	Das Darstellungsgewicht von Objekten und Objektgruppen	50
7.5.2	Informationsgehalt-Grenzwerte	50
7.6	Anpassung der Signaturmasse: Liniensignaturen	51
7.7	Anpassung der Punktsignaturen	54
7.7.1	Positionskorrektur entlang der x-Achse	55
7.7.2	Positionskorrektur entlang der y-Achse	55
7.8	Beschriftung von Punktsignaturen	55
7.8.1	Massstabsabhängige Optimierung der Schriftgrösse	56
7.8.2	Platzierung der Kartenbeschriftung	56
7.8.3	Adaptive Platzierung der Kartenbeschriftung	57
7.8.4	Platzierung der Kartenbeschriftung im Grafikprogramm	58
8	Entwicklungsumgebung und Ausgabeplattform	59
8.1	Die Entwicklungsumgebung und deren Möglichkeiten	59
8.1.1	Produktion eines SVG-Dokumentes mit Illustrator	59
8.1.2	Adobe GoLive als JavaScript-Editor	60
8.1.3	BBEdit Lite als SVG-Editor	60
8.2	Der Adobe SVG-Viewer mit Netscape Communicator als Ausgabeplattform	60
8.2.1	Bedienung des Adobe SVG-Viewer	61
8.2.2	Einschränkungen des Adobe SVG-Viewer	61
9	Aufbau des adaptiven Datensatzes	63
9.1	Ausgangsmassstab und maximale Vergrößerung	63
9.2	Bearbeitung des adaptiven Datensatzes	64
9.2.1	Anpassungen für die Karte im Ausgangsmassstab	64
9.2.2	Festlegung der Skalierungsgrenzwerte	65
9.2.3	Provisorische Festlegung der Informationsgehalt-Grenzwerte	66
9.2.4	Anpassung der Signaturmasse auf alle Massstäbe	68
9.2.5	Anpassung der Kartenbeschriftung	68
9.3	Bezeichnung von Ebenen	69
9.4	Aufbau der SVG-Anwendung	70
9.4.1	Objektstile in der Document Type Definition DTD	70
9.4.2	Attributes des SVG-Tag	71
9.4.3	Inhalt des SVG-Dokumentes	71

9.5	Repetitive Anpassung der Kartengeometrie	72
9.5.1	Export aus Illustrator	72
9.5.2	Bearbeitung im Texteditor	72
10	Das Steuerprogramm für die adaptive Karte	73
10.1	Notation	73
10.2	Starten des Steuerprogramms für die adaptive Karte	74
10.2.1	White Space Handling	74
10.2.2	Einblenden der Objektgruppen für den Ausgangsmassstab	75
10.2.3	Zuweisen von Objektstilen: Die Funktion appendStyle	75
10.2.4	Unsichtbarmachen von Objektgruppen: Die Funktion removeVisibility	76
10.3	Der Zoomprozess	76
10.3.1	Anpassung der Objektsignaturen: Die Funktion adaptSigns	78
10.3.2	Berechnung der Punktsignaturgrössendifferenz: Die Funktion signShift	80
10.3.3	Berechnung der Versalhöhendifferenz: Die Funktion versalShift	80
10.3.4	Verschiebung der Objektgruppen: Die Funktion shiftGroup	80
10.3.5	Berücksichtigung der Labelgrössen: Die Funktion shiftIndividuals	80
10.3.6	Ermitteln der Buchstabengrösse: Die Funktion letterSize	81
11	Feintuning	83
11.1	Festlegung der Art des Wachstums von Signaturmassen	83
11.2	Festlegung der Informationsgehalt-Grenzwerte	83
12	Resultate	85
12.1	Allgemeine Resultate	85
12.1.1	Eignung von Bildschirmen zur Ausgabe von Karten	85
12.1.2	SVG als Entwicklungsumgebung für Bildschirmkarten	86
12.1.3	Verbesserung der Darstellungsqualität von Bildschirmkarten	86
12.2	Projektspezifische Resultate	86
12.2.1	Das Konzept «adaptives Zoomen»	86
12.2.2	Der adaptive Datensatz	87
12.2.3	Informationsdichte und Informationsgehalt	87
12.2.4	Anpassung der Signaturmasse	88
12.2.5	Anpassung von Punktsignaturen	88
12.2.6	Anpassung der Beschriftung von Punktsignaturen	88
12.3	Analyse der Implementationsphase	88
12.3.1	Kartografische Darstellung	89
12.3.2	Benutzerinteraktion	89
12.3.3	Karteninhalt	89
12.3.4	Implementation	90
12.4	Weiterführung des Projektes	90
12.4.1	Beschriftung weiterer Objekte	90
12.4.2	Schriftfreistellung	90
12.4.3	Verhalten von Farbe	91
12.4.4	Legenden für adaptive Karten	91
12.5	Persönlicher Nutzen der Arbeit	91
12.5.1	Neue Technologien	91
12.5.2	Projektmanagement	92
	Literatur	93

II Abbildungen

1	Ein Kartenausschnitt mit (1a und 1b) und ohne (2a und 2b) Antialiasing	7
2	Prioritätenreihenfolge für die Anordnung von Labels	21
3	Schwächen von Antialiasing	22
4	Darstellungsgrenzen von Strichstärken	23
5	Minimale Linienabstände	23
6	Minimale Flächendimensionen	23
7	Minimale Flächenzwischenräume	24
8	Logische und physische Struktur einer XML-Anwendung (nach Bradley, 2000:4)	32
9	Ein XML-Element mit Inhalt und seinen start und end tags (nach Bradley, 2000:17)	33
10	Document tree einer XML-Anwendung	33
11	Beziehungen zwischen Elementen	33
12	Die document entity mit weiteren entities (nach Bradley, 2000:35)	34
13	www.karto.ethz.ch/~an/cartography/vienna	41
14	Schematischer Aufbau eines adaptiven Datensatzes	45
15	Proportionale Vergrößerung eines Kartenausschnittes auf 200%	51
16	Proportionale Vergrößerung der Signaturmasse	51
17	Konstanthalten der Signaturmasse	51
18	Anpassung der Signaturmasse mit dem Wurzelgesetz	51
19	Der Graph der Funktion (5)	52
20	Der Graph der Funktion (6)	53
21	Der Graph der Funktion (9)	53
22	Der Graph der Funktion (10)	53
23	Der Ursprung eines Textes in SVG	55
24	Positionskorrektur von Punktsignaturen entlang der x-Achse	55
25	Wirkung der Größenänderung von Punktsignaturen	57
26	Wirkung der Größenänderung von Labels und die Raumaufteilung in Quadranten	57
27	Das Kontextmenü des Adobe SVG-Viewer 1.0	61
28	Schematische Darstellung der Skalierungsgrenzen	63
29	Der Datensatz s10	64
30	Der Datensatz s05	64
31	Provisorische Zoombereiche und Skalierungsgrenzwerte	65
32	Zusammenfassung von Objektgruppen einer Objektklasse	67
33	Provisorische Festlegung der Informationsgehalt-Grenzwerte	68
34	Sequenz (A ist die Sequenz. B und C sind atomare Komponenten dieser Sequenz)	73
35	Iteration (A ist die Iteration. B ist die atomare Komponente dieser Iteration)	73
36	Selektion (A ist die Selektion. B und C sind die Komponenten dieser Selektion)	74
37	Ausschnitt aus dem SVG-Dokument	74
38	Teilfunktion zur Entfernung überflüssiger text nodes	75
39	Teilfunktion zur Einblendung der richtigen Objekte	75

x	40	Die Funktion appendStyle	76
	41	Die Funktion removeVisibility	76
	42	Die Funktion zoomMap	77
	43	Die Funktion adaptSigns	79
	44	Die im Projekt vorkommenden Punktsignaturen	80
	45	Die Funktion signShift	80
	46	Die Funktion versalShift	80
	47	Die Funktion shiftGroup	80
	48	Die Funktion shiftIndividuals	81
	49	Die Funktion letterSize	81
	50	Festlegung von r_0 und t	83
	51	Überarbeitete Informationsgehalt-Grenzwerte	84
	52	Der Graph der Funktion für die Anpassung der Signaturmasse	88

III Tabellen

1	Vergleich häufig verwendeter Farbtiefen	6
2	Anforderungen an ein Vektorformat	14
3	Grafische Mindestgrößen für Papierkarten	20
4	Grafische Mindestgrößen für Bildschirmkarten mit <i>Antialiasing</i>	24
5	Objektgruppen, die in der Karte enthalten sein sollen	47
6	Die Beträge der Positionskorrektur für Kartenbeschriftung	58
7	Die vorhandenen Objektgruppen und ihre Merkmale (Erläuterungen im Text)	66
8	In den Ebenennamen verwendete Bezeichnungen und ihre Bedeutung	70
9	Die wichtigsten Stileigenschaften für Bildschirmkarten	71

1 Zielsetzung und Überblick

Bildschirmkarten erfreuen sich beim Publikum grosser Beliebtheit, sei dies auf CD-ROM oder im Internet, eingebunden in einen interaktiven Atlas oder auch als Teil einer anderen Multimediaanwendung.

Viele solcher Bildschirmkarten verfügen über ein begrenztes Mass an Interaktivität. Oft sind bestimmte Inhalte zu- oder abschaltbar, oder einzelne Elemente können animiert sein. Fast immer können Bildschirmkarten interaktiv im Massstab verändert – gezoomt – werden.

Das Zoomen von Bildschirmkarten ist aber gewissen Grenzen unterworfen: Bei Karten, die als Bitmap – also als eine Sammlung von Pixeln – vorliegen, werden die einzelnen Pixel mit jeder Zoomstufe grösser, bis die Darstellung ab einem bestimmten Massstab nicht mehr tragbar ist. Karten in Vektorform unterliegen zwar der durch die Auflösung bedingten Einschränkung nicht. Was die Generalisierung betrifft, verhalten sie sich aber nicht anders als Pixelkarten: Die Detailgenauigkeit wird durch die Massstabsvergrösserung in keiner Weise verbessert.

Mit der vorliegenden Arbeit soll eine Bildschirmkarte entstehen, deren Detailgenauigkeit durch interaktives Zoomen verändert werden kann: Der Benutzer wird durch ein- oder auszoomen per Knopfdruck die Menge der dargestellten Informationen erhöhen bzw. verkleinern können. Expertenwissen darf dabei nicht verlangt werden, die Karte soll unabhängig vom Wissensstand eines Benutzers bedienbar sein. Für jeden Massstab soll die Kartengrafik einwandfrei und bestmöglich lesbar sein, die Karte soll sich also adaptiv an den gewählten Massstab anpassen. Ausgabepattform wird aufgrund der zunehmenden Bedeutung des Internet ein gewöhnlicher Webbrowser sein.

Ziel dieser Arbeit

1.1 Inhaltsüberblick

Die Arbeit ist so ausgelegt, dass sämtliche Methoden und Konzepte, die entwickelt werden und hier zusammengestellt sind, auf andere, ähnlich ausgelegte Projekte ebenso angewandt werden können. Die Grundlagen für die adaptiv zoombare Karte werden daher weitgehend unabhängig vom Projekt entwickelt.

Grob gliedert sich die Entwicklung der adaptiv zoombaren Karte in zwei Teile. Der erste, eher allgemeine Teil beschäftigt sich mit kartografischen Fragestellungen und technischen Belangen. Es werden diverse grundsätzliche Fragen beantwortet und einiges Know-how aus der Literatur zusammengetragen. Obwohl allgemein gehalten, wird doch stets versucht, die zu behandelnden Themen im Hinblick auf interaktive Massstabsänderungen auszuwählen.

Der zweite, projektspezifische Teil wendet die Resultate aus dem ersten Teil an und erläutert die Entwicklung der adaptiven Karte.

1.1.1 Allgemeiner Teil

Zuerst soll untersucht werden, wie gut und mit welchen Einschränkungen der Bildschirm als Ausgabemedium für Karten geeignet ist. Fragen, die beantwortet werden, sind unter anderem:

- ◆ Wie wird Grafik auf dem Bildschirm dargestellt?
- ◆ Welche Möglichkeiten stehen zur Verfügung, um die Ausgabequalität von Grafik auf Bildschirmen zu verbessern?

Kapitel 2 und Kapitel 3

- ◆ Wie gut ist – rein technisch gesehen – die Bildschirmausgabe einer Karte im Vergleich mit der Papiausgabe?
- ◆ Welche Vorteile hat die Bildschirmausgabe von Karten?
- ◆ Welche Möglichkeiten gibt es, Karten auf Bildschirmen auszugeben?
- ◆ Warum ist das XML-basierte Vektorgrafik-Dateiformat SVG besonders gut für Bildschirmkarten geeignet?

Diese und ähnliche Fragen sollen in Kapitel 2 ‹Der Bildschirm als Ausgabe-medium für Grafik› ab Seite 5 und Kapitel 3 ‹Der Bildschirm als Ausgabe-medium für Karten› ab Seite 11 beantwortet werden.

Kapitel 4

Weiter wird von Interesse sein, wie Karten für das Ausgabemedium Bildschirm zu gestalten sind, damit das Medium und seine Möglichkeiten optimal genutzt werden können. Vor allem gibt es im Vergleich zur Herstellung von Papierkarten einige Unterschiede zu beachten, welche die technischen Einschränkungen von Bildschirmen berücksichtigen. Auch der im Vergleich zu Papier deutlich grössere Leseabstand ist ein Thema. Alle untersuchten Fragen, welche den Kartenentwurf und die Kartentechnik für das Ausgabe-medium Bildschirm behandeln, sind im Kapitel 4 ‹Gestaltung von Bildschirmkarten› ab Seite 17 zusammengestellt.

Kapitel 5

Ein Thema, das hier – trotz seiner Wichtigkeit – leider nur am Rand behandelt werden kann, da es den Rahmen dieser Arbeit sprengen würde, ist die Benutzerführung in multimedialen Projekten. Wer sich mit Interaktion beschäftigt, kommt nicht darum herum, eine grafische Oberfläche zu gestalten, die den Benutzern einen möglichst einfachen Zugang zu der Anwendung ermöglicht. Der Gestaltung von Benutzeroberflächen wird heute – vor allem im Internet – oft zu wenig Beachtung geschenkt. Einen kurzen Überblick zur Bedeutung von grafischen Benutzeroberflächen bietet das Kapitel 5 ‹Interaktion mit Bildschirmkarten› ab Seite 27.

Kapitel 6

Der Aufbau des sehr neuen Vektorformates SVG (*scalable vector graphics*) ist aus der technischen Perspektive von entscheidender Bedeutung. Um mit SVG umgehen zu können, muss man auch etwas über XML (*extensible markup language*) wissen. Das Kapitel 6 ‹Technische Grundlagen von SVG› ab Seite 31 gibt eine auf das Notwendigste reduzierte Übersicht zu SVG, XML und den verwandten Technologien.

1.1.2 Projektspezifischer Teil

Kapitel 7

Bevor die Karte gezeichnet und ein Steuerprogramm entwickelt werden kann, müssen die einzelnen Probleme, die durch die Massstabsveränderung auftauchen, auf konzeptioneller Ebene und unabhängig von der Implementation gelöst werden. Mitunter sollen die folgenden Fragen beantwortet werden:

- ◆ Wie muss eine Bildschirmkarte aufgebaut sein, damit sie sich an einen Massstab anpassen kann?
- ◆ Wie verhält sich die Informationsdichte einer Karte, wenn ihr Massstab verändert wird?
- ◆ Müssen Strichstärken, Punktsignaturgrössen und Schriftgrössen an den gewählten Massstab angepasst werden, und wenn ja, wie?

Das Kapitel 7 ‹Interaktive Massstabsveränderungen durch adaptives Zoomen› ab Seite 43 erläutert diese für adaptive Karten benötigten Grundlagen.

Kapitel 8

In Kapitel 8 ‹Entwicklungsumgebung und Ausgabeplattform› ab Seite 59 werden im Sinn einer Überleitung die für die Entwicklung verwendete Software und die Vor- und Nachteile der gewählten Ausgabeplattform geschildert.

Das Erstellen des kartografischen Datensatzes ist der nächste Schritt, mit dem sich die Arbeit auseinandersetzt. Der Datensatz muss so gestaltet sein, dass von aussen, mit Hilfe eines Steuerprogrammes, seine interaktive Änderung möglich ist. Dies verlangt von Anfang an eine konsequente und durchdachte Vorgehensweise im grafischen Aufbau der Karte, die in Kapitel 9 ‹Aufbau des adaptiven Datensatzes› ab Seite 63 beschrieben ist.

Kapitel 9

Das Programm, welches auf Benutzerinteraktion reagiert und somit die interaktive Karte steuert, ist Thema des letzten projektspezifischen Teils. Der logische Aufbau der Anwendung soll im Kapitel 10 ‹Das Steuerprogramm für die adaptive Karte› ab Seite 73 detailliert erklärt werden.

Kapitel 10

Die adaptive Karte muss im Anschluss analysiert und optimiert werden. Dies geschieht in Kapitel 11 ‹Feintuning› ab Seite 83.

Kapitel 11

Einen Überblick über alle wichtigen Resultate der Arbeit im Sinne einer Zusammenfassung und einer Kritik liefert abschliessend das Kapitel 12 ‹Resultate› ab Seite 85.

Kapitel 12

1.2 Konventionen in der Layoutgestaltung


Diese Arbeit wird auf drei Arten publiziert: Es gibt eine gedruckte Variante, eine PDF- und eine Online-Version. Sofern Sie die gedruckte oder die PDF-Version lesen, ist das Layout dieser Arbeit zweispaltig ausgelegt. Die bundseitige Spalte enthält den Haupttext. Die Randspalte lässt Raum für kleine Abbildungen, schmale Tabellen und ergänzende Texte.

1.2.1 Die Randspalte

Die Randspalte ist generell eine Hilfe für die Navigation im Text. Sie enthält bei besonders wichtigen Absätzen eine Kurzfassung des Inhaltes, eine sogenannte Marginalie.

Marginalien

Zu einigen Kapiteln findet man interaktive oder solche Abbildungen, die nur auf dem Bildschirm bewertet werden können, auf www.seven-r.ch/svg/. Solche Abbildungen sind an den betreffenden Stellen mit einer Marke gekennzeichnet.

Ein Verweis auf eine interaktive Abbildung: 

Zu bestimmten Themen, die nicht ausführlich diskutiert, sondern nur angeschnitten werden, sind Hinweise auf ausführlichere Beschreibungen ebenfalls in der Randspalte angegeben. Diese Hinweise können Links ins Internet oder Verweise auf das Literaturverzeichnis sein. Alle Links wurden zuletzt Anfang September 2000 auf ihre Gültigkeit geprüft.

www.seven-r.ch/svg/

Falls der Haupttext eine Definition enthält, ist dies in der Randspalte speziell gekennzeichnet.

Def. Definition

1.2.2 Konventionen in der Textauszeichnung

Kursiv hervorgehoben sind *technische Ausdrücke*, *Firmennamen* und *Produktenamen*. Code-Elemente werden in einer serifenlosen Schrift dargestellt (z. B. `<path d="..." />`). Wenn Sie dieses Dokument in einer elektronischen Form (als PDF oder online) lesen, dann sind Querverweise gleichzeitig Hyperlinks an die entsprechenden Stellen.

2 Der Bildschirm als Ausgabemedium für Grafik

Rasterbildschirme gewannen mit dem aufkommenden *Desktop Publishing* Mitte der Achtziger Jahre stark an Bedeutung. Von zentraler Wichtigkeit war – nicht nur in der Druckvorstufe – deren Funktion als Kontrollgerät für das zu erwartende Druckergebnis (Kiehn und Titzmann, 1998:20). WYSIWYG («*what you see is what you get*») wurde erst durch Rasterbildschirme ermöglicht.

Die Entwicklung der digitalen Medien, zuerst Multimedia-CD-ROMs, später das World Wide Web, gaben dem Bildschirm eine neue Bedeutung: Das bisherige Kontrollgerät für Papiausdrucke wandelte sich zu einem zentralen Ausgabemedium für Multimediaanwendungen (Kiehn und Titzmann, 1998:20).

Der Bildschirm als Ausgabemedium

2.1 Darstellung von Grafik auf dem Bildschirm

Grafik wird auf Rasterbildschirmen mit Hilfe von Pixeln dargestellt, die jeweils eine bestimmte Farbe annehmen können. Über die Qualität der Darstellung bestimmen vor allem die Auflösung und Methoden zur Darstellungsoptimierung.

2.1.1 Darstellung von Pixeln

Jedes auf einem Bildschirm dargestellte Element – ob Bild, Text oder Video – setzt sich immer aus einzelnen Punkten zusammen. Es liegt daher nahe, einzelne Elemente, die auf dem Bildschirm erscheinen, auch aus einzelnen Bildpunkten aufzubauen und zu speichern (Bitmapbilder).

Bitmapbilder

Bei Grafiken, die aus Linien und einfarbigen Flächen bestehen, gibt es eine speichersparende Alternative (Eberl und Jacobsen, 2000:186): Geometrische Objekte wie Linien, Rechtecke oder Kreise lassen sich sehr einfach in Vektorform beschreiben. Eine derartige Beschreibung reicht aus, um eine Grafik, die aus solchen Objekten besteht, zu speichern. Beispielsweise speichert man für eine Linie den Anfangs- und den Endpunkt sowie die Strichstärke. Die Anwendung berechnet aus diesen Daten die einzelnen Pixel, die dann dargestellt werden.

Vektorbilder

Auf dem Bildschirm sieht man den Unterschied, wie das Element gespeichert ist, meist nicht. Erst wenn Elemente, die aus Punkten bestehen, vergrößert oder verkleinert werden, kann man erkennen, dass es sich um eine Bitmapgrafik handelt. Vektorbilder hingegen können ohne sichtbare Veränderungen in der Qualität skaliert werden (Eberl und Jacobsen, 2000:186).

2.1.2 Darstellung von Farbe

Die Darstellung von Punkten auf einem Bildschirm funktioniert nach dem Grundprinzip der Braunschen Röhre: In der luftleeren Röhre wird ein Elektronenstrahl Zeile für Zeile auf den Leuchtschirm der Innenseite des Bildschirms gelenkt, auf dem sich eine Phosphorschicht befindet. Diese Schicht leuchtet auf, sobald sie vom Elektronenstrahl getroffen wird. Um den Elektronenstrahl auf jeden beliebigen Punkt des Bildschirms zu steuern, kann er durch ein senkrecht dazu erzeugtes elektrisches oder magnetisches Feld in seiner Richtung abgelenkt werden.

Funktionsweise der Braunschen Röhre

Anzahl Farben	Bit/px
2	1
4	2
16	4
256	8
<i>HiColor</i> (65 536)	16
<i>TrueColor</i> (mehr als 16 Mio)	24
<i>TrueColor</i> mit Alpha-Kanal	32

Tab. 1 Vergleich häufig verwendeter Farbtiefen

Auflösungsvermögen und Tripelabstände

Die Stärke des Strahles bzw. die Stärke des Elektronenstromes bestimmt die Helligkeit des Leuchtpunktes auf dem Bildschirm. Das heisst, ein Elektronenstrahl kann bei einem Monitor, auf dem 256 Graustufen dargestellt werden können, 256 Stärken annehmen (Kiehn und Titzmann, 1998:21).

Für die Farbdarstellung werden drei Elektronenstrahlen und drei verschiedene Phosphorarten auf dem Leuchtschirm benötigt. Auf der Innenseite des Bildschirms sind immer drei Leuchtpunkte (rot, grün, blau) so dicht nebeneinander angeordnet, dass das Auge sie nur als einen Punkt wahrnimmt. Eine solche Konstellation wird als «Tripel» bezeichnet. Diese drei Leuchtpunkte bilden zusammen einen Bildpunkt. Mittels der drei Grundfarben Rot, Grün und Blau der drei Leuchtpunkte kann ein grosses Spektrum an Farben dargestellt werden (Kiehn und Titzmann, 1998:21).

Abhängig von Tonwert und Farbwert einer darzustellenden Farbe werden die drei Leuchtpunkte unterschiedlich stark aktiviert. Eine volle Aktivierung aller drei Leuchtpunkte führt zu der Darstellung von Weiss auf dem Bildschirm (Kiehn und Titzmann, 1998:21). Um die Information zu speichern, welcher Anteil an der jeweiligen Farbe eingesetzt werden soll, wird eine bestimmte Anzahl Bits verwendet (siehe Tab. 1). Die Anzahl angezeigter Farben wird auch «Farbtiefe» genannt.

2.1.3 Auflösung

Die Dichte der Punkte, die ein Bild auf einem Bildschirm zusammensetzen können, wird als «Auflösung» bezeichnet. Meist wird die Auflösung in *dots per inch* (dpi) angegeben. Gleichzeitig heisst aber auch die Gesamtzahl an Pixeln (px) waagrecht und senkrecht auf dem Bildschirm Auflösung (Eberl und Jacobsen, 2000:187). Genau das Gleiche gilt für Bilder, die auf dem Bildschirm dargestellt werden; auch sie haben eine in Pixel gemessene Breite und Höhe und eine Auflösung in dpi.

Das Auflösungsvermögen eines Farbbildschirms wird vom Abstand eines Tripelpunktes zum nächsten bestimmt. Dabei sollten 0,31 mm nicht überschritten werden, da sonst die Abbildungsschärfe bei Linien oder feinen Punkten abnimmt (Kiehn und Titzmann, 1998:21). Hochwertige Monitore haben Tripelabstände von weit unter 0,28 mm. Rechnet man nun diese Abstandswerte auf die übliche Einheit für die Auflösung dpi um, so erhält man je nach Qualität der Bildschirme auf Röhrenmonitoren Werte zwischen 72 und 100 dpi (Kiehn und Titzmann, 1998:22), auf TFT-Displays sogar deutlich über 100 dpi.

Die Firma *Apple Computer, Inc.* empfiehlt gemäss der Firma *VillageTronic* (VillageTronic, 1998:34) eine Bildschirmauflösung von 72 dpi, was einem Tripelabstand von 0,35 mm entspricht. Diese Empfehlung stammt aus den Anfangszeiten des *Desktop Publishing* und ist Teil einer Vereinbarung unter Software-Herstellern im Prepress-Bereich. Fast alle Programme setzen eine Auflösung von 72 dpi voraus, um eine 1:1-Darstellung auf Bildschirmen zu gewährleisten. Heute bieten aber alle Monitore kleinere Tripelabstände (in der Regel unter 0,28 mm) und somit höhere Auflösungen. Üblicherweise werden zum Beispiel auf einem Monitor mit 17 Zoll Bildschirmdiagonale in der empfohlenen Auflösung von 1024 * 768 px ca. 80 dpi dargestellt. Um eine 1:1-Wiedergabe auf dem Bildschirm zu erreichen, muss also ein Vergrößerungsfaktor von ca. 110% eingestellt werden.

Trotz der Verbesserung der Darstellungsqualität von Bildschirmen empfehlen Lehrbücher für Bildschirmgrafik weiterhin, für Grafiken eine Auflösung von 72 dpi zu verwenden.

2.1.4 Grafische Mindestgrößen

Die feinste darstellbare Linie hat auf dem Bildschirm eine Strichstärke von einem Pixel. Geht man von der Bildschirmauflösung von 72 dpi aus, so erhält man für die Abmessungen eines Pixels $\frac{1}{72}$ Inch. Dies entspricht 0,3528 mm (= 1 *point*). Feinere Linien können nur mit Hilfe von *Antialiasing* (vgl. Kapitel 2.1.5) dargestellt werden (Kiehn und Titzmann, 1998:82). Eine ausführliche Auseinandersetzung mit grafischen Mindestgrößen folgt in Kapitel 4.4 «Bildschirmspezifische Modifizierungen der Grundsätze zur Kartengestaltung» ab Seite 22).

2.1.5 Antialiasing

Das Raster-Ausgabeverfahren von gängigen Bildschirmen mit Auflösungen von unter 100 dpi kann bewirken, dass Bildelemente, die weder horizontal noch vertikal liegen, gezackte Kanten erhalten. Besonders bei Grafiken mit harten Kontrasten oder bei der Darstellung von Schrift ergeben sich an den Rändern oder Kanten von Linien unschöne Verfremdungseffekte. Eine Linie wird möglicherweise als treppenförmiges Zick-Zack-Muster wahrgenommen, oder es entstehen Überlagerungseffekte durch nahe beieinander verlaufende Kurvenstücke. Dies bezeichnet man als «*Aliasing*» (Bungartz et al., 1996:36).

Um diesem Effekt entgegenzuwirken, ohne die Auflösung zu erhöhen, bedient man sich des sogenannten «*Antialiasing*». Synonym verwendet man im Deutschen die Begriffe «geglättete Darstellung», «Glättung» und «Kantenglättung». Mit *Antialiasing* verschwinden Randpixel in Phasen, die ausgehend von der Ursprungsfarbe über verschiedene Zwischentöne in die benachbarte Farbe übergehen (siehe Abb. 1). Der in Darstellungen ohne *Antialiasing* entstehende Effekt der gezackten Linien kann so vermieden werden, Linien werden viel weicher und erscheinen optisch schöner (Real3D, 1998).

Der richtige Eindruck entsteht allerdings erst ab einem gewissen Betrachtungsabstand. Ausserdem ist der Einsatz dieser Technik nicht unproblematisch. *Antialiasing* ist sehr speicheraufwendig und verlangsamt vor allem ältere Computersysteme stark. Für das *Antialiasing* werden überdies zusätzliche Graustufen und Farben benötigt (Eberl und Jacobsen, 2000:191).

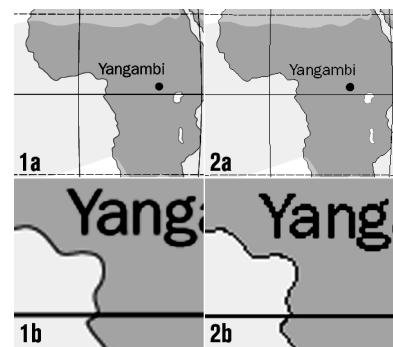


Abb. 1 Ein Kartenausschnitt mit (1a und 1b) und ohne (2a und 2b) *Antialiasing*

2.2 Wahrnehmung von Grafik auf dem Bildschirm

Eine wichtige Einflussgrösse auf die Art der Bildschirmausgabe ist die Wahrnehmungsfähigkeit des menschlichen Auges für Bildschirmgrafik. Sie bildet zusammen mit dem Bildschirm ein Wahrnehmungssystem aus Sender und Empfänger. Die Funktionsmöglichkeiten dieses Systems sind begrenzt und müssen daher beim Design von Bildschirmapplikationen berücksichtigt werden (Kiehn und Titzmann, 1998:20).

2.2.1 Ergonomische Aspekte

Das menschliche Auge nimmt Strukturen bis zu einer als Auflösungsvermögen bezeichneten Feinheit wahr. Dieses Auflösungsvermögen ist stark abhängig vom Leseabstand. Zudem kann das Auge Helligkeiten innerhalb eines begrenzten Umfangs erkennen und Farben unterscheiden, die voneinander um eine minimale Nuance differieren. Beim Sehen kommt es

an einigen Stellen des Sichtfeldes immer wieder zu mehr oder weniger deutlichen Effekten bzw. Störungen (Kiehn und Titzmann, 1998:24).

Seine maximale Auflösung erreicht das menschliche Auge dann, wenn ein Objekt so nahe am Auge ist, dass es gerade noch fokussiert werden kann. Im Normalfall beträgt dieser Abstand – der Leseabstand – etwa 25 cm, er ist aber individuell unterschiedlich. Bei einer Winkelauflösung des Auges von ca. $\frac{1}{60}^\circ$ können dann zwei Objekte im Abstand von 73 μm , beispielsweise zwei Linien, gerade noch als getrennt wahrgenommen werden (Popa, 1997). Dies entspricht einer Auflösung von etwa 348 dpi. Höhere Auflösungen bewirken bei normalem Leseabstand nicht mehr Verbesserungen im Detailreichtum, sondern haben vor allem Einfluss auf die Qualität der Darstellung von Graustufen und Farben. Gleichzeitig lassen höhere Auflösungen aber die Betrachtung aus geringerer Distanz (kleinerer Leseabstand) oder mit der Lupe zu.

Der Leseabstand zum Bildschirm variiert gemäss Kiehn (Kiehn und Titzmann, 1998:25) je nach Bildschirmgröße und körperlicher Konstitution des Betrachters zwischen 40 und 80 cm. Der Richtwert für einen geeigneten Abstand zum Bildschirm beträgt 60 bis 80 cm. Ausgehend von 60 cm Leseabstand nimmt das Auge Haarlinien, die einen Abstand von 0,175 mm haben, als getrennt wahr (Kiehn und Titzmann, 1998:25). Dies entspricht in etwa einer Auflösung von 145 dpi. Die Bildschirmauflösung beträgt aber nur zwischen 72 und 100 dpi; es fällt dem Leser am Bildschirm also relativ leicht, einzelne Bildpunkte zu erkennen.

2.2.2 Wahrnehmung von Farbe

Für Bildschirmdarstellungen sind Farben von Bedeutung. Einerseits verursacht der Einsatz von Farbe auf dem Bildschirm, ganz anders als auf Papier, keine Mehrkosten. Andererseits müssen die Sinne der Benutzerin statt über den Tastsinn, bei Papier von nicht zu unterschätzender Bedeutung, über optische Reize angesprochen werden (Kiehn und Titzmann, 1998:91).

Farbwahrnehmungen erzeugen beim Betrachter bestimmte Reaktionen. Vegetative Reize, die durch die Farbwahrnehmung im Auge erzeugt werden, beeinflussen das Nervensystem und die Hormonsteuerung bei allen Menschen auf die gleiche Weise (Kiehn und Titzmann, 1998:87). Dies bedeutet, dass man durch den gezielten Einsatz von Farben die Assoziationen der Betrachter steuern kann.

2.2.3 Adaptionsfähigkeit des menschlichen Auges

Das Auge ist in der Lage, sich in seiner Empfindlichkeit an Änderungen der Leuchtdichteverhältnisse, z. B. zwischen Bildschirm und Umgebung, zu gewöhnen. Einige Erscheinungen der subjektiven Wahrnehmung von Kontrast sind für Bildschirmgrafik von Bedeutung und haben ihre Ursache im Adaptionsverhalten des menschlichen Auges.

Die *direkte Adaption* wird von der Helligkeit des direkt betrachteten Teils bestimmt. Sie hat eine Einstellzeit von ca. zehn Sekunden und wirkt sich als Verringerung der subjektiven Kontrastempfindung aus. Die *laterale Adaption* läuft innerhalb von einer Sekunde ab. Sie wird von der Helligkeit des Umfeldes des direkt betrachteten Fixierpunktes bestimmt. Durch sie kommt es zu den Effekten des Simultankontrastes (Kiehn und Titzmann, 1998:27).

Simultankontrast tritt dann auf, wenn eine unbunte, neutrale Fläche auf einer farbigen Fläche, etwa Rot, Gelb, Grün oder Blau, liegt. Da das visuelle System des Menschen bei der Betrachtung von Farben jeweils auch die Gegenfarbe fordert, nimmt die unbunte Fläche einen Ton der jeweiligen Kom-

plementärfarbe des betreffenden bunten Farbtons an (Kiehn und Titzmann, 1998:90). Diese Adaptionseffekte sind für Bildschirmgrafik entscheidend; sie kommen bei der Festlegung z. B. der Farben von direkt benachbarten Flächen zum Tragen.

2.3 Ausgabe von Grafik auf dem Bildschirm

Multimediaanwendungen können Inhalte auf zwei sich grundsätzlich unterscheidende Arten auf einem Bildschirm ausgeben. Einerseits übernimmt die Grafikdarstellung ein Webbrowser. Andererseits – dieses Verfahren benutzen in der Regel Multimedia-CD-ROMs – geschieht die Ausgabe durch die *Runtime* eines Autoren- oder Animationstools wie *Macromedia Director* oder *Tribeworks iShell*.

Welche der erwähnten Methoden für ein Produkt geeigneter ist, hängt von diversen Faktoren wie z. B. Zielpublikum ab.

2.3.1 Ausgabe von Grafik in Webbrowsern

Die Verwendung von Webbrowsern als Ausgabeplattform für Multimediaprojekte hat den grossen Vorteil, dass die Ausgabe auf einer standardisierten Software unabhängig vom verwendeten Betriebssystem geschieht. Die Datenlieferung kann sowohl via Internet als auch ab einer CD-ROM oder in einer Kombination beider Möglichkeiten erfolgen.

Nachteilig ist, dass die Ausgabe auf ein oder mehrere Fenster begrenzt ist; Fensterlayout wie auch Pulldown-Menüs sind vorgegeben und können durch die Autorin einer Multimediaanwendung bei den heute erhältlichen Webbrowsern nicht oder nur wenig beeinflusst werden. Die ganze Bildschirmfläche kann nie ausgenutzt werden. Benutzer können Fenster jederzeit in der Grösse verändern, was in der Anwendungsentwicklung durch Ermöglichen dynamischer Anpassungen des Layouts an die Fenstergrösse berücksichtigt werden muss.

Alle Webbrowser unterstützen neben HTML-Text die Ausgabe von Bitmapgrafik in den Formaten GIF (gut geeignet für Grafiken mit vorwiegend homogenen Flächen) und JPEG (unterstützt 24-Bit-Farbe und ist daher gut geeignet für die Ausgabe von Fotos). Das Format PNG (verlustfreie Kompressionsmethode, vereint die Vorteile von GIF und JPEG) wird zur Zeit noch nicht von allen Browsern unterstützt (Weinman, 1998:38–51). Vektorgrafik und alle anderen Formen von Multimedia (z. B. interaktive Filme) können nur mit Hilfe von Plug-Ins ausgegeben werden.

Als «major player» im Markt der Webbrowser können heute der *Internet Explorer* von *Microsoft* und der *Netscape Communicator* angesehen werden, die fast den ganzen Markt unter sich aufgeteilt haben.

2.3.2 Ausgabe von Grafik in Autorentools

Autorentools erlauben eine viel flexiblere Gestaltung von Oberflächen, ohne weiteres kann auch der ganze Bildschirm ausgenutzt werden. Pulldown-Menüs können selber gestaltet, die Grösse von Fenstern von der Autorin vorgegeben werden.

Der Benutzer bekommt von einer solchen Multimediaanwendung nur eine *Runtime* zu sehen, also ein Programm, das in einem proprietären Format durch das Autorentool geschrieben wurde und unabhängig vom Autorentool läuft. Eine solche *Runtime* ist normalerweise auf ein bestimmtes Betriebssystem zugeschnitten, die Multimediaanwendung muss also pro Betriebssystem einmal produziert werden.

Standardisierte Form der Grafikausgabe

Hohe Flexibilität in einem proprietären Format

Runtimes können nicht über Internet verbreitet werden, allerhöchstens deren Inhalt. Vertrieb via CD-ROM ist die einzige Möglichkeit, eine runtime-basierte Multimediaanwendung an Kunden zu liefern.

Mit Autorentools ist es aber möglich, Anwendungen in Form von flächenhaft klein dimensionierten, runtimeartigen Elementen zu produzieren, die mit Hilfe eines Plug-Ins in einem Webbrowser platziert werden können. In eine Webseite integrierte interaktive Filme sind ein Beispiel.

Die meisten Autorentools unterstützen mindestens die folgenden Möglichkeiten, Inhalt auszugeben:

- Bitmapgrafik (Formate BMP, GIF, JPEG, PSD, TIFF, PICT, usw.)
- einfache Vektorformen
- komplexe Vektorgrafik im *Flash*-Format
- interaktive *Quicktime*-Filme

Am meisten im Markt etabliert hat sich heute das Programm *Macromedia Director*, das zur Zeit in der Version 8 vorliegt. Ebenfalls stark verbreitet ist die Software *Macromedia Flash*, die sich gut für die Produktion von Inhalten für Webbrowser mit dem *Flash*-Plug-In eignet. Neu auf dem Markt ist das vielversprechende *Tribeworks iShell*, das in direkte Konkurrenz zu *Director* zu treten versucht.

3 Der Bildschirm als Ausgabemedium für Karten

In Kapitel 2 «Der Bildschirm als Ausgabemedium für Grafik» wurde bereits auf die Art der Ausgabe von Grafik auf Bildschirmen eingegangen. In diesem Kapitel wird nun untersucht, in welcher Form *Karten* auf Bildschirmen ausgegeben werden können.

3.1 Wichtigkeit der dynamischen Struktur von Karten

Bildschirmkarten ergänzen (und ersetzen teilweise auch) die traditionellen Papierkarten. Dennoch ist der Einfluss der Papierkarten auf das Design von Bildschirmkarten sehr gross. So sind viele der heute erhältlichen Bildschirmkarten – ob auf CD-ROM oder World Wide Web – nur gescannte Versionen von Papierkarten, auf denen mit unhandlichen Symbolen interaktive Elemente signalisiert werden (Arleth, 1999:849). Das grosse Potential des Ausgabemediums Bildschirm wird so bei weitem nicht genutzt.

Interaktive Medien sind heute weit verbreitet. Durch den Einfluss des omnipräsenten World Wide Web erwarten Benutzer auf dem Bildschirm fast generell eine vernetzte Struktur (Cartwright et al., 1999:602). Eine Anzeige, die statisch ist, ist uninteressant, und dies gilt auch bei Bildschirmkarten. Längeres Betrachten auf dem Bildschirm ist eher die Ausnahme (Spiess, 1996a:470): Bewegung und Aktion ist die Regel. Ein Ausschnitt wird meistens nur kurz durchgemustert und dann nach *Hotspots* abgesucht, die neue Operationen auslösen oder Veränderungen hervorrufen. Die Benutzerin will quasi «in die Karte hineingehen», sowohl räumlich, wie auch konzeptionell. Sie will tiefere Ebenen erforschen. Sie will Informationsstücke selber zusammensetzen. Interaktion ist gemäss Cartwright der Schlüssel zur Wissensbildung (Cartwright et al., 1999:602).

Von zentraler Bedeutung für den Autoren von Bildschirmkarten muss das Medium Bildschirm sein (Arleth, 1999:849). Die Schwächen der Bildschirmausgabe von Karten, allen voran die im Vergleich zum Druck schlechte Darstellungsqualität, können durch zahlreiche Stärken wie Interaktion und Animation mehr als wett gemacht werden (Brown, 1993:135). Es gilt dabei auch zu beachten, dass die Herstellung einer Bildschirmkarte nicht nur bedeutet, die Karte zu produzieren (Arleth, 1999:850). Genau so wichtig sind eine einfach zu lernende, konsistente Benutzeroberfläche und relevante Werkzeuge, welche die Interaktionen steuern.

Vernetzte und dynamische Karten

Nutzen der Stärken von Bildschirmen

3.2 Unterschiede zwischen Papierkarten und Bildschirmkarten

Papier als materieller Zeichenträger weist im Vergleich zum nicht-materiellen Ausgabemedium Bildschirm einige wesentliche Unterschiede auf. Abweichungen findet man in Ausgabequalität, Konsistenz der Ausgabe, Handlichkeit, Benutzerpotenzial und Entwicklungspotenzial, vor allem aber in Interaktivität und Nachführung.

3.2.1 Ausgabequalität und grafische Mindestgrössen

Papierkarten profitieren in der Regel von einer sehr guten Druckqualität. Mehr als 1000 dpi sind Standard, 2000 dpi und darüber sind in der

*Sehr unterschiedliche grafische
Mindestgrössen*

*Keine Kontrolle über das
Ausgabemedium des Benutzers*

Kartografie üblich. Bildschirmgrafik wird dagegen mit 72 bis 100 dpi ausgegeben (vgl. Kapitel 2.1.3 «Auflösung» auf Seite 6).

Bedingt durch die hohe Auflösung ist es bei gedruckter Ausgabe auch möglich, sehr viel kleinere Dimensionen darzustellen als auf Bildschirmen. In Kapitel 2.1.4 «Grafische Mindestgrössen» auf Seite 7 wurde bereits gezeigt, dass die feinste auf dem Bildschirm darstellbare Linie ohne *Anti-aliasing* eine Strichstärke von 0,35 mm hat. Dieser Wert ist sieben mal grösser als die 0,05 mm, die im Lehrbuch «Kartographie» (Hake und Grünreich, 1994:93) für die Papierausgabe vorgeschlagen werden (vgl. auch Kapitel 4.2.6 «Grafische Mindestgrössen» auf Seite 19).

3.2.2 Konsistenz in der Ausgabe

Die Kartenautorin kann bei einer Papierkarte sehr genau kontrollieren, was der Benutzer sieht. Die Autorin kann selbst bestimmen und auch prüfen, wie gut die Ausgabequalität der Karte ist. Bei Bildschirmkarten entzieht sich das Ausgabemedium der Kontrolle der Kartenautorin. Die Ausgabequalität auf Bildschirmen ist stark abhängig von verschiedenen Faktoren:

- ♦ *Computerkonfiguration:* Die Kartenautorin kann zwar für interaktive Karten eine bestimmte Computerkonfiguration als Minimalanforderung verlangen, sie weiss aber nie, wie die Konfiguration des Benutzers genau aussieht. Beispielsweise können Animationen daher oft mit anderen Geschwindigkeiten abgespielt und daher vom Benutzer anders empfunden werden, als dies die Kartenautorin vorgesehen hat.
- ♦ *Leistung der Grafikkarte:* Je nach Einstellungen und Qualität der Grafikkarte kann ein Bildschirm unterschiedlich viele Farben anzeigen (Farbtiefe). Die gängigen Farbtiefen sind in Tab. 1 auf Seite 6 zusammengefasst.
- ♦ *Bildschirmgrösse, Bildschirmauflösung:* Die Kartenautorin weiss nicht, wie gross der Bildschirm des Benutzers ist, und welche Bildschirmauflösung eingestellt ist. Z. B. auf 17 Zoll-Monitoren ist eine Auflösung von 1024 * 768 px üblich, viele Benutzer arbeiten aber auch mit der Auflösung 832 * 624 px.
- ♦ *Betriebssystem:* Bei der digitalen Bildverarbeitung gibt der *Gammawert* das Verhältnis zwischen Eingabe- und Ausgabekontrast eines Bildes an. Der Standard-Gammawert variiert bei unterschiedlichen Betriebssystemen ziemlich stark; z. B. das Mac OS hat einen Gammawert von 1,8, SGI 1,7, Windows 2,5 (Ballhaus, 1998:54). Zudem kann der Benutzer den Gammawert auch manuell verändern.
- ♦ *Benutzereinstellungen:* Der Benutzer kann bei seinem Bildschirm Helligkeit und Kontrast beliebig wählen.

3.2.3 Handlichkeit

Papierkarten sind sehr viel handlicher als Bildschirmkarten. Sie eignen sich z. B. für Aktivitäten im Gelände hervorragend. Eine Bildschirmkarte kann niemals den gleich guten Überblick verschaffen wie eine ausgebreitete Papierkarte (Spiess, 1996a:469). Auch mangelnde Resistenz gegen Umwelteinflüsse wie Lichteinfall auf den Bildschirm lassen eine Bildschirmkarte im Gelände schnell unbrauchbar erscheinen.

3.2.4 Benutzerpotenzial

Eine Papierkarte kann im Prinzip jeder verwenden. Für Bildschirmkarten muss die Benutzerin aber eine ganze Reihe von Voraussetzungen erfüllen. Unter anderem braucht sie Zugang zu einem einigermaßen leistungsfähigen Computer und muss zumindest Grundkenntnisse in der Bedienung von Computern haben.

3.2.5 Entwicklungspotenzial

Auf Papier ist eine hohe Entwicklungsstufe in der Ausgabequalität erreicht, technische Fortschritte sind zwar noch möglich, aber nicht mehr sehr gross. Anders sieht es bei Bildschirmen aus: Technologien werden erneuert oder auch abgelöst. Die Darstellungsqualität wird ständig verbessert.

3.2.6 Interaktivität

Papier als materieller Datenträger ist ein sehr statisches Wiedergabemedium. Eine einmal gedruckte Karte lässt sich nicht mehr verändern und zeigt somit immer die gleichen Informationen. Der Betrachter wird bestenfalls bei der Produktion der Karte miteinbezogen.

Ganz anders sieht die Situation bei Bildschirmkarten aus: Eine Bildschirmkarte ist die grafische Repräsentation von digitalen Kartendaten. Eine Bildschirmkarte ist somit nur eine von mehreren möglichen Repräsentationen der zugrundeliegenden Daten (Arleth, 1999:850).

Durch geschickten Einbezug des Benutzers kann eine Karte optimal an dessen Vorkenntnisse oder Interessen angepasst werden. Die Karte wird leichter zu lesen. Einzelne Themen lassen sich einfach und schnell aus-, und später wieder einblenden. Ausführlichere Informationen zum Thema Interaktivität findet man in Kapitel 5 «Interaktion mit Bildschirmkarten» ab Seite 27.

Anpassung einer Karte an die Vorkenntnisse und Interessen eines Benutzers

3.2.7 Nachführung

Nachgeführte Karten müssen von Kartenbenutzern neu gekauft werden. Aus Benutzersicht werden Papierkarten also ersetzt und nicht aktualisiert.

Einen ganz anderen Weg kann man mit Bildschirmkarten gehen. Schnelle und kostengünstige Update-Möglichkeiten sind beispielsweise via Internet denkbar. Wenn die Kartenautorin diese Möglichkeiten nutzt, kann dies zu einem sehr viel höheren Aktualitätsgrad führen, als dies bei Papierkarten der Fall ist.

Schnelle und kostengünstige Update-Möglichkeiten

3.3 Eine geeignete Entwicklungsumgebung für Bildschirmkarten

Es gibt verschiedene Umgebungen, die es ermöglichen, eine Karte für den Bildschirm zu entwickeln. Alle haben ihre Qualitäten, aber auch diverse Nachteile. Da keine Entwicklungsumgebung vorbehaltlos geeignet ist, geht es darum, diejenige zu finden, die am wenigsten Nachteile hat.

3.3.1 Unterschiedliche Dateiformate

Aus der technischen Perspektive stehen dem Kartenauteuren zwei Möglichkeiten offen, eine Karte auf einem Rasterbildschirm auszugeben, nämlich in Raster- oder in Vektorform. Bereits in Kapitel 2.1.1 «Darstellung von Pixeln» auf Seite 5 wurde kurz auf den Unterschied zwischen rasterbasierter und vektorbasierter Datenausgabe auf Bildschirmen eingegangen.

Rasterformate

Grob können alle Formate entweder der Gruppe der *Rasterformate* oder der Gruppe der *Vektorformate* zugeteilt werden. In der Regel unterstützen Rasterformate nur die simple Speicherung von Grafik, ausnahmsweise Transparenz (meist gespeichert im Alphakanal), selten Animation (Animated GIF). Durch die starre Struktur von Rasterformaten wird Interaktion praktisch verunmöglicht. Beispiele für Rasterformate sind GIF, JPEG, PNG, TIFF, BMP.

Dennoch werden zur Zeit die meisten Karten in Rasterform auf Bildschirmen dargestellt. Der Grund dafür ist, dass oft – der Einfachheit halber, aus Zeit- und Kostengründen oder mangels besserer Möglichkeiten – auf Papier vorliegende Karten gescannt und die meist nur unzureichend für die Monitorausgabe überarbeiteten Resultate auf den Bildschirm gebracht werden.

Vektorformate

Die Vorteile der Ausgabe von Karten in Vektorform sind zahlreich; kleine Datenmengen, hohe Flexibilität, Qualitätssteigerung durch *Antialiasing* und Skalierbarkeit zählen zu den wichtigsten. Zur exakten Beschreibung einer Fläche auf der Grundlage des Vektormodells reichen wenige Koordinatenpaare aus, während die Abbildung der gleichen Fläche durch das Rastermodell die Speicherung jeden einzelnen Punktes der Fläche fordern würde (Dickmann und Zehner, 1999:67).

Vektorformate bieten normalerweise neben der reinen Speicherung der Geometrie sehr viel mehr Funktionalität. Oft ist *Antialiasing* möglich (vgl. Kapitel 2.1.5 «Antialiasing» auf Seite 7). Zooming, Ebenenhandling, (benutzergesteuerte) Bewegung von Objekten und Animation sind je nach Vektorformat Arten der Interaktion, die unterstützt werden. Beispiele für gängige Vektorformate sind DWF, DWG, Flash SWF, PGML, SVG (Winter und Neumann, 2000), aber auch EPS, PDF, PICT.

3.3.2 Kartografische Anforderungen an ein Dateiformat

Für Bildschirmkarten sind Vektorformate offenkundig besser geeignet als Rasterformate, obwohl – wie schon erwähnt – der Bearbeitungsaufwand deutlich grösser ist. Verschiedene Vektorformate haben aber sehr unterschiedliche Eigenschaften. Vor allem im Grad der möglichen Interaktivität unterscheiden sie sich stark.

Um alle wichtigen Vorteile von Bildschirmkarten nutzen zu können, muss ein Vektorformat möglichst viele der folgenden Anforderungen erfüllen:

Anforderung	Beispiel
Einfaches Erstellen der Datei	Export aus Grafikprogrammen
Einfache, möglichst plattformunabhängige Wiedergabe	Webbrowser
Kontrolle über die Objektstile und die Beschriftung während der Kartenherstellung	Linienstile (Strichstärke, Linienfarbe, <i>Antialiasing</i>) und Flächenstile (Flächenfarbe, Flächenkontur, Konturfarbe, Transparenz) beliebig wählbar

Tab. 2 Anforderungen an ein Vektorformat

Anforderung	Beispiel
Kontrolle über die Objektstile und die Beschriftung nach Fertigstellung der Karte	Spätere Korrekturen und Nachführung
Bewegen von Objekten	Setzen von persönlichen Marken
Animation	Darstellungen von zeitabhängigen Veränderungen
Zoom	Detailansichten
Ebenen	Vereinfachung des Kartenbildes
Hyperlink-Navigation	Verknüpfung mit externen Informationen
Externe Skripts zur Steuerung von: <ul style="list-style-type: none"> • Objektstilen • Inhalten • Animation • Transparenz • Beschriftung • allen Interaktionsmöglichkeiten 	Steuerelemente für den Benutzer
Wiedergabe von Rasterbildern	Reliefs
Transparenz	Reliefs
Kopierschutz	Copyrightschutz
Ausgabemöglichkeit auf Drucker	

Tab. 2 Anforderungen an ein Vektorformat (Forts.)

3.3.3 SVG als Dateiformat für Bildschirmkarten

Auf der Basis der obigen Ausführungen liegt es nahe, für die Kartenausgabe auf Bildschirmen ein Vektorformat zu wählen. Keines der heute verbreiteten Vektorformate erfüllt aber alle aufgelisteten Anforderungen.

Dem Katalog am nächsten kommt das neue Vektorformat SVG (*scalable vector graphics*), das sich im Jahr 2000 noch ganz am Anfang seines Lebenszyklus befindet. Zur Zeit liegen die Spezifikationen von SVG erst als *candidate recommendation* vor. Dieses Format hat infolge der vielen grossen Vorteile aber zumindest gute Chancen, sich in den nächsten Jahren als Standard durchzusetzen (Adobe, 2000a:2), nicht zuletzt auch deshalb, weil dem Entwicklungsteam Mitarbeiter von allen grossen Softwareherstellern angehören. Mit dabei sind neben *Adobe*, die eine Führungsrolle übernommen hat, auch *Apple*, *Autodesk*, *IBM*, *Macromedia*, *Microsoft*, *AOL/Netscape*, *Sun*, *Xerox* und andere.

SVG ist eine Sprache, mit der zweidimensionale Grafiken unter Verwendung von XML (*extensible markup language*) und mit Hilfe von CSS (*cascading style sheets*) oder XSL (*extensible stylesheet language*) beschrieben werden. XML ist eine Empfehlung des *World Wide Web-Consortium* (W3C) für strukturierten Informationsaustausch. Auf dieser Basis ist es möglich, dass SVG-Grafik nicht nur in einem reinen SVG-Viewer in der Umgebung eines Webbrowsers ausgegeben, sondern auch innerhalb von XML-Anwendungen benutzt werden kann.

SVG erlaubt drei Typen von Grafikobjekten: Formen, Bilder und Text. Grafische Objekte können gestaltet, gruppiert, verschoben, gedreht, skaliert, oder zu anderen Objekten zusammengesetzt werden. Weiter sind unter anderem Transformationen, Beschneidungspfade, Alphakanäle, Filtereffekte und Vorlagenobjekte möglich. SVG-Grafiken können interaktiv und dynamisch sein (Bowler et al., 2000:15–23).

Animationen können entweder anhand von SVG-Animationselementen oder mit Hilfe von externen Skripten definiert und ausgelöst werden. Die

Elementare Grafiktypen von SVG

Schnittstelle von SVG nach aussen ist dabei das DOM (*document object model*). Es stellt einen kompletten Zugang zu allen Objekten, Stilen und Eigenschaften einer SVG-Grafik zur Verfügung (Bowler et al., 2000:15–23).

Für kartografische Anwendungen sind die folgenden Vorteile von SVG entscheidend:

- ◆ Einfaches Erstellen durch Export aus Grafikprogrammen wie *Illustrator*, *Freehand* oder *CorelDraw*
- ◆ Datei im Textformat: einfaches und direktes Editieren in jeder Textverarbeitung
- ◆ Echte Vektoren und Bézierkurven
- ◆ Vollständige Kontrolle über Ebenen
- ◆ Gute Bildschirmausgabe durch *Antialiasing*
- ◆ Einbettung von ganzen Schriften (Postscript Type 1-Fonts)
- ◆ Verwendung von Stilvorlagen (CSS), sowohl intern als auch extern
- ◆ Darstellung von Pixelbildern (JPEG, GIF, PNG)
- ◆ Transparenz
- ◆ ICC-Farben zur plattformübergreifenden, systemunabhängigen Farbdarstellung
- ◆ Interaktive Steuerung durch dynamische Funktionen:
 - Geometrische Transformationen
 - Zeitabhängige Animation durch Elemente des SVG-Standards
 - Animation durch DOM-Schnittstelle
 - Benutzerinteraktion durch grafische Eingabegeräte (z. B. Maus) und Tastatureingaben
 - Dynamisches Einfügen von neuen SVG-Elementen
- ◆ Ausdruck auf Papier in Vektorqualität

Nachteile von SVG

Die Nachteile von SVG sind gering. Der wichtigste ist, dass das Format noch sehr jung ist und sich daher bisher nur in Webbrowsern und mit einem entsprechenden Plug-In betrachten lässt. Neue Webbrowser werden aber SVG ohne Plug-In unterstützen, da es dem XML-Modell entspricht, welche von künftigen Browsern interpretiert werden kann. Auch browserunabhängiger Einsatz von SVG wird dereinst auf der Basis von XML möglich sein. Fast alle grossen Softwarehersteller haben SVG- und/oder XML-Unterstützung in künftigen Produkten angekündigt. Sowohl mit Platzierungsmöglichkeiten in Datenbanken und Autorentools als auch mit Import- und Exportfiltern für Grafikprogramme wird man in den nächsten Monaten und Jahren rechnen können.

SVG, XML, DOM und die zugrundeliegenden Konzepte werden ausführlich im Kapitel 6 «Technische Grundlagen von SVG» auf Seite 31 beschrieben.

4 Gestaltung von Bildschirmkarten

Gute Kartengrafik ist eine der wichtigsten Voraussetzungen für eine attraktive Karte auf Papier wie auf dem Bildschirm. Feine Auflösung, differenzierte Signaturen, harmonische Farbgebung, gut lesbare und bestmöglich platzierte Schriften, optimale Bilddichte und ein guter Bildaufbau machen eine grafisch attraktive Karte aus (Spiess, 1996a:468).

In früheren Kapiteln wurde bereits gezeigt, dass sich Bildschirmkarten und Papierkarten in vielen Dingen stark unterscheiden. Vor allem auf die enorme Tragweite der Interaktion mit Bildschirmkarten soll hier nochmals hingewiesen werden. Unabhängig davon, in welchem Grad Interaktion mit Bildschirmkarten möglich ist, bleibt das Ziel jeder Interaktion aber immer die *Repräsentation einer Karte auf dem Bildschirm*.

4.1 Qualität der Kartendarstellung

Die Repräsentation einer Bildschirmkarte – die «Sichtbarkeit des Inhaltes» – ist direkt vergleichbar mit Papierkarten. Sie macht zwar – ganz im Gegensatz zu Papierkarten – nur einen Teil einer Bildschirmkarte aus. Trotzdem muss sie gewissen kartografischen Qualitätsanforderungen genügen. Ähnlich wie bei Papierkarten basiert die grafische Modellbildung von Bildschirmkarten auf der Gestaltung von punkt-, linien- und flächenhaften Kartenzeichen. Sie folgt daher den gleichen Grundsätzen wie die topografische und die thematische Kartografie, hat zugleich aber auch Prinzipien zu beachten, die den Vor- und Nachteilen der Bildschirmausgabe Rechnung tragen. Die Vorteile der Bildschirmausgabe sind vor allem in der Interaktion und Nachführung zu suchen, die auf die Repräsentation keinen direkten Einfluss haben.

Die Nachteile der Bildschirmausgabe hingegen – allen voran die schlechte Bildschirmauflösung – haben bedeutende Auswirkungen auf die Kartendarstellung. Erschwerend kommt hinzu, dass sich die technischen Voraussetzungen (Bildschirmqualität, Rechnerleistung) kontinuierlich verändern und eine Bildschirmkarte so gezeichnet werden sollte, dass sie nicht nur heute, sondern auch in Zukunft gut dargestellt werden kann.

Der Kartenbenutzer hat Anspruch auf ein klar lesbares, grafisch gut gestaltetes und einwandfrei interpretierbares Kartenbild (Spiess, 1996a:468). Vom Kartenantor wird daher für die Gestaltung einer kartografisch hochwertigen Karte eine modifizierte Anwendung der Regeln, die für die Modellierung von Papierkarten gelten, verlangt.

Starke Auswirkungen der schlechten Bildschirmauflösung

4.2 Grundsätze zur Kartengestaltung

Für die Gestaltung eines kartografischen Produktes gibt es einige allgemeine Grundsätze. Diese Grundsätze zum Zeichensystem behandeln sowohl grafische Variation als auch grafische Mindestgrößen, ausserdem sind wahrnehmungspsychologische Prozesse zu beachten. Die meisten dieser Grundsätze sind von allgemeiner Natur und können unabhängig vom Ausgabemedium angewandt werden. Grundsätze, die grafische Mindestgrößen oder grafische Variation ansprechen – also diejenigen, die sich direkt auf die Wiedergabe beziehen –, bedürfen auf Dimensionierungsstufe einer Anpassung. Dies wird im Kapitel 4.4 «Bildschirmspezifische Modifizierungen der Grundsätze zur Kartengestaltung» auf Seite 22 geschehen.

Grundlage für die Beschreibung der Kartengrafik als Zeichensystem bildet hier das Kapitel «Kartographische Modellbildung» aus dem Lehrbuch «Kartographie» (Hake und Grünreich, 1994:88–143). Nur Regeln, die aus anderen Quellen stammen, sind speziell vermerkt.

4.2.1 Aufbau des kartografischen Zeichensystems

Kartengrafik ist dreistufig aufgebaut:

1. *Grafische Elemente* sind Punkte, Linien und Flächen. Sie bilden die Bausteine jeder grafischen Darstellung.
2. *Zusammengesetzte Zeichen* sind spezifische Zusammenfügungen der grafischen Elemente zu höheren Gebilden. Für die Kartografie typisch ist die Signatur. Grafische Elemente und zusammengesetzte Zeichen bilden zusammen die kartografischen Gestaltungsmittel.
3. *Grafische Gefüge* ergeben sich, wenn die Elemente und Zeichen bei jeweils bestimmten Objektarten typische grafische Strukturen erzeugen.

4.2.2 Grafische Variation der Zeichen

Durch den Einsatz von grafischen Variablen können die Zeichen in ihrer Erscheinung verändert werden. Allgemein führen alle Variablen zu einer der folgenden Wirkungen:

- ◆ Objektive Gliederung nach Qualitäten bzw. Quantitäten der Objekte durch differenzierte Darstellung.
- ◆ Subjektive Bewertung durch Betonen oder Zurückdrängen.
- ◆ Verstärkte Anschaulichkeit auf der Basis von Assoziationen.

Die nachstehenden Variablen sind möglich:

- Grösse (unterschiedliche Quantitäten)
- Form (unterschiedliche Qualitäten)
- Füllung (meist unterschiedliche Quantitäten, manchmal Qualitäten)
- Tonwert (meist unterschiedliche Quantitäten)
- Richtung (weitere Aufgliederung von Merkmalen, zeitliches Verhalten)
- Farbe (verschiedene Qualitäten, teilweise auch Quantitäten)

Je nach Medium kann die Bedeutung der verschiedenen Variablen sehr unterschiedlich sein. Näheres dazu findet man im Kapitel 4.4 «Bildschirmspezifische Modifizierungen der Grundsätze zur Kartengestaltung» auf Seite 22.

4.2.3 Kartenlogische Bedingungen für die Kartengrafik

Für die Anwendung der Kartengrafik gelten die folgenden Rahmenbedingungen:

1. Massstab und Grundrissdarstellung erfordern eine geometrisch möglichst exakte Anordnung der Zeichen.
2. Bedeutung und Generalisierung der Zeichen führen bei den Gestaltungsmitteln und ihrer Variation zu diesen Grundsätzen:
 - Gleiches gleich, Ungleiches ungleich darstellen
 - Wichtiges erhalten, Unwichtiges weglassen
 - Charakteristisches betonen, weniger Typisches weglassen.

3. Die Lesbarkeit des einzelnen Kartenzeichens setzt voraus:
 - eine visuell noch wahrnehmbare grafische Mindestgrösse
 - die Wahrnehmbarkeit seiner typischen Gestalt
 - die Realisierbarkeit und Konstanzhaltung in den technischen Prozessen wie Zeichnung, Vervielfältigung.
4. Die Lesbarkeit in Bezug auf die gegenseitigen Beziehungen zwischen den Zeichen hat die Aspekte der grafischen Dichte, von Kontrast, Differenzierung und Gewichtung zu berücksichtigen.

4.2.4 Gestaltwahrnehmung

Für die Gestaltwahrnehmung gelten die folgenden Grundsätze:

- ◆ Die grafische Differenzierung muss ausreichend gross sein.
- ◆ Die grafische Dichte darf nicht zu gross sein.
- ◆ Kontrast und Objektrennung müssen ausreichend sein.
- ◆ Der Kontext der Darstellung soll das Erkennen bestimmter Strukturen erleichtern.
- ◆ Optische Täuschungen sollen vermieden werden.

4.2.5 Kartografische Gestaltungsmittel

Mit den kartografischen Gestaltungsmitteln und ihren grafischen Variationen lassen sich jeweils typische Aussagen über Objektmerkmale vornehmen. Damit ergibt sich eine Zuordnung zwischen Objekteigenschaften und Kartengrafik.

- ◆ *Punkte* geben jeweils einzeln die Lage eines Objektes an. Grafische Variation ist nur über Farbe möglich.
- ◆ *Linien* geben eine Lage an. Eine grafische Variation ist nach Farbe oder Strichstärke möglich.
- ◆ *Flächen* sind in ihrer gesamten Ausdehnung in Farbton und Tonwert konstant. Grafische Variation ist nach Farbton und Tonwert möglich.
- ◆ *Signaturen* sind abstrahierte Objektbilder oder konventionelle Zeichen, die man allen grafischen Variationen unterziehen kann. Möglich sind lokale Signaturen, lineare Signaturen, flächenhafte Signaturen.
- ◆ *Diagramme* sind grafische Mittel zur Wiedergabe quantitativer Daten. Die grafische Variation von Diagrammen ändert Art und Umfang der Aussage nicht, kann aber die Verdeutlichung fördern.
- ◆ *Halbtöne* sind Flächen, die im Gegensatz zu Flächenfarben wechselnde Tonwerte aufweisen, z. B. Schummerung.
- ◆ *Kartenschrift* besitzt die geringste geometrische Aussagemöglichkeit, sie ist aber dafür das wichtigste erläuternde Element der Karte.

4.2.6 Grafische Mindestgrössen

Mit kleiner werdendem Massstab schrumpft auch jede massstäbliche Objektwiedergabe immer mehr zusammen, bis schliesslich ihre Lesbarkeit in Frage gestellt ist. Die Mindestgrösse eines gerade noch lesbaren Zeichens spielt daher eine wichtige Rolle. Sie wird durch das menschliche Sehvermögen, die Leistungsfähigkeit der kartentechnischen Verfahren und Umweltbedingungen bestimmt. Unter normalen Beleuchtungsverhältnissen und

mit dem üblichen Betrachtungsabstand gelten für Papierkarten die folgenden Regeln:

Objekt	Dimension
Strichstärken	schwarze Linien 0,05 mm farbige Linien 0,08–0,1 mm
Linienabstände	Haarlinien 0,25 mm etwas stärkere Linien 0,15 mm
Flächendimensionen	runder Punkt 0,3 mm quadratischer Punkt 0,5 mm
Flächenzwischenräume	0,15 mm

Tab. 3 Grafische Mindestgrößen für Papierkarten

Aufgrund der Einschränkungen in der Ausgabequalität gelten auf Bildschirmen ganz andere Werte für Mindestgrößen (vgl. Kapitel 4.4.1 «Grafische Mindestgrößen» auf Seite 23). Sie werden später genau untersucht.

4.2.7 Wahl der Kartenschrift

Nicht jede Schrift ist für Karten gleich gut geeignet. Einige Kriterien, die eine Schrift als gut oder eher weniger gut taxieren, sind hier zusammengestellt (Spiess, 1996b):

- ◆ Schriften sollen nicht zu breit laufen, um ein Kartenbild nicht übermässig zu belasten.
- ◆ Schriften sollen nicht zu eng laufen, da sie dann schwer lesbar werden.
- ◆ Schriften sollen möglichst einfache Formen haben.

4.2.8 Anordnung der Kartenbeschriftung

Entscheidend für eine eindeutige und gute Informationsvermittlung sind die folgenden Aspekte der Schriftanordnung (Spiess, 1996b):

- ◆ Benannte Karteninhalte sollen eindeutig identifizierbar sein.
- ◆ Namen sollen eindeutig lesbar sein.
- ◆ Zusammengehörige Namensteile sollen eindeutig erkennbar sein.
- ◆ Der Karteninhalt soll durch die Schrift möglichst gering beeinträchtigt werden.

Das Schriftbild ist auf jede Störung durch andere Namen oder Kartenelemente sehr empfindlich. Die Labels, also die einzelnen Textobjekte, müssen sehr sorgfältig angeordnet werden. Die Schriftanordnung soll den folgenden Anforderungen entsprechen:

Allgemein:

1. Wo immer möglich sollen die Namen auf horizontale Schriftzeilen gesetzt werden.
2. Die Wortsilhouette ist möglichst freizustellen, insbesondere die Ober- und Unterlängen.
3. Benachbarte Namen sollen vertikal etwas gegeneinander versetzt sein, sich aber nicht auf der selben Schriftzeile befinden. Dies gilt auch für Signaturen.
4. Speziell sorgfältig zu platzieren sind die Namen in Gebieten, wo sie sich drängen, damit eine eindeutige Zuordnung möglich ist. In diesen Fällen

ordnet man die Namen am besten rings um diese dichten Zonen an und setzt möglichst wenige in die Zonen hinein.

5. Trennungen wo immer möglich vermeiden.

Punktsignaturen:

1. Der Zwischenraum zwischen Punktsignatur und Label soll mindestens die halbe Höhe eines Grossbuchstabens haben.
2. Namen sollen wo immer möglich auf derselben Flussseite stehen wie die Stadt, innerhalb der Grenzen des Gebietes, zu dem sie gehören. Namen von Hafenstädten können ins Meer gesetzt, diejenigen von Landstädten sollen dagegen nicht ins Meer gesetzt werden.
1. Für die Anordnung eines Labels bezüglich der Punktsignatur gilt die in Abb. 2 gezeigte Prioritätenreihenfolge.

Flächen:

1. Gebietsnamen sollen durch die Mitte der Fläche geführt werden und sich etwa über $\frac{2}{3}$ der Flächenausdehnung erstrecken.
2. Die Krümmung eines Namens soll nicht gegenläufig zu derjenigen des zugehörigen Gebietes sein.
3. Wo immer möglich sollen auch Flächennamen auf horizontale Linien gesetzt werden, da sie in dieser Stellung am besten zu lesen sind.
4. Gesperrte Namen müssen visuell als eine Einheit erfasst werden können.
5. Gekrümmte Namen sollen immer auf regelmässig gekrümmte Kurven platziert werden. Kombinationen unterschiedlicher Krümmungsradien sind nicht zu empfehlen.
6. Das Verschneiden zweier gesperrter Namen muss immer unter einem möglichst rechten Winkel erfolgen.
7. Namen sollen nie auf gerade, aber rotierte Balken in eine Fläche hineingesetzt werden. Horizontale oder regelmässig gekrümmte Namen sind immer vorzuziehen.

Lineare Elemente:

1. Namen sollen mit etwas Abstand über oder unter die Linie gesetzt werden. Die halbe Höhe der Kleinbuchstaben ist ein gutes Mass für diesen Abstand.
2. Namen sollen nicht auf stark gekrümmte Linienstücke gesetzt werden, sondern dort, wo die Krümmung relativ gering und gleichmässig ist.
3. Namen mit Unterlängen sollen mit Vorteil nicht über, sondern unter die Linie gesetzt werden. Ausnahmen sind höchstens dort möglich, wo sich die Unterlängen gut in die Krümmung einfügen.

4.3 Technische Qualität der Kartendarstellung auf Bildschirmen

Heute stehen eigentlich nur zwei technisch orientierte Wege zur Verfügung, die Qualität der Kartendarstellung auf Bildschirmen zu verbessern. Der erste Weg berücksichtigt das *Antialiasing*, der zweite die Bildschirmauflösung.

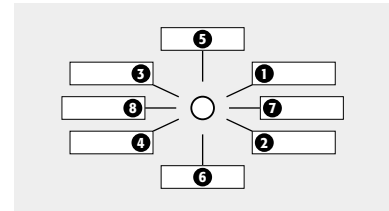


Abb. 2 Prioritätenreihenfolge für die Anordnung von Labels

Täuschung des Auges durch Glättung der Darstellung

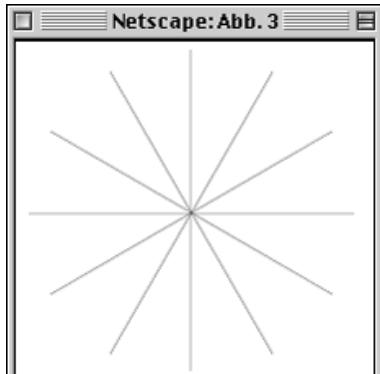


Abb. 3 Schwächen von Antialiasing

Optimale Auflösung für Bildschirme 145 dpi

4.3.1 Einfluss des Antialiasing

Durch *Antialiasing* (Kantenglättung, vgl. Kapitel 2.1.5 «Antialiasing» auf Seite 7) gewinnt die Darstellung aller Objekte generell stark an Qualität. Die Punktgrösse wird optisch auf einen Bruchteil ihrer ursprünglichen Dimension verringert; beim Vektorformat SVG erhält der Punkt optisch 25% der Ursprungsgrösse. Physisch ändert sich die Punktgrösse aber nicht. Das Auge wird dadurch getäuscht, dass der Punkt aufgehellte gezeichnet wird. Konsequenzen hat dies – abgesehen von der Farbveränderung – vor allem auf die Minimalabstände zu anderen Objekten: Wenn Zwischenräume nicht grau erscheinen sollen, müssen sie im Vergleich zu ungeglätteter Darstellung etwas grösser sein.

Im weiteren sind die Darstellung von Linien, die genau horizontal oder vertikal verlaufen, und Flächenkonturen, die horizontal oder vertikal liegen, bei aktiviertem *Antialiasing* problematisch, da sie je nach ihrer Lage sehr unterschiedlich behandelt werden. Liegen die Koordinaten einer Linie genau auf dem Raster des Bildschirms und ist die Strichstärke geeignet, so wird die Linie schärfer ausgegeben, als wenn sie durch Interpolation und mit Hilfe von *Antialiasing* dargestellt werden muss. Linien, deren Winkel von Horizontale und Vertikale abweichen, werden hingegen immer geglättet. Diese Linien werden gleichmässiger, dafür etwas unschärfer dargestellt (siehe Abb. 3).

4.3.2 Einfluss der Bildschirmauflösung

Je grösser die Auflösung ist, desto besser wird auch die Grafikdarstellung. Das Kapitel 2.1.3 «Auflösung» auf Seite 6 hat gezeigt, dass die Auflösung von Rasterbildschirmen in einem Bereich zwischen ca. 72 und 100 dpi variieren kann und diejenige von TFT-Displays sogar deutlich über 100 dpi liegt. Es kann erwartet werden, dass dieser Wert in Zukunft noch um einiges verbessert wird.

Bei einem Leseabstand von 60 cm kann das menschliche Auge Objekt-abstände von 0,175 mm gerade noch als getrennt wahrnehmen (vgl. Kapitel 2.2.1 «Ergonomische Aspekte» auf Seite 7). Unter dem Aspekt des Detailreichtums läge die optimale Auflösung für Bildschirme demnach, gleichbleibender Leseabstand vorausgesetzt, bei 145 dpi. Grössere Auflösungen bringen keinen signifikanten Qualitätsgewinn für Detailzeichnung, da der minimale Objekt-abstand erreicht ist.

Falls der Bildschirm nun eine Auflösung von beispielsweise 81 dpi hat (17-Zoll-Bildschirm mit 1024 * 768 px), wird die Grafik genau dann 1 : 1 ausgegeben, wenn sie auch 81 dpi hat. Verwendet man für die Grafik aber die heute üblichen 72 dpi, so wird die Grafik verkleinert, bei 100 dpi vergrössert.

Dieser Aspekt kann auf dynamische Weise berücksichtigt werden: Bevor eine Karte auf dem Bildschirm ausgegeben wird, ermittelt der Computer sowohl Bildschirmgrösse (z. B. 17 Zoll) als auch Bildschirmauflösung (z. B. 1024 * 768 px). Daraus wird die verwendete Auflösung errechnet (z. B. 81 dpi) und die Grafik entsprechend skaliert.

4.4 Bildschirmspezifische Modifizierungen der Grundsätze zur Kartengestaltung

Um den speziellen Ausgabebedingungen auf Bildschirmen gerecht zu werden, ist es notwendig, einige Grundsätze festzuhalten, die helfen, die Qualität von Bildschirmkarten zu optimieren. Insbesondere die unbekanntenen und sich stets ändernden Ausgabebedingungen (vgl. Kapitel 3.2.2 «Kon-

sistenz in der Ausgabe» auf Seite 12) sollen berücksichtigt werden. Unter anderem aufgrund einer Reihe von Tests werden auf den nächsten Seiten einige wichtige Grundsätze formuliert.

4.4.1 Grafische Mindestgrößen

Basis für die folgenden Untersuchungen ist die Größe eines Punktes. Diese Größe ist relativ und hängt von der Bildschirmauflösung ab (vgl. Kapitel 2.1.4 «Grafische Mindestgrößen» auf Seite 7). Bei 72 dpi Bildschirmauflösung beträgt die Punktgröße 0,35 mm, bei 100 dpi ist sie 0,25 mm.

Um allen Bildschirmen gerecht zu werden, wird vorläufig mit 72 dpi gerechnet. Später soll die Punktgröße aber dynamisch berechnet werden.

- ♦ **Strichstärken:** In Abb. 4 ist erkennbar, dass die Grenze zwischen der feinsten und der nächstgrößeren Strichstärke ungeglättet bei 0,52 mm, geglättet bei 0,13 mm liegt. Die Strichstärken unterhalb von dieser Grenze unterscheiden sich nicht, sie werden alle mathematisch auf den Wert einer Punktgröße gerundet, also 0,35 mm bzw. 0,0875 mm (25% von 0,35 mm).

Die Strichstärkengrenze errechnet sich als das arithmetische Mittel zwischen der Größe eines Punktes und der Größe von zwei Punkten, also:

$$\text{ungeglättet: } \frac{0,35 \text{ mm} + 2(0,35 \text{ mm})}{2} = 0,525 \text{ mm}$$

$$\text{geglättet: } \frac{0,0875 \text{ mm} + 2(0,0875 \text{ mm})}{2} = 0,13125 \text{ mm}$$

- ♦ **Strichstärkenabstufungen:** Je nach Art der Darstellung ist die Abstufung der darstellbaren Strichstärken sehr unterschiedlich: Ungeglättet nimmt die dargestellte Strichstärke sprunghaft als Vielfaches der Punktgröße zu. Geglättet ist insgesamt eine kontinuierliche Zunahme feststellbar.
- ♦ **Linienabstände:** Optisch zuverlässig durch einen weissen Zwischenraum trennbar sind zwei Haarlinien (feinste darstellbare Strichstärke) erst, wenn ihr Abstand geglättet zwei Punktgrößen (0,7 mm) beträgt (siehe Abb. 5). Ungeglättet werden nicht mehr alle Linien dargestellt. Stärkere Linien können ungeglättet und geglättet ab 0,5 mm getrennt werden. Der Zwischenraum erscheint allerdings etwas grau. Der hohe Wert für geglättete Haarlinien kann damit erklärt werden, dass die Linie, wie oben erwähnt, nur durch einen Grauwert optisch fein erscheint, aber eigentlich 0,35 mm stark ist. Durch den Grauwert ist der Kontrast zum Hintergrund gering und der Abstand muss aus Kompensationsgründen deutlicher sein. Bei stärkeren Linien wird durch das *Antialiasing* ein Hof um die Linie gebildet. Fallen zwei dieser Höfe zusammen, so ergänzen sie sich und erscheinen dunkler; die Linie ist nicht mehr sauber trennbar.
- ♦ **Flächendimensionen:** In geglätteter Darstellung wird ein eckiger Punkt erstmals bei einer Kantenlänge von 1,2 mm als Quadrat erkennbar (siehe Abb. 6). Ein runder Punkt kann ab 1,4 mm als Kreis erkannt werden. Ungeglättet sind die Werte so hoch, dass sie nicht mehr berücksichtigt werden müssen.
- ♦ **Flächenzwischenräume:** Ungeglättet sind Flächenzwischenräume erst ab der zweifachen Punktgröße (0,7 mm) klar unterscheidbar (siehe Abb. 7). Geglättet erscheinen zwei Flächen ab etwa 0,5 mm optisch getrennt, wobei der Zwischenraum aber etwas grau erscheint.

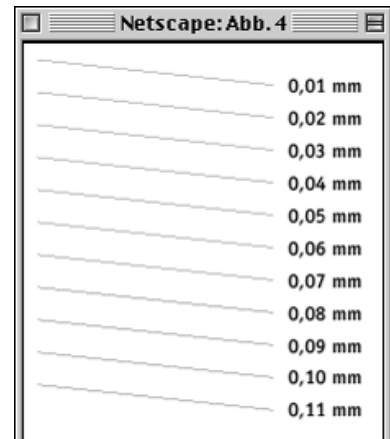


Abb. 4 Darstellungsgrenzen von Strichstärken

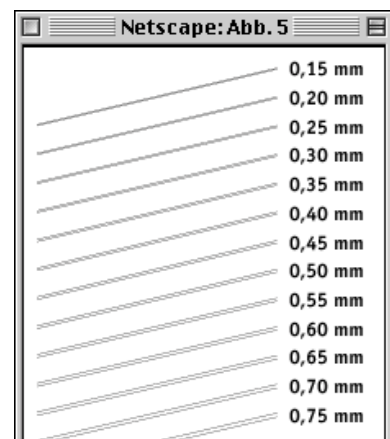


Abb. 5 Minimale Linienabstände

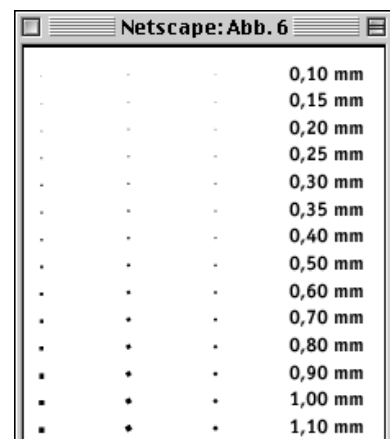


Abb. 6 Minimale Flächendimensionen

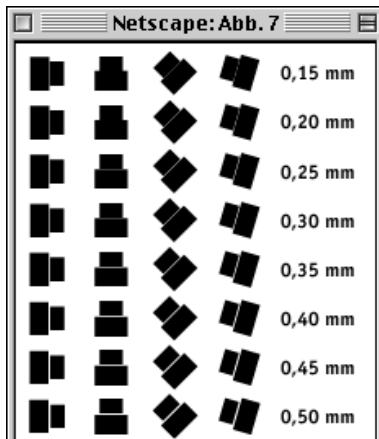


Abb. 7 Minimale Flächenzwischenräume

Die Ausgabe ohne *Antialiasing* wird angesichts der offensichtlich grossen Nachteile in Zukunft nicht mehr berücksichtigt (vgl. Kapitel 4.2.6 ‹Grafische Mindestgrössen› auf Seite 19).

Zusammenfassend können folgende grafische Mindestgrössen festgestellt werden:

Objekt	Dimension (mm)	Dimension (px)
Strichstärken	0,09	0,26
Linienabstände	Haarlinien 0,7 stärkere Linien 0,5	Haarlinien 2,0 stärkere Linien 1,4
Flächendimensionen	runder Punkt 1,4 quadrat. Punkt 1,2	runder Punkt 4,0 quadrat. Punkt 3,4
Flächenzwischenräume	0,5	1,4

Tab. 4 Grafische Mindestgrössen für Bildschirmkarten mit *Antialiasing*

4.4.2 Wahl der Schrift für Bildschirmkarten

Nicht alle Schriften sind für Bildschirmkarten gleich gut geeignet. Die Kriterien, die für die Wahl von Kartenschriften in Kapitel 4.2.7 ‹Wahl der Kartenschrift› auf Seite 20 zusammengestellt sind, gelten auch für Bildschirmkarten. Sie müssen aber aufgrund der schlechten Auflösung von Bildschirmen noch deutlich erweitert werden.

Würde eine Bildschirmkarte ohne *Antialiasing* ausgegeben, dann wären auf Bildschirmen nur diejenigen Schriften zu verwenden, welche das gerade benutzte Betriebssystem als Bildschirmschriften anbietet. Beim Mac OS kämen nur gerade *Geneva* in 10 und 11 px Grösse und *Charcoal* in 14 px Grösse oder andere speziell für Bildschirme entwickelte Schriften in Frage. Die meisten anderen Schriften wären kaum lesbar.

Mit *Antialiasing* sieht die Situation etwas besser aus. Die Auswahl der Schriften ist deutlich grösser, und sie können in unterschiedlichen Schriftgrössen verwendet werden. Dennoch sind Schriften nicht zu klein oder zu gross zu wählen, weil ihre Lesbarkeit aus technischen wie aus typografischen Gründen schnell beeinträchtigt ist. Schriften, die sehr feine Elemente enthalten, sind zu vermeiden, da diese – falls sie überhaupt dargestellt werden – nicht im richtigen Ausmass zur Geltung kommen. Auf Schriften mit Serifen soll also grundsätzlich verzichtet werden. Kursiv gestellte Schriften sind kritisch, können aber unter Umständen verwendet werden.

4.5 Bildschirmspezifische Vergrösserung der Kartengeometrie

Wenn eine für Papier konzipierte, aber am Computer produzierte Karte für die Bildschirmausgabe aufbereitet werden soll, dann muss sie im Vergleich vergrössert dargestellt werden. Dieser *Vergrösserungsfaktor* F setzt sich aus zwei Teilen zusammen. Einerseits ist die notwendige Vergrösserung kartografisch durch die Generalisierung begründet; dies äussert sich im *kartografischen Vergrösserungsfaktor* F_k . Andererseits ist die Stärke der Vergrösserung abhängig von den Bildschirmeinstellungen, sichtbar im *technischen Vergrösserungsfaktor* F_t .

Der *kartografische Vergrösserungsfaktor* F_k hat zum Ziel, die Generalisierung der Kartengeometrie an die Bildschirmauflösung anzugleichen. Trotz der unterschiedlichen Ausgabebedingungen soll der optische Eindruck etwa der gleiche sein. Der Faktor hängt insbesondere von der Distanz zum Medium ab (vgl. Kapitel 2.2.1 ‹Ergonomische Aspekte› auf Seite 7 und

Kapitel 4.3.2 «Einfluss der Bildschirmauflösung» auf Seite 22). Der Betrachter ist von Papier in der Regel etwa 25 cm, vom Bildschirm 60 cm entfernt. Dies sind Durchschnittswerte, die individuell stark verschieden sein können. Das Verhältnis zwischen den Distanzen bleibt aber immer etwa gleich. Ein Leser, der – z. B. wegen Altersweitsichtigkeit – 30 cm vom Papier entfernt ist, wird entsprechend auch etwa 70 cm vom Bildschirm entfernt sein.

Der kartografische Vergrößerungsfaktor lässt sich mit Hilfe des Strahlensatzes berechnen: Das Verhältnis zwischen den Distanzen (60 cm : 25 cm) entspricht dem Verhältnis der Vergrößerung. Der kartografische Vergrößerungsfaktor beträgt folglich:

$$F_k = \frac{60}{25} = 2.4 \quad (1)$$

Um den optischen Eindruck einer Bildschirmkarte im Vergleich zu ihrem Papierpendant zu erhalten, ist demnach eine Vergrößerung der Kartengeometrie um den Faktor 2,4 notwendig.

Vergrößerung der Kartengeometrie um Faktor 2,4

Theoretisch gilt dieser Wert nur für eine Bildschirmauflösung von 145 dpi. Bei tieferen Auflösungen würde der Faktor um einiges grösser. Da Grafik aber in der Regel mit Hilfe von *Antialiasing* ausgegeben und somit eine höhere Auflösung simuliert wird (vgl. Kapitel 4.3.1 «Einfluss des Antialiasing» auf Seite 22), genügt dieser Faktor, um auch eine Auflösung von 72 dpi noch gut darzustellen. Umgekehrt gilt aber folgerichtig auch, dass eine Erhöhung der Auflösung auf den Detailreichtum keinen positiven Effekt hat, sondern nur auf die Darstellungsqualität.

Der *technische Vergrößerungsfaktor* F_t ist abhängig von der Umgebung. Heute werden viele Grafikformate, auch SVG, grundsätzlich mit 72 dpi ausgegeben. Folglich wird nur dann, wenn ein Bildschirm genau auf 72 dpi eingestellt ist (z. B. ein 15-Zoll-Bildschirm, der 800 * 600 px anzeigt), Grafik wirklich 1 : 1 dargestellt (vgl. Kapitel 4.3.2 «Einfluss der Bildschirmauflösung» auf Seite 22). In allen anderen Fällen mit grösserer Auflösung wird die Grafik einfach verkleinert.

Diese Eigenheit lässt sich ausgleichen, wenn die genaue Bildschirmauflösung bekannt ist. Der technische Vergrößerungsfaktor wird dann wie folgt berechnet:

$$F_t = \frac{R}{72} \quad (2)$$

wobei R = Bildschirmauflösung des benutzten Monitors.

Der *Vergrößerungsfaktor* F berechnet sich also nach (3):

$$F = F_k F_t = 2.4 \left(\frac{R}{72} \right) \quad (3)$$

Beispielsweise auf einem 17-Zoll-Bildschirm mit einer Bildfläche von 325 * 240 mm und 1024 * 768 angezeigten Pixeln (81 dpi) muss eine Karte demnach für eine optimale Darstellung um den Faktor 2,7 vergrössert werden.

Während der kartografische Vergrößerungsfaktor hart codiert werden kann, muss der technische Vergrößerungsfaktor je nach Umfeld jederzeit dynamisch zugewiesen werden können.

5 Interaktion mit Bildschirmkarten

Zu einem grossen Teil sind Bildschirmkarten heute nichts anderes als eine Umsetzung von Papierkarten auf den Bildschirm. Multimediatechnologie ermöglicht es aber, eine Karte anders zu gestalten, als dies auf Papier möglich ist, also ein Produkt zu kreieren, das die Technologie ausnützt und einen anderen Weg der Präsentation geografischer Daten geht.

Zur Vermittlung komplexer räumlicher Daten hat die virtuelle Anzeige deutliche Vorteile gegenüber der traditionellen Kartografie (MacEachren et al., 1999). Multimedia hat einerseits das Potenzial, Informationen durch sehr viel mehr Kanäle, etwa bewegte Bilder oder Ton, zu vermitteln als Papierkarten. Benutzern soll andererseits ermöglicht werden, ihren Weg durch die Informationsbasis selber zu suchen (Cartwright et al., 1999:602), ohne dass der Informationslieferant einen klaren Ablauf vorgibt. Die Anzeige der virtuellen Welt soll – parallel zum Wissensstand und zur Erfahrung der Benutzerin – nach und nach individuell erweitert werden können.

Im Gegensatz zu Papierkarten beinhaltet die Entwicklung einer interaktiven Bildschirmkarte mehr als die schlichte Produktion der Karte. Von grosser Bedeutung ist auch die Art der visuellen Repräsentation. Hierfür gut geeignet sind Methoden, die den grösstmöglichen Nutzen aus denjenigen menschlichen Sensoren und kognitiven Systemen ziehen, die für die Interaktion mit der realen Welt zuständig sind (MacEachren et al., 1999).

Informationsvermittlung durch diverse Kanäle

5.1 Mentale Schnittstelle: Metamodell für die Repräsentation

Für das Verständnis räumlicher Darstellungen ist die mentale Schnittstelle, die Konstruktion eines mentalen Bildes im menschlichen Kopf, sehr wichtig (Bidoshi et al., 1999:592). Eine grosse Hilfe dazu sind Metamodelle. Im folgenden wird kurz das Metamodell vorgestellt, das Heim (Heim, 1998, zit. in MacEachren et al., 1999) und MacEachren (MacEachren et al., 1999) entwickelt haben.

Räumliche virtuelle Umgebungen (*geospatial virtual environment*, GeoVE), wie sie MacEachren (MacEachren et al., 1999) erwähnt, können zur dreidimensionalen Beschreibung von mehr als nur den sichtbaren Charakteristiken von geografischen Umgebungen verwendet werden. In so genannten räumlichen virtuellen *super environments* kann der Benutzer nicht nur sehen, was in der realen Welt sichtbar wäre. Er kann auch das normalerweise Unsichtbare erfahren und kontrollieren, was sich üblicherweise der menschlichen Kontrolle entzieht (MacEachren et al., 1999).

Heim und MacEachren schlagen vier Metafaktoren vor, die GeoVE mit realen Umgebungen verbinden: *Eintauchen* beschreibt das Erlebnis, «in einer Umgebung zu sein». *Interaktivität* erlaubt es dem Benutzer, seine Sichtrichtung oder auch seine relative Position zu ändern. *Informationsintensität* bezieht sich auf den Detaillierungsgrad, in dem Objekte und Merkmale dargestellt werden. Die *Intelligenz der Anzeigeobjekte* verweist auf das Ausmass des kontextsensitiven Verhaltens von Anzeigeobjekten bei Interaktion.

MacEachren beschreibt diese Metafaktoren der virtuellen Welt als Grundlage für (dreidimensionale) GeoVE. Sie sind aber auch die Basis für zweidimensionale räumliche Darstellungen, wie sie in diesem Kapitel beschrieben werden.

5.1.1 Eintauchen

In der realen Welt sind alle Sinne stets in irgend einer Form involviert. Der Grad des Eintauchens in die virtuelle Umgebung wird daher dann grösser, wenn mehr Sinne in höherem Ausmass stimuliert werden. Ein Schlüsselproblem ist es, diejenigen typischen Merkmale der Umgebung zu identifizieren, die zu einem möglichst guten Eintauchen in die räumliche Darstellung führen.

5.1.2 Interaktivität

Heim (Heim, 1998, zit. in MacEachren et al., 1999) versteht unter Interaktivität, dass eine Benutzerin an einer virtuellen Erfahrung durch Änderung ihrer Position oder ihrer Blickrichtung teilhaben kann. Auch andere Formen der Interaktion, wie Manipulation der Objektcharakteristiken (Farbe, Sichtbarkeit, usw.) sind denkbar.

5.1.3 Informationsintensität

Objekte und deren Eigenschaften müssen in räumlichen Darstellungen nicht aussehen wie in der realen Umgebung. Notwendig ist aber, dass der Detaillierungsgrad in etwa demjenigen entspricht, den der Benutzer in der realen Umgebung abhängig von seiner Distanz zum Objekt auch erwartet. Ausserdem soll, genau wie in der realen Umgebung, durch eine Annäherung an die Objekte der Detaillierungsgrad steigen.

5.1.4 Intelligenz der Anzeigeobjekte

Die Wissensvermittlung durch räumliche Darstellungen kann verbessert werden, wenn die Objekte kontextsensitives Verhalten aufweisen. Dies kann z. B. in Form von Animation sein, etwa, um die Aufmerksamkeit des Benutzers auf ein bestimmtes Objekt zu lenken (Bidoshi et al., 1999:595).

5.2 Visuelle Schnittstelle: Bedeutung von grafischen Benutzeroberflächen

Die visuelle Schnittstelle zwischen dem Benutzer und der räumlichen Darstellung ist die Benutzeroberfläche. Der Erfolg eines Multimedia-Produktes wird hauptsächlich von der Qualität der Benutzeroberfläche bestimmt (Apple Computer, Inc., 1994, zit. in Miller, 1999:54). Weil für räumliche Darstellungen Interaktion sehr wichtig ist – fast alle Vorteile der Bildschirmkarten im Vergleich zu Papierkarten liegen hier –, kommt der Benutzeroberfläche enorme Bedeutung zu.

Anders als bei Papierkarten ist das Verständnis des Inhalts und des möglichen Nutzens einer Bildschirmkarte in grossem Masse abhängig von der Benutzeroberfläche (Arleth, 1999:854). Schnittstellenmetaphern grafischer Benutzeroberflächen sind nicht nur konzeptionelle, sondern auch präsentierende Ausgabearten (MacEachren et al., 1999). Eine räumliche Darstellung mit einer schlecht gestalteten Benutzeroberfläche wird den kognitiven Prozess blockieren. Wenn die Benutzerin Zeit und geistige Ressourcen aufwenden muss, um herauszufinden, wie eine Bildschirmkarte benutzt wird, kann sie weniger Zeit auf die Lösung ihrer Aufgabe verwenden (Lindholm und Sarjakoski, 1994, zit. in Arleth, 1999:854).

*Sehr gute Einführung zu
Benutzeroberflächen in
Raskin, 2000*

5.3 Möglichkeiten der Interaktion

Mit den Mitteln von Multimedia sind zahlreiche Arten der Interaktion für den Benutzer möglich. Sieber und Bär (Sieber und Bär, 1997) schlagen die im folgenden aufgeführten Möglichkeiten für interaktive Atlanten vor. Viele dieser Möglichkeiten eignen sich nur für interaktive Atlanten oder spezielle Formen der Bildschirmkarten, andere wiederum sind eher von allgemeiner Natur.

5.3.1 Funktionen zur thematischen Navigation

Nutzbringende Informationen sollen nicht zufällig angetroffen, sondern aktiv dem Benutzer angeboten werden. Deshalb sind Funktionen zur thematischen Navigation für interaktive Atlanten wesentlich. In vereinfachter Form sind sie auch für nicht in einen Atlas eingebundene Bildschirmkarten wichtig.

- ◆ Freie Wahl eines Themas (innerhalb der zur Verfügung stehenden Möglichkeiten)
- ◆ Schneller Themenwechsel
- ◆ Index-Funktionen, Bookmarks
- ◆ Hinweise zu verwandten Themen

5.3.2 Funktionen zur räumlichen Navigation und Orientierung

Orientierungs- und Navigationsfunktionen nehmen nicht nur in interaktiven Atlanten einen hohen Stellenwert ein, sie sind auch für räumliche Darstellungen, die nicht in einen Atlas eingebunden sind, von sehr grosser Bedeutung.

- ◆ Orientierung mit Hilfe einer Referenzkarte
- ◆ Zoomen
- ◆ Scrollen
- ◆ Verschieben des Kartenausschnittes
- ◆ Lupen-Funktion
- ◆ Karten-Rotation
- ◆ Ortsindex (Suchen), Lokalität
- ◆ Gebietsindex
- ◆ Persönliche Marken

5.3.3 Funktionen zur Visualisierung

Funktionen, welche die Darstellung von Atlaskomponenten verändern, sollen autorenkontrolliert eingesetzt und damit selektiv zur Verstärkung der Kartenaussage verwendet werden. Ihnen kommt in jeder Form der räumlichen Visualisierung grosse Bedeutung zu.

- ◆ Veränderung der Symbolisierung (z. B. Farbgebung)
- ◆ Wahl des Diagrammtyps
- ◆ Komposition von Ebenen
- ◆ Veränderbare Klassenbildung und Wechsel der räumlichen Einheiten bei Karten
- ◆ Abspielen von Bildsequenzen
- ◆ Zusammenstellen benutzerdefinierter Sequenzen

5.3.4 Gis-Funktionen

Die GIS-Funktionalität ist darauf ausgerichtet, sowohl grafische wie auch in Datenform vorliegende Informationen zu behandeln.

- ◆ Raumbezogene Abfragen
- ◆ Thematische Abfragen
- ◆ Messungen von Distanzen und Flächen
- ◆ Vergleiche innerhalb einer Karte oder eines Themas
- ◆ Analysefunktionen wie Zonenbildung, Verschneidung, Aggregation

6 Technische Grundlagen von SVG

Im Kapitel 3.3.3 «SVG als Dateiformat für Bildschirmkarten» auf Seite 15 wurde gezeigt, dass das Dateiformat SVG (*scalable vector graphics*) für die interaktive Kartografie hervorragend geeignet ist. Auf den folgenden Seiten wird der Aufbau von SVG als XML-Anwendung kurz beschrieben, und die für interaktive Karten relevanten Funktionen und Möglichkeiten werden erläutert. Dieses Kapitel ist als kurzer Überblick über die Möglichkeiten zu verstehen; genaue Funktionsweisen sind nicht beschrieben.

Zum Zeitpunkt der Entstehung dieser Arbeit befindet sich, wie früher schon erwähnt, SVG noch in der Testphase. Werkzeuge stehen nur spärlich zur Verfügung und haben leider noch nicht alle notwendigen und gewünschten Funktionen implementiert (vgl. auch Kapitel 3.3.3 «SVG als Dateiformat für Bildschirmkarten» auf Seite 15).

Beispielweise das Vektorgrafikprogramm *Adobe Illustrator* unterstützt in der Version 9 zwar den Export von SVG (vgl. Kapitel 3.3.2 «Kartografische Anforderungen an ein Dateiformat» auf Seite 14). Dieser beschränkt sich aber leider fast vollständig auf die Vektoren. Wichtige Authoring-Möglichkeiten, z. B. *scripting*, sind nur sehr rudimentär ausgebaut oder gar nicht implementiert, Online-Hilfe und Manual kaum erwähnenswert.

Umgekehrt bietet nur ein mit einem entsprechenden Plug-In ausgestatteter Webbrowser zurzeit die Möglichkeit der Ausgabe. Auch hier ist die Funktionalität – je nach Browser – stark eingeschränkt. Im Augenblick bietet unter *Mac OS* nur *Netscape Communicator* ab Version 4 und SVG-Plug-In Interaktivität, der *Internet Explorer* von *Microsoft* versteht sich ausschliesslich auf die statische Ausgabe (*Adobe, 2000c*). Unter *Windows* sieht die Situation etwas besser aus, sowohl *Internet Explorer* als auch *Netscape* kennen Interaktivität.

SVG ist eine XML-Anwendung. Um mit SVG arbeiten zu können, müssen die Grundbegriffe von XML daher bekannt sein. Hochentwickelter Einsatz von SVG ist durch die Verwendung von zusätzlichen Skriptsprachen wie *JavaScript* oder Programmiersprachen wie *Java* möglich, die durch das DOM von SVG Zugang zu allen Objekten, Attributen und Eigenschaften haben. Um dem Entwickler maximale Flexibilität bei möglichst einfacher Dokumentpflege und kleinen Dokumentgrössen zu ermöglichen, verwendet SVG intensiv Stilvorlagen, die sich sowohl auf Schrift- als auch auf Grafikstile beziehen können.

6.1 XML im Überblick

XML (*extensible markup language*) ist eine Methode, mit der strukturierte Daten wie z. B. *spreadsheets*, Adressbücher, Konfigurationsparameter, Finanztransaktionen oder technische Zeichnungen in ein Textdokument gepackt werden. Wenn die Daten in ein Text-, statt in ein binäres Dokument gespeichert werden, hat dies den Vorteil, dass sie ohne das Programm, das die Daten ursprünglich geschrieben hat, betrachtet und modifiziert werden können; ein einfacher Texteditor genügt einem Entwickler, XML-Anwendungen zu verändern.

XML ist eine Regelsammlung, die hilft, Textdokumente für strukturierte Daten so zu gestalten, dass sie leicht zu generieren und zu lesen sind, dass sie unzweideutig sind, und dass sie keine der verbreiteten Missstände wie Plattformabhängigkeit, das Fehlen von Erweiterungsmöglichkeiten oder mangelnde Lokalisierung enthalten. XML ist ähnlich aufgebaut wie HTML; es benutzt *tags* und *attributes*. Während aber in HTML spezifiziert ist, was jedes *tag* und jedes *attribute* bedeutet und wie der Text in einem Browser aussehen

www.w3.org/xml/
www.ibm.com/developer/xml/
www.filemaker.com/xml/
www.developerlife.com/

XML zur Gestaltung strukturierter Daten

soll, ist XML nur definiert, was ein *tag* oder ein *attribute* ist. XML ist also eine Metasprache, mit deren Hilfe andere *markup languages* definiert werden können (Netscape Communications Corporation, 1999c). Sie gebraucht *tags* allein, um Datenteile abzugrenzen, und überlässt die Interpretation der Daten vollständig der Applikation, die sie liest. Beispielsweise ein `<p>`-tag muss in XML im Gegensatz zu HTML nicht bedeuten, dass es sich beim nachfolgenden Text um einen Absatz (*paragraph*) handelt. Genau so gut könnte `<p>` ein Parameter oder ein Preis sein (Bos, 1999).

Die Regeln für XML sind sehr viel strenger als für HTML. Ein vergessenes *tag* oder ein *attribute* ohne Anführungszeichen macht das Dokument unlesbar. In der XML-Spezifikation wird – ganz im Gegensatz zu HTML – Anwendungen der Versuch ausdrücklich untersagt, ein falsches *tag* anders interpretieren zu wollen. Eine fehlerhafte XML-Anwendung muss am Ort des Fehlers stoppen und eine Fehlermeldung ausgeben.

Um XML herum gibt es eine stetig grösser werdende Anzahl von optionalen Modulen, die ihrerseits Sets von *tags*, *attributes* und Richtlinien für spezifische Aufgaben zur Verfügung stellen. Solche Module nennt man auch *XML-applications*. Beispiele sind XLink (*XML linking language*, ein Standardweg, um Hyperlinks in eine XML-Anwendung einzufügen), CSS (*cascading style sheets*, Stilvorlagen für XML- und HTML-Dokumente), XSL (*extensible style sheet language*, eine fortgeschrittene Sprache, um Stilvorlagen für XML-Anwendungen zu beschreiben) oder DOM (*document object model*, ein Standardsatz von Funktionsaufrufen, um XML-Anwendungen von aussen zu manipulieren) (Bos, 1999). Auch SVG ist eines dieser Module. Die meisten der erwähnten Module sind zur Zeit der Entstehung dieser Arbeit noch in der Entwurfsphase, oft als *last call* oder *candidate recommendation*.

XML ist zwar eine neue Sprache, aber dennoch nicht ganz neu. Die aktuelle Version XML 1.0 wurde zwar von 1996 bis 1998 entwickelt. Aber XML ist eigentlich ein *subset* von SGML (*standard generalized markup language*, eine ISO-Norm von 1986), das verbreitet für grosse Dokumentationsprojekte angewendet wird.

6.1.1 Die Konzepte von XML

Eine XML-Anwendung hat sowohl eine logische als auch eine physische Struktur (siehe Abb. 8). Die logische Struktur erlaubt es, ein Dokument in bezeichnete Einheiten und Untereinheiten, sogenannte Elemente, aufzuteilen. Die physische Struktur hingegen ermöglicht es, dass beliebige Komponenten eines Dokumentes, sogenannte *entities*, einzeln bezeichnet und abgelegt werden können. Auf diese Weise können die Information mehrmals benutzt, anderen XML-Anwendungen zugänglich gemacht, oder etwa Rasterbilder, die nicht als XML-Anwendung vorhanden sind, per Referenz eingefügt werden. Beispielsweise kann jedes Kapitel eines Buches durch ein Element repräsentiert werden, das weitere Elemente enthalten kann, die jeden Absatz, jede Tabelle und jedes Bild beschreiben. Die eigentlichen Bild-daten und die Absätze, die mehrmals verwendet werden, legt man hingegen in separaten *entity files* ab.

Die logische Struktur einer XML-Anwendung obliegt einigen Zwängen. Ein *parser* prüft daher vorerst bei der Ausgabe, ob eine XML-Anwendung Fehler aufweist. Dokumente, die sich aus mehreren *entities* zusammensetzen, werden dann durch einen *entity manager* zusammengefügt. Software, die beide Aufgaben gleichzeitig übernimmt, wird korrekterweise *XML-processor*, oft aber einfach nur *parser* genannt (Bradley, 2000:4).

XML bietet im Prinzip komplette Freiheit in der Gestaltung von Elementen, vordefinierte Elemente existieren nicht. Um aber deren Anzahl zugunsten der Übersicht in Grenzen zu halten, ist es möglich, die in der spezifischen Anwendung erlaubten Elemente vorzudefinieren. Eine DTD (*document type definition*) enthält dann alle definierten Elemente, und

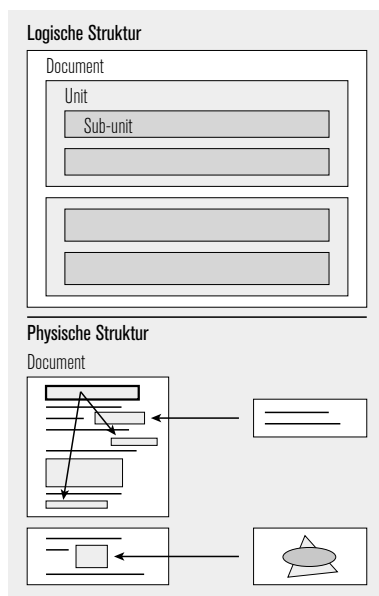


Abb. 8 Logische und physische Struktur einer XML-Anwendung (nach Bradley, 2000:4)

ein entsprechender *parser* vergleicht die DTD-Regeln mit dem Dokument und prüft dessen Gültigkeit. Mit dieser Eigenschaft kann ein Dokument so aufbereitet werden, dass es einer bekannten Spezifikation entspricht, und es kann geprüft werden, ob die Elemente korrekt angewandt wurden (Bradley, 2000:5).

In XML werden Elemente benutzt, deren Namen die Natur der Objekte beschreiben, und nicht, wie sie angezeigt oder gedruckt werden sollen. Im Gegensatz zu traditionellen stilbasierten Instruktionen hat dieser *generalized markup*-Ansatz den grossen Vorteil, dass die Information selbstbeschreibend ist, sie kann daher beliebig lokalisiert oder manipuliert werden. Andererseits ist es nicht möglich, eine passende Präsentation des Dokumentes automatisch zu produzieren, da es für den Computer nicht offensichtlich sein kann, wie ein Element, das beispielsweise *paragraph* heisst, dargestellt werden muss. Daher wird für die Spezifizierung eines Ausgabeformates eine Stilvorlage (*style sheet*) für jedes Element benötigt. Der grosse Vorteil dieser Methode ist, dass beispielsweise für verschiedene Ausgabemedien unterschiedlich optimierte *style sheets* benutzt werden können (Bradley, 2000:6).

6.1.2 XML Document Markup

XML-tags rahmen ein identifiziertes Objekt ein (siehe Abb. 9): Ein *start tag* und ein *end tag* zusammen mit den eingeschlossenen Daten entsprechen einem vollständigen Element. Auf diese Art identifiziert eine XML-Anwendung einzelne Objekte.

XML-Elemente können weitere, in sie eingebettete Elemente enthalten, und das ganze Dokument muss in einem *document element* enthalten sein. Die hierarchische Dokumentstruktur kann als Boxen (siehe Abb. 8 auf Seite 32) oder auch als Baum (*document tree*) mit *nodes* und *leaves* visualisiert werden (siehe Abb. 10). Ein typisches Beispiel wäre ein Buchelement, das Kapitelelemente enthalten kann, die wiederum Absatzelemente enthalten können. Jedes Element muss dabei komplett in ein übergeordnetes Element eingeschlossen sein (Bradley, 2000:17).

Die Beziehungen zwischen einzelnen Elementen können wie folgt beschrieben werden: Beispielsweise aus der Perspektive eines Kapitelelements sind alle anderen Kapitelelemente *siblings*, also Geschwister, das Buchelement ist der *parent*, und alle im Kapitelelement enthaltenen Absatzelemente sind *children* (siehe Abb. 11).

Jedes Element kann nur ein einzelner *parent*, aber beliebig viele *children* haben, mit Ausnahme des *document element*, das weder *parent* noch *siblings* haben kann.

Ein Element kann neben einem Elementnamen weitere Informationen speichern; diese Informationen sind in einem *attribute* abgespeichert, wobei ein Element beliebig viele *attributes* enthalten kann. Solche *attributes* können z. B. ein Sicherheitslevel oder der Name des Autors sein. In Verbindung mit einer entsprechenden DTD sind *attributes* ein hervorragendes Hilfsmittel (Bradley, 2000:20), von dem auch SVG extensiven Gebrauch macht.

6.1.3 Physische Struktur einer XML-Anwendung

Die XML-Spezifikation enthält mit den *entities* eine Möglichkeit, jeden beliebigen Teil eines Dokumentes zu isolieren und separat zu speichern. Beispielsweise kann jedes Kapitel eines Buches separat gespeichert werden. Jede Einheit von Information ist eine *entity* und muss einen eindeutigen Namen haben, mit dessen Hilfe sie identifiziert werden kann.

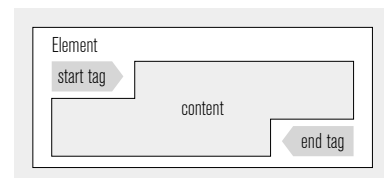


Abb. 9 Ein XML-Element mit Inhalt und seinen start und end tags (nach Bradley, 2000:17)

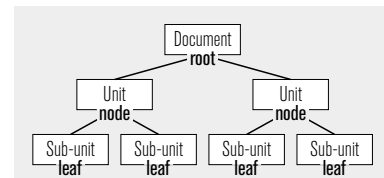


Abb. 10 Document tree einer XML-Anwendung

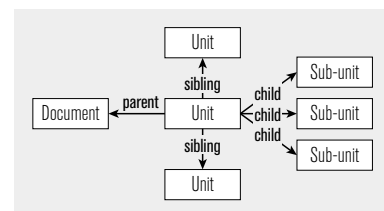


Abb. 11 Beziehungen zwischen Elementen

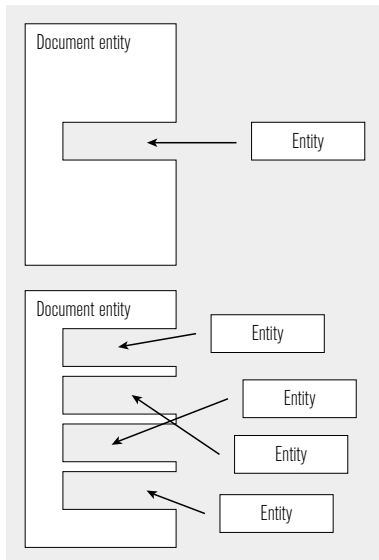


Abb. 12 Die *document entity* mit weiteren *entities* (nach Bradley, 2000:35)

Die *document entity* ist das gesamte Dokument und repräsentiert den Hauptinhalt. Sie muss als einzige keinen Namen haben. In einfachen Fällen ist die *document entity* die einzige *entity*. Häufiger aber enthält die *document entity* den grössten Teil des Dokumenteninhaltes und benutzt weitere *entities*, um die bestehenden Lücken zu füllen (siehe Abb. 12). Im Extremfall kann das Dokument nur noch aus einem Rahmen bestehen, der dazu benutzt wird, den Inhalt anderer *entities* zu platzieren (Bradley, 2000:35).

Eine *entity* ist mit Hilfe eines speziellen tags, der *entity declaration*, definiert; dieses tag erscheint am Anfang der *document entity* und deklariert insbesondere deren Existenz. Im weiteren erhält die *entity* einen Namen zur künftigen Bezugnahme und enthält entweder deren Inhalt (*internal entity*), oder verweist auf ein entsprechendes Dokument (*external entity*). Eine *external entity* kann auch Daten enthalten, die nicht dem XML-Format entsprechen, beispielsweise Pixelbilder (Bradley, 2000:36).

Entities werden benutzt, indem man innerhalb des Textes Referenzen zu ihnen platziert. Eine *entity reference* identifiziert die benötigte *entity*, und ihre Position definiert, wo ihr Inhalt erscheinen soll. Es sind beliebig viele *entity references* möglich. *Entities* können hierarchisch aufgebaut und beliebig tief verschachtelt sein, dürfen aber keine *references* zu sich selbst enthalten (Bradley, 2000:36).

6.1.4 Logische Struktur einer XML-Anwendung

Eine wichtige Eigenschaft von XML ist die Fähigkeit, eine Vorlage für die *document markup* zu schaffen, so dass die Platzierung der Elemente und deren *attributes* kontrolliert erfolgen und später reproduziert werden kann. Dies geschieht in der DTD (*document type definition*). Die DTD ist eine optionale, aber sehr mächtige Eigenschaft von XML, die ein formales Regelwerk zur Definition der Dokumentstruktur liefert. Sie ist vergleichbar mit den Stilvorlagen in Textverarbeitungsprogrammen, aber ungleich stärker. Mit der DTD werden formale Dokumentstrukturregeln festgelegt. Sie definiert die Elemente, die benutzt werden können, und schreibt vor, wo sie in Beziehung untereinander verwendet werden können. Sie definiert letztendlich die Hierarchie des Dokuments (Bradley, 2000:48).

6.1.5 Dokumentformatierung mit css

CSS (*cascading style sheets*) ist eine Stilvorlagensprache, die es Autoren und Benutzern erlaubt, strukturierte Dokumente mit *attributes* (z. B. Schriftfamilien, Abstände, Schriftgrößen, Farben, ...) zu versehen. Mit CSS ist es möglich, die Formatierung eines Dokumentes klar von dessen Inhalt zu trennen. Die Sprache ist einfach und gut verständlich und basiert auf der gängigen *Desktop Publishing*-Terminologie.

Ein Stil ist grundsätzlich eine Regel, die dem Browser sagt, wie er ein bestimmtes tag darstellen soll. Jedes tag hat eine Anzahl von mit ihm verknüpften Eigenschaften, deren Werte definieren, wie dieses tag vom Browser angezeigt wird (Bos und Lie, 1996). Stile können für HTML- und XML-Anwendungen gebraucht werden. Zwar ist CSS eine statische Auszeichnung (*markup language*), und nicht eine Skriptsprache, aber mit Hilfe von DOM-interfaces und einer Skriptsprache wie *JavaScript* ist es möglich, CSS-Stile dynamisch zu verändern (Netscape Communications Corporation, 1999a).

Eine der fundamentalsten Eigenschaften von CSS ist es, dass die Stilvorlagen kaskadenartig verschachtelt sind (*cascading*). Ein Autor kann einem Dokument eine bevorzugte Stilvorlage zuweisen, während der Leser andere Stilvorlagen verwenden kann, die beispielsweise technische Hindernisse ausgleichen (Bos und Lie, 1996).

www.w3.org/style/css/

Es gibt drei Arten, wie ein Stil definiert werden kann: im Dokument selber, im *header* des Dokumentes oder in einem separaten Dokument. Diese Methoden können auch in einem Dokument gemischt werden, dabei werden die Definitionen der einzelnen Stile entweder zusammengeführt oder aufgrund der verschiedenen Stile neu definiert. Dabei werden kaskadenartig – sofern vorhanden – zuerst die externen Stilvorlagen vom Browser angewandt. Findet der Browser im *header* des Dokumentes weitere Stilvorlagen, die sich mit in Teilen oder vollständig mit den externen Stilvorlagen überschneiden, dann hat jeweils der Stil im *header* die höhere Priorität; ist beispielsweise die Schriftfarbe für die Titel im externen Stilvorlagendokument rot, in der *header*-Stilvorlage aber blau definiert, so werden die Titel blau dargestellt. Die höchste Priorität genießen die Stildefinitionen im Dokument (Musciano und Kennedy, 1999:278).

CSS liegt als W3C-Empfehlung inzwischen in der Version 2 vor (Bos et al., 1998). Alternativ käme die Stilvorlagensprache XSL (*extensible style sheet language*) in Frage, die ebenfalls vom W3C entwickelt wird (Sharon et al., 2000). Diese Sprache befindet sich aber in einem noch sehr frühen Stadium der Entwicklung. Aus diesem Grund wird hier nicht weiter darauf eingegangen.

6.1.6 Das Document Object Model

Das DOM (*document object model*) ist eine API (*application programming interface*, eine Schnittstelle, welche die Programmierung von Anwendungen von aussen zulässt) und repräsentiert Objekte in einem Dokument und deren Eigenschaften auf anwendungs-, plattform-, browser- und programmiersprachunabhängige Art. Es definiert die logische Struktur von Dokumenten und den Weg, wie auf ein Dokument zugegriffen und dieses manipuliert werden kann. Das DOM wird verwendet, um mit strukturierten Daten umzugehen. Mit dem DOM können Programmierer Dokumente kreieren, durch ihre Struktur navigieren, und Elemente und Inhalte hinzufügen, modifizieren oder löschen. Auf praktisch alles, was es in XML- oder HTML-Dokumenten gibt, kann zugegriffen werden (Apparao et al., 1998).

www.w3.org/dom/
www.xml101.com/dom/

Das DOM wurde so gestaltet, dass es mit jeder Programmiersprache benutzt werden kann. Zusätzlich wurde das DOM mit *JavaScript*- und *ECMAScript*-Einbindung versehen.

Im DOM haben Dokumente eine logische Struktur, die wie ein Baum (im Sinne von <Strukturmodell>) aufgebaut ist. Genauer gleicht das DOM sogar einem <Wald>, da es mehr als einen Baum enthalten kann. Das DOM spezifiziert aber nicht, dass Dokumente als Baum oder <Wald> *implementiert* sein müssen, es ist auch nicht spezifiziert, wie die Beziehung zwischen Objekten implementiert wird. Das DOM ist ein logisches Modell, das in jeder passenden Form implementiert werden kann. Eine wichtige Eigenschaft des DOM-Strukturmodells ist seine strukturelle Isomorphie: Wenn zwei unterschiedliche DOM-Implementationen für das gleiche Objekt benutzt werden, werden sie das gleiche Strukturmodell mit genau den gleichen Objekten und Beziehungen produzieren (Apparao et al., 1998).

Das DOM ist ein Objektmodell im traditionellen objektorientierten Sinn: Dokumente werden mit Hilfe von Objekten modelliert, und das Modell umfasst nicht nur die Struktur eines Dokumentes, sondern auch das Verhalten des Dokumentes und der Objekte, aus denen es besteht.

In anderen *markup languages* werden Dokumente nicht durch ein Objektmodell, sondern durch ein abstraktes Datenmodell beschrieben. In einem abstrakten Datenmodell ist das Modell um die Daten herum aufgebaut. Im Gegensatz dazu werden in objektorientierten Umgebungen wie dem DOM die Daten selbst in Objekte eingeschlossen, welche die Daten verstecken und dadurch vor direkter externer Manipulation schützen. Die Funktionen, die mit diesen Objekten verknüpft sind, bestimmen, wie die

Objekte manipuliert werden können; sie sind Teil des Objektmodells (Apparao et al., 1998).

Die vom DOM spezifizierten *interfaces*, die verwendet werden, um XML- und HTML-Dokumente zu verwalten, sind Abstraktionen. Sie sind eine Methode, einen Weg zu spezifizieren, um auf die interne Repräsentation einer Anwendung zugreifen und sie manipulieren zu können. Die *interfaces* enthalten dabei keine konkreten Implementationen. Jede DOM-Anwendung kann beliebig aussehen, solange die benutzten *interfaces* vom DOM unterstützt werden (Apparao et al., 1998).

Im DOM-*core* ist ein minimaler Set von Objekten und *interfaces* definiert, mit deren Hilfe Entwickler auf Objekte in XML- und HTML-Dokumenten zugreifen und diese manipulieren kann. Erweiternd gibt es das DOM HTML, das Objekte und Methoden beschreibt, die sich auf HTML-Dokumente beziehen (Apparao et al., 1998).

Im Wesentlichen definiert das DOM *interfaces*, die benutzt werden, um *nodes* zu verwalten. *Nodes* beschreiben Elemente, Text, Kommentare, *entity references* und alle anderen in XML-Anwendungen vorkommenden Objekte. *Nodes* werden ausserdem zur Repräsentation von Element-*attributes* verwendet, obwohl diese nicht wirklich Teil des *document tree* sind (Bradley, 2000:263).

Das *node interface* ist der zentrale Bestandteil des DOM. Eigentlich würde dieses *interface* genügen, alle anderen *interfaces* (*document*, *element*, *attr*, *entity*, usw.) dienen zur Vereinfachung. Im *node interface* sind Methoden definiert, die sich auf die *node*-Charakteristik, die relative Position eines *nodes* und die Manipulation von *nodes* beziehen (Bradley, 2000:263).

6.1.7 Interaktivität mit JavaScript

developer.netscape.com/

JavaScript ist eine plattformunabhängige, objektorientierte Skriptsprache, die von *Netscape* entwickelt worden ist. Mit *JavaScript* ist es unter anderem möglich, HTML- und XML-Anwendungen durch das DOM untereinander zu verbinden. *JavaScript* ist eine interpretierte Sprache, die eine mit *C* oder *Java* vergleichbare Syntax hat (Netscape Communications Corporation, 1999b). Mit ihr werden Programme geschrieben, die über das Internet ausgeführt werden können. *Client-side-JavaScript*-Programme werden auf dem Browser, *Server-side-JavaScript*-Programme auf dem Server ausgeführt. *JavaScript* erlaubt es, dynamische HTML- und XML-Anwendungen zu erstellen, die beispielsweise Benutzerinteraktion verarbeiten können. Z. B. können *JavaScript*-Funktionen, die direkt in HTML- oder XML-Anwendungen eingebettet sind, auf durch den Benutzer ausgelöste Ereignisse wie Mausbewegung oder Seitennavigation reagieren.

6.2 Svg im Überblick

www.w3.org/graphics/svg/
www.adobe.com/svg/
www.svgcentral.com/

SVG ist, sehr vereinfacht, eine *open-standard* Vektorgrafiksprache, mit deren Hilfe Webseiten mit hochauflösenden Grafiken und komplexen Elementen (z. B. mit Verläufen, Animation oder Filtereffekten) gestaltet werden können. Mit SVG kann Präzision und Qualität auf Webseiten gebracht werden, da die freie Wahl von Schrift, Farben und Interaktivität möglich ist. Dies schliesst die folgenden Möglichkeiten mit ein (Adobe, 2000e):

- ◆ Im Gegensatz zu den traditionellen Web-Technologien unterstützt SVG fast beliebig viele Farben, ICC-Farbprofile und Verläufe.
- ◆ Hervorragende typografische Eigenschaften wie etwa Kerning, Text entlang von Pfaden oder Ligaturen und die Möglichkeit, beliebige Schriften

einbetten zu können, erlauben professionellen Satz. Da SVG auf XML aufbaut, ist es möglich, Text innerhalb von SVG-Grafik zu suchen, zu editieren und vollständig zu internationalisieren.

- ◆ Eine SVG-Grafik kann durch dynamische Interaktivität auf Benutzerinteraktion reagieren, z. B. durch *highlighting*, *tool tips*, Spezialeffekte, Animation oder Echtzeit-Veränderungen.
- ◆ SVG ist eine textbasierte Sammlung von XML-Befehlen. Dies bedeutet, dass SVG-Anwendungen nicht nur in einem beliebigen Texteditor bearbeitet, sondern auch serverseitig durch Perl- oder CGI-Skripts dynamisch produziert werden können.
- ◆ Aufgrund des verlässlichen Farbmodells werden SVG-Grafiken mit den gleichen Farben gedruckt, die schon auf dem Bildschirm sichtbar sind.

Da SVG auf XML aufbaut, können existierende Standards wie HTML, CSS, DOM, XSL, *JavaScript* und CGI problemlos integriert werden. Weil SVG mit dem DOM übereinstimmt, ist es auch genau so durch *scripts* steuerbar wie DHTML. Dies bedeutet, dass interaktive und dynamische Effekte in HTML und SVG mit nur einem einzigen *script* gesteuert werden können.

XML beschreibt Information als strukturierte Daten, so dass Anwendungen das gleiche SVG-Bild unterschiedlich verarbeiten können; vorstellbar sind z. B. Optimierung für verschiedene Auflösungen oder reine Textdarstellungen.

6.2.1 Die Konzepte von svg

SVG-Grafik ist skalierbar, wobei man unter Skalierbarkeit eine gleichmässige Vergrösserung oder Verkleinerung versteht. Aus der grafischen Perspektive meint Skalierbarkeit, nicht auf eine einzige, fixe Pixelgrösse limitiert zu sein. Im Internet bedeutet Skalierbarkeit, dass eine bestimmte Technologie anwachsen kann auf eine grosse Zahl von Dokumenten, Anwendungen oder Benutzern. Für SVG treffen beide Bedeutungen zu (Bowler et al., 2000).

SVG-Grafik kann einerseits in einem reinen SVG-Viewer betrachtet werden. Eine wichtige Eigenschaft von SVG ist es aber, dass SVG-Grafik als ein Modul innerhalb einer grösseren XML-Anwendung mit verschiedenen Modulen (*multi-namespace XML application*) angewandt werden kann. Dadurch kann der Nutzen der einzelnen verwendeten Module vervielfacht werden. Beispielsweise könnte SVG-Grafik Bestandteil eines wissenschaftlichen Dokumentes sein, das ausserdem das XML-Modul MathML (für mathematische Ausdrücke in einer XML-Anwendung) verwendet.

Ein allgemein gültiges `<path>`-Element wird benutzt, um alle möglichen grafischen Objekte als Pfade zu zeichnen. Zusätzlich gibt es für die gängigsten geometrischen Figuren wie Rechtecke oder Ellipsen vordefinierte Elemente. Gut kontrollierbar sind in SVG das Koordinatensystem, in welchem die grafischen Objekte definiert sind, und die Transformationen (z. B. Verschiebung, Skalierung, Rotation), die während dem Rendering verwendet werden.

SVG kennt keine vordefinierten Symbole. Dafür ist es möglich, eigene Symbole zu kreieren, diese mehrmals pro Dokument zu benutzen oder sie mit anderem Dokumenten zu teilen, ohne auf eine zentrale Aufbewahrungsstelle angewiesen zu sein. Sämtliche Stile von Symbolen inklusive Grösse und Orientierung sind natürlich jederzeit kontrollierbar.

SVG-Grafik als Modul einer grösseren XML-Anwendung

6.2.2 Das Rendering-Modell

SVG benützt ein *painters model* zum Rendern; dies bedeutet, dass jede Operation über eine gewisse Fläche des Ausgabemediums malt. Sollte die neu gemalte Fläche eine bereits bestehende überlappen, so wird die neue die alte abdunkeln oder gar vollständig überdecken, je nach Grad der Flächenopazität. Gerendert wird dasjenige Element zuerst, das in der SVG-Anwendung an erster Stelle steht. Dieses Element ist folglich zuunterst (Bowler et al., 2000).

SVG unterstützt drei Haupttypen von renderbaren Grafikobjekten:

- ◆ Shapes (Kombinationen von geraden Linien und Kurven)
- ◆ Text
- ◆ Rasterbilder

Shapes und Text können gefüllt sein und eine Kontur haben, wobei Füllung und Kontur je eigene Opazitätseinstellungen haben können. Füllungen und Konturen können mit einer Farbe versehen sein oder Verläufe oder Muster haben.

6.2.3 Struktur eines SVG-Dokumentes

Ein SVG-Dokument besteht aus einer Anzahl von SVG-Elementen, die sich zwischen den *tags* `<svg>` und `</svg>` befinden. Es kann entweder nur ein einzelnes Element, z. B. ein Rechteck, enthalten, oder sich aus einer komplexen, ineinander verschachtelten Sammlung von Grafikelementen zusammensetzen. Ein SVG-Dokument kann entweder für sich selbst stehen oder als ein Fragment in eine XML-Anwendung eingebettet sein (*in line*).

Analog zu den aus Grafikprogrammen bekannten Ebenen gibt es die Möglichkeit, Objekte in Gruppen zusammenzufassen und diese zu benennen. Dies geschieht mit dem `<g>`-Element. Eine Zusammenfassung zu Objektgruppen ist dann sinnvoll, wenn mehrere Objekte die gleichen Objektattribute haben. Praktisch ist die Möglichkeit, hierarchische Objektgruppen zu bilden: Mehrere Objektgruppen können wiederum zu einer Gruppe zusammengefasst werden.

Eine wichtige Bedeutung kommt URI-Referenzen (*uniform resource identifiers*) zu anderen Objekten zu. Wenn beispielsweise ein Rechteck mit einem Verlauf gefüllt werden soll, so wird zuerst ein `<linearGradient>`-Element definiert und mit einer ID versehen. Danach kann der Verlauf als *attribute* der Rechtecksfüllung mittels ID referenziert werden.

Mit dem `<symbol>`-Element und dem `<defs>`-Element ist es möglich, grafische Vorlageobjekte zu definieren, deren Instanzen später mit dem `<use>`-Element beliebig oft verwendet werden können. Der grosse Vorteil liegt darin, dass dem Dokument mehr Struktur und Übersicht verlieht werden kann; solche Dokumente sind leichter verständlich und überschaubarer. Die `<symbol>`- und `<defs>`-Elemente selber werden nicht gerendert, sie dienen nur als Vorlage für ihre Instanzen.

Mit dem `<use>`-Element können nicht nur Instanzen von `<symbol>`-Elementen wieder verwendet werden, sondern es können auch `<g>`- und sogar ganze `<svg>`-Elemente instantiiert werden. Im Gegensatz zu `<symbol>`- und `<defs>`-Elementen wird aber auch das Original gerendert.

Ferner ist bedingtes Rendern möglich. Mit dem `<switch>`-Element und *attributes*, welche die Systemsprache und die Systemeigenschaften betreffen (*system-language*, *system-required*) kann die Grafikausgabe spezifiziert werden.

Bei allen SVG-Elementen kann das *attribute* `id` verwendet werden. Es dient dazu, einzelne Elemente mit einem beliebigen Namen eindeutig zu identifizieren. Genau so steht auch das *attribute* `class` stets zur Verfügung.

Def. Objektgruppen

URI-Referenzen

`<symbol>`, `<defs>`, `<use>`

`<switch>`

`id`, `class`

Mittels `class` können mehrere Elemente zu einer Klasse zusammengefasst und so gemeinsam angesprochen werden.

Zur Steuerung eines SVG-Dokumentes existiert eine Anzahl von DOM-Interfaces, die jeweils eigene Attribute und Methoden haben. Beispielsweise kann mit der Methode `getElementById` ein Element mit der festgelegten ID zurückgegeben oder mit `pauseAnimation` eine gerade laufende Animation gestoppt werden.

6.2.4 Styling Properties

Styling properties definieren, wie die Grafikelemente im SVG-Inhalt gerendert werden sollen. Grosse Bedeutung hat *styling* einerseits bei allem, was quasi in der Natur sichtbar ist (Füllfarbe, Strichfarbe, Strichstärke, usw.), andererseits bei *text styling* wie Schriftfamilie und Schriftgrösse. Bis auf wenige, SVG-spezifische Zusätze, verwendet SVG die *styling properties* von CSS (bzw. CSS 2) und XSL.

CSS können extern definiert und im aktuellen Dokument referenziert sein. Andererseits können sie auch intern mit Hilfe des `<style>`-tag oder gar *in line* mit einem Stilattribut definiert sein. `<style>`

6.2.5 Koordinatensysteme und Transformationen

Der Ort, wo auf dem Ausgabemedium der SVG-Inhalt gerendert wird, heisst *canvas*. Der *canvas* ist im Prinzip unendlich gross, aber eine Grafik wird relativ zu einem endlich dimensionierten, rechteckigen Ausschnitt der Fläche gerendert. Dieser Ausschnitt wird *viewport* genannt und entspricht der für die Benutzerin sichtbaren Fläche. Der *viewport* verfügt über ein initiales Koordinatensystem, das mit Transform-Funktionen (Skalierung, Verschiebung, Rotation, Neigung) verändert werden kann.

6.2.6 Path-Elemente

Path-Elemente repräsentieren die Kontur einer Form, die gefüllt und/oder mit einer Linie versehen werden kann. Ein *path* ist also die Geometrie eines Objektes.

Ein *path* ist mit einem `<path>`-tag gekennzeichnet, das ein `d='(path data)'`-attribute enthält. *Path data* kann aus verschiedenen Instruktionen wie *moveto* (M, m), *line* (L, l), *closepath* (z) bestehen, gefolgt von den jeweiligen Koordinaten. Ein Grossbuchstabe bedeutet, dass die Koordinate absolut zu verstehen ist, ein Kleinbuchstabe steht für relative Koordinaten. Ein Beispiel könnte so aussehen: `<path>`

```
<path d='M357.603,199.002l-0.497-0.313v0.728l0.497-0.414z'/>
```

Mit ebensolchen Definitionen lassen sich auch Bézierkurven zeichnen. Zusätzlich zu `<path>` stehen ausserdem die tags `<rect>`, `<circle>`, `<ellipse>`, `<line>`, `<polyline>`, `<polygon>` bereit.

6.2.7 Text-Elemente

Text wird durch das `<text>`-tag identifiziert. Textelemente werden wie andere Grafikelemente auch gerendert. Dies bedeutet, dass Koordinatensystem-Transformationen, Farben usw. auf Textelemente genauso angewandt werden können, wie auf andere Objekte auch. `<text>`

Jedes Textelement muss einzeln gerendert werden. SVG unterstützt keinen automatischen Zeilenumbruch, dies bedeutet, dass Mehrzeilertext schon vom Autor auf Zeilen verteilt werden muss.

‹display›, ‹visibility›

Textelemente können einem Pfad entlang laufen. Dabei kann der Autor entscheiden, ob der Text horizontal oder vertikal, von links nach rechts oder von rechts nach links läuft.

6.2.8 Kontrolle über die Sichtbarkeit

Mit ‹display› und ‹visibility› bietet SVG zwei Möglichkeiten, die Sichtbarkeit grafischen Inhaltes zu kontrollieren. Beide Möglichkeiten können ein Element unsichtbar machen. Der wichtige Unterschied liegt darin, dass ein Element mit der Eigenschaft ‹display:none› behandelt wird, als wäre es nicht vorhanden, während ein Element mit ‹visibility:hidden› genau gleich abgearbeitet wird, wie wenn es sichtbar wäre, ohne aber dargestellt zu werden.

6.2.9 Interaktivität

SVG-Anwendungen können interaktiv sein, z. B. durch Reaktion auf Benutzereingaben. Die folgenden Eigenschaften werden in SVG genutzt:

- ◆ Benutzerinitiierte Aktionen wie Mausklicks oder Tastatureingaben können Animationen auslösen oder *scripts* starten.
- ◆ Die Benutzerin kann Hyperlinks zu anderen Webseiten folgen.
- ◆ Oft ist es dem Benutzer möglich, in der Grafik ein- und auszuzoomen.
- ◆ Die *Cursor*-Form kann sich in Abhängigkeit von der Position verändern.

6.3 Svg für Bildschirmkarten

www.carto.net/

Das Potenzial von SVG für die Bildschirmkartografie ist enorm. Dinge, die bisher schon denkbar waren, sind mit SVG meist unvergleichbar einfacher und eleganter zu lösen. Eine ganze Reihe neuer Möglichkeiten kommt hinzu. Einige Autoren haben heute bereits Anwendungen entwickelt, die auf SVG basieren, ein gutes Beispiel ist etwa die interaktive Karte von Wien (Neumann, 2000, vgl. Abb. 13).

Vorstellbar sind auf der Basis von SVG die in den nächsten Abschnitten geschilderten Ideen.

6.3.1 Dynamische Kartenproduktion

Es ist möglich (wenn zur Zeit von *Adobe* auch noch nicht implementiert), eine SVG-Anwendung vollständig dynamisch serverseitig zu produzieren. Bei einem webbasierten Atlas können auf diese Art schnell und einfach Kärtchen mit aktuellen Informationen produziert werden, gegebenenfalls sogar vollautomatisch aus den Angaben einer Tabellenkalkulation.

6.3.2 Mediumsorientierte Ausgabe

Je nach Medium kann die Ausgabe anders erfolgen. Stellt man sich etwa die Karte aller Hotels einer Stadt vor, so kann mit der gleichen SVG-Anwendung eine farbige Karte für Webbrowser, eine einfarbige Karte für den Ausdruck auf Papier und ein Verzeichnis aller Hotels in Listenform für WAP-Handys produzieren. Die SVG-Anwendung wird dabei nicht verändert, sondern nach Medien aufgeschlüsselt unterschiedlich interpretiert.

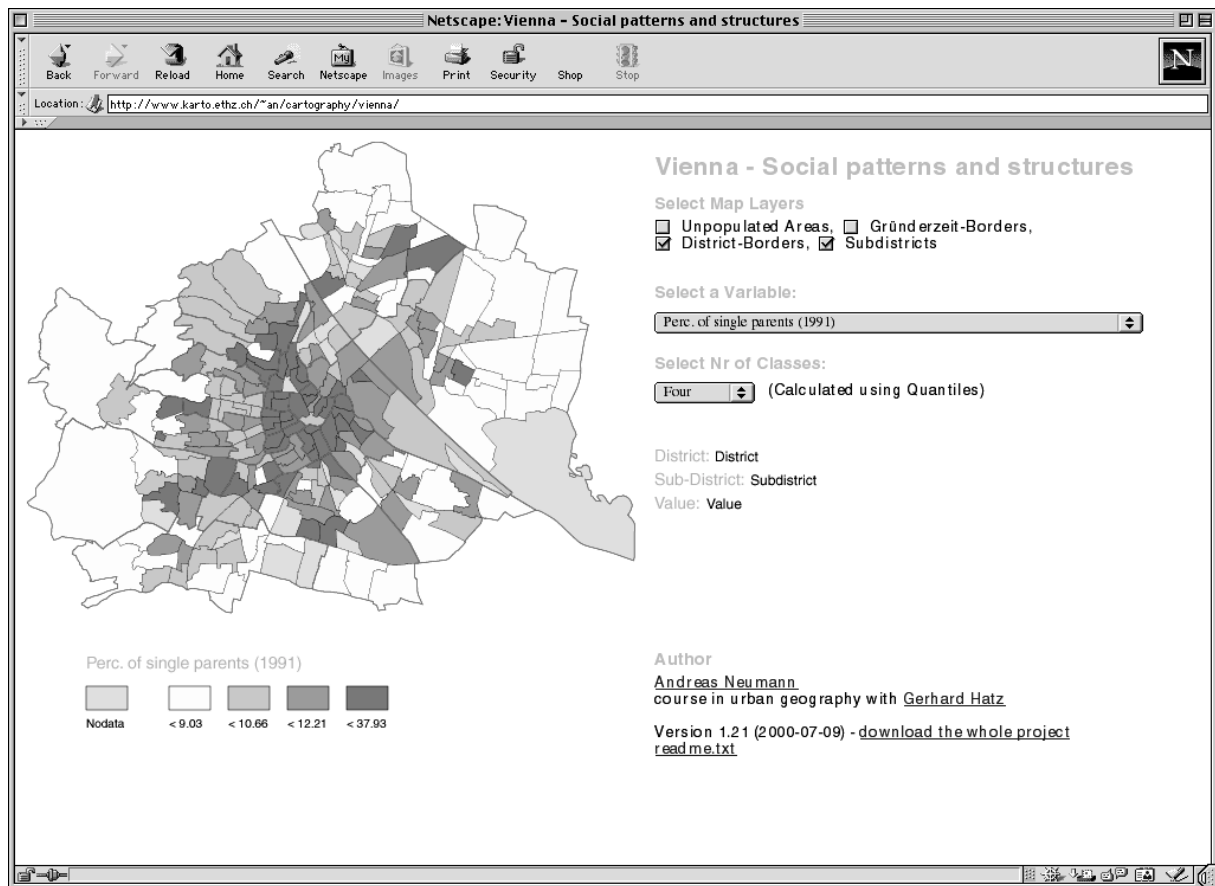


Abb. 13 www.karto.ethz.ch/~an/cartography/vienna/

6.3.3 Internationalisierung

Eine ähnliche Methode lässt die Anpassung einer Karte auf die jeweilige Landessprache zu. In der gleichen SVG-Anwendung werden mehrere Sprachversionen abgespeichert. Je nachdem, welche Sprache beim Browser des Benutzers eingestellt ist, erfolgt eine andere Anzeige.

6.3.4 Dynamische Legende

Anstelle der üblichen Legende, die sich irgendwo am Bildschirmrand befindet und unnötig den sowieso beschränkten Platz noch mehr reduziert, kann mit relativ einfachen Mitteln eine dynamische Legendenfunktion geschaffen werden. Abhängig von der Position des Cursors könnte im Sinne von *tool tips* die Bedeutung der Signatur eingeblendet werden, möglicherweise auch ergänzt mit Zusatzinformationen.

Neben deutlichem Platzgewinn sorgt diese Methode auch für mehr Klarheit und Übersichtlichkeit. Etwa eine geologische Karte, die naturgemäss eine sehr umfangreiche Legende hat, kann einzelne Gesteine bei Mausrollover einer Fläche erklären und gleichzeitig ein Foto des Gesteins zeigen.

6.3.5 Interaktive und dynamische Stilveränderungen

Da Objekte und Stile vollständig voneinander getrennt sind, können auf bequeme Art Stile wie Flächen- und Strichfarbe, Strichstärke, Strichart, Schriftgrösse usw. interaktiv oder dynamisch verändert werden. Die Benut-

zerin kann z. B. die von ihr bevorzugten Farben wählen oder die Strichstärke verändern. Beim Ein- bzw. Auszoomen ist eine dynamische Stilveränderung hilfreich. Diese Stärke von SVG ist für adaptives Zoomen von grossem Vorteil und wird im nächsten Kapitel ausführlich behandelt.

7 Interaktive Massstabsveränderungen durch adaptives Zoomen

Das letzte Kapitel hat gezeigt, dass das Potenzial von interaktiver Kartografie im Zusammenspiel mit XML bzw. SVG sehr gross ist. Am Beispiel der interaktiven Massstabsänderung wird sich dieses Kapitel genauer mit einigen der neuen Möglichkeiten auseinandersetzen. Auf den nächsten Seiten werden diejenigen Konzepte entworfen und ausgearbeitet, die für die Funktionalität einer Bildschirmkarte, deren Massstab interaktiv verändert werden soll, notwendig sind.

Obwohl versucht wird, die Konzepte unabhängig von der Art der Implementation zu halten, können sich einzelne Lösungen dennoch ausdrücklich auf SVG beziehen, da sie zum Teil Stärken von SVG direkt ausnutzen oder Schwächen überbrücken.

Von der Implementation unabhängige Konzepte

7.1 Hintergrund

Interaktive Massstabsänderung kann etwas allgemeiner auch als *Zoomen* bezeichnet werden. *Zoomen* ist ein Konzept, das mit der Sicht aus einem Flugzeug verglichen werden kann. Sinkflug bedeutet bessere, Steigflug schlechtere Detailsichtbarkeit. Durch *Zoomen* wird eine kleinere Fläche eines Bildes mit einer höheren Vergrösserung (einzoomen) bzw. eine grössere Fläche mit einer kleineren Vergrösserung (auszoomen) gezeigt. Bei optischen Systemen bedeutet Einzoomen eine Verbesserung der Sichtbarkeit von Details eines Bildes. Bei Computersystemen wird hingegen normalerweise das Bild ohne Veränderung der Detailsichtbarkeit vergrössert, da die Anzeige auf das limitiert ist, was im Bild gespeichert ist (Lexico.com, 1997).

Def. Zoomen

Aus naheliegenden Gründen kann alleiniges Einzoomen ohne Verbesserung der Detailsichtbarkeit für interaktive Karten nicht genügen. Es existieren daher von verschiedenen Autoren, vor allem aus dem Bereich der geografischen Informationssysteme (GIS), zumindest Ansätze von Lösungen.

7.1.1 Abgrenzung von grafischen GIS-Produkten und interaktiven Karten

Auf den ersten Blick scheinen grafische GIS-Produkte und interaktive Karten fast dasselbe zu sein; eine enge Verwandtschaft lässt sich auch nicht übersehen. Dennoch gibt es klare Eigenschaften, die helfen, beide voneinander abzugrenzen.

GIS sind gut dazu geeignet, räumliche Daten zu sammeln, speichern, überprüfen, integrieren, manipulieren, analysieren oder anzuzeigen (Lexico.com, 1997). Im Vordergrund stehen die *Organisation* und die *Exploration* räumlicher Daten, wobei die Anzeige eine von vielen Möglichkeiten von GIS ist. Im Gegensatz dazu ist das Ziel der kartografischen Produkte die *Vermittlung* räumlicher Daten. Die Daten können dabei unter anderem aus einem GIS kommen; das GIS übernimmt dann die Rolle eines Datenlieferanten.

Interaktive Karten zur Vermittlung räumlicher Daten

GIS können in der Regel als Expertensysteme betrachtet werden. Die Bedienung von GIS und die Analyse der Resultate setzt eindeutig Fachkenntnisse voraus. Ein ganz anderes Zielpublikum versucht man hingegen mit interaktiven Karten zu erreichen. Von der Benutzerin dürfen keine Fachkenntnisse erwartet werden; es wird versucht, bereits bekannte spatiale Informationen zu visualisieren und der Benutzerin die Resultate auf übersichtliche und ansprechende Art zu präsentieren.

Inhalt einer interaktiven Karte durch den Autor vorbestimmt

Das Mittel «Interaktion» dient nicht der Analyse, der Manipulation oder der Schaffung neuer Daten, sondern einzig, um komplexe Abläufe verständlicher darstellen zu können oder eine grosse Menge von Information in visuell erfassbare Teile zu gliedern.

Während die Anwenderin von GIS selber bestimmen kann, was er erfahren möchte und wie die Resultate visualisiert werden sollen, verhält sich eine interaktive Karte eher wie ein Film oder gar wie ein Computerspiel. Was die Benutzerin zu sehen bekommt, ist vom Autor vorbestimmt. Der Benutzerin bleiben – je nach Art der Implementation – vor allem die Entscheidungen, ob, wann und wie detailliert sie über einzelne Informationen Bescheid wissen möchte.

7.1.2 Masstabsveränderungen in grafischen GIS-Produkten

Die zunehmende Verbreitung von GIS im Laufe der letzten Jahre weckte das Bedürfnis nach einer Funktion, mit deren Hilfe grafische GIS-Produkte in beliebigen Masstäben ausgegeben werden können. Alleinigiges Skalieren der GIS-Daten genügt nicht, wie bereits weiter oben dargelegt wurde.

Einige Autoren gehen von einem sogenannten *masstabsfreien Grunddatensatz* aus, von dem mit Hilfe automatischer oder halbautomatischer kartografischer Generalisierungsmethoden Karten beliebiger Masstäbe abgeleitet werden können: «It must be possible to derive display products from a basic database at arbitrary scale or symbolization, and to maintain good readability» (Weibel, 1995:56).

Andere Autoren mit eher kartografischem Hintergrund sind der Überzeugung, dass «der sogenannte masstabsfreie Grunddatensatz in der Kartografie eine Illusion» sei (Spiess, 1996a:468): «Erfahrungsgemäss kann ein Datensatz nur einen gewissen Masstabsbereich abdecken. (...) Eine vollautomatisierte Generalisierung bis in kleinste Masstäbe ist vorläufig noch unrealistisch.»

Eine dritte Gruppe von Autoren geht den Mittelweg und schlägt ein Modell vor, in dem kartografische Daten in mehreren hierarchischen Masstäben gespeichert werden: «Objects are stored with increasing detail and can be used to compose a map at a particular scale» (Timpf, 1997).

7.1.3 Masstabsveränderungen in kartografischen Produkten

Interaktive Karten sollen ihre Daten aus verschiedenen Quellen beziehen können, mitunter auch aus einem oder mehreren GIS. Schon aus diesem Grund ist das Konzept eines masstabsfreien Grunddatensatzes für interaktive Karten nicht geeignet: Die Beschränkung auf einen einzelnen masstabsfreien Grunddatensatz würde bedeuten, dass alle Daten in der gleichen Form, mit der gleichen Qualität und im gleichen Dokument vorhanden sein müssten. Einerseits ist der Aufwand, einen solchen Datensatz zu erstellen, für interaktive Karten unverhältnismässig gross, andererseits mangelt es dieser Lösung auch an der nötigen Flexibilität, vor allem in Bezug auf die Ergänzung weiterer Daten, die nicht in einer normierten Form vorhanden sind. Nebenbei bemerkt ist in diesem Bereich die Forschung auch noch nicht weit genug fortgeschritten, dass eine realistische Anwendung bereits heute möglich wäre.

Ein anderer Ansatz lässt deutlich mehr Flexibilität zu, ist viel einfacher in der Handhabung und praxisnaher. Ausserdem lässt er sich mit den heute zur Verfügung stehenden Mitteln relativ gut umsetzen. Dieser Ansatz geht davon aus, dass ein kartografischer Datensatz zwar nicht masstabsfrei ist, sich aber in gewissen Grenzen gut skalieren lässt (Spiess, 1996a:468). So gesehen deckt jeder Datensatz einen bestimmten Masstabsbereich ab. Stehen

Kartografische Datensätze lassen sich in gewissen Grenzen gut skalieren

nun mehrere aufeinander abgestimmte kartografische Datensätze zur Verfügung, deren Massstabbereiche sich ergänzen, so ist relativ schnell ein recht grosser Zoombereich möglich.

7.2 Konzept und Definitionen

Der im Folgenden beschriebene Vorschlag, interaktive Massstabänderung zu ermöglichen, geht von der Idee aus, kartografische Datensätze in gewissen Grenzen zu skalieren und sie dabei an den jeweiligen Massstab anzupassen. Dieses Vorgehen kann als *adaptives Zoomen* bezeichnet werden:

Ein Zoomprozess ist dann adaptiv, wenn sich die Darstellung einer Karte so an den Kartenmassstab respektive den Zoomfaktor anpasst, dass die Kartenqualität immer als gut empfunden wird. Dies gilt insbesondere für den Informationsgehalt, die Informationsdichte und die grafische Darstellung aller Signaturen.

Eine *adaptive Karte* ist eine spezielle Form einer interaktiven Karte, die sich obiger Definition entsprechend adaptiv an einen wählbaren Massstab anpasst. Die adaptive Karte besteht notwendigerweise immer aus der Kartengeometrie und einem Programm, das die Steuerung der notwendigen Prozesse übernimmt.

Die Kartengeometrie heisst *adaptiver Datensatz*. Dieser umfasst sämtliche von der Anwendung benötigte Objekte und besteht aus einem Dokument (siehe Abb. 14).

Im adaptiven Datensatz enthalten sind mindestens ein, besser aber mehrere sich massstäblich ergänzende Kartenteile nebeneinander. Solche Kartenteile werden *Massstabbereichs-Geometriesätze* oder einfach *Geometriesätze* genannt. Jeder dieser Geometriesätze kann pro Objektgruppe verschiedene Dichtestufen haben, z. B. kann die Objektgruppe Flüsse in den Dichtestufen «locker», «mittel», «dicht» vorkommen.

Beim Zoomprozess wird, ausgehend von der äussersten Zoomstufe bzw. dem kleinsten Massstab, ein Geometriesatz kontinuierlich so lange vergrössert, bis sein Vergrösserungslimit erreicht ist. Gleichzeitig mit der Vergrösserung des Massstabs müssen neue Dichtestufen einzelner Objektgruppen hinzukommen, und zwar so, dass die totale Informationsdichte konstant bleibt. Wenn das Vergrösserungslimit erreicht ist, wird der Geometriesatz aus- und ein neuer, auf den anschliessenden Zoombereich optimierter Geometriesatz eingeblendet.

Um die Qualität der kartografischen Darstellung beibehalten zu können, müssen *Objektstile* wie Strichstärke, Signaturgrösse und Schriftgrösse (zusammengefasst unter dem Oberbegriff *adaptive Signaturmasse* oder einfach nur *Signaturmasse*, da alle Signaturmasse adaptiv sind) bei Vergrösserung des Massstabs in der Dimension relativ zu ihrer Umgebung kleiner gewählt werden können als im Vorläufermassstab. Die Signaturmasse müssen jeweils so bemessen sein, dass kartografische Regeln wie Mindestabstand, Mindestgrösse oder auch Schriftplatzierung eingehalten werden können.

Ein harmonisches Darstellungsbild über mehrere Massstäbe hinweg kann nur entstehen, wenn die Unterschiede von einer Zoomstufe zur nächsten gering und unauffällig sind. Veränderungen in der Darstellung von Signaturmassen und in der Kartengeometrie müssen also entsprechend vorsichtig gewählt werden.

Def. Adaptives Zoomen

Def. Adaptive Karte

Def. Adaptiver Datensatz

Def. Geometriesatz

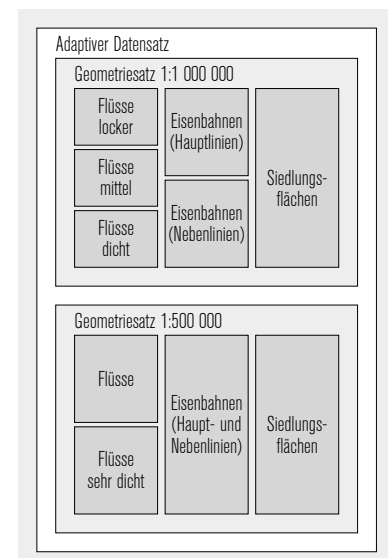


Abb. 14 Schematischer Aufbau eines adaptiven Datensatzes

7.3 Projektspezifische Ziele

Als Ergebnis des hier beschriebenen Projektes wird eine webbasierte interaktive Karte entstehen, in welcher der Benutzer in gewissen Grenzen einen Massstab frei wählen kann. Jede der darstellbaren Karten soll für den gewählten Massstab und auf dem Ausgabemedium Bildschirm eine optimale Informationsdichte bei guter grafischer Darstellung unter Berücksichtigung der kartografischen Grundsätze haben. Die Karte wird möglichst viele kartografisch repräsentative Objekte enthalten und soll dadurch als Basis für vergleichbare Projekte dienen.

7.3.1 Kartografische Ziele

Neben den Grundsätzen zur Gestaltung von Karten (vgl. Kapitel 4.2 «Grundsätze zur Kartengestaltung» ab Seite 17) gibt es einige projektspezifische kartografische Ziele, die erfüllt werden sollen.

1. Die adaptive Karte soll gemäss Kapitel 4.5 «Bildschirmspezifische Vergrößerung der Kartengeometrie» ab Seite 24 vergrössert werden; sowohl der technische, als auch der kartografische Vergrößerungsfaktor sollen berücksichtigt werden.
2. sowohl der kartografische, als auch der technische Vergrößerungsfaktor sollen berücksichtigt werden.
3. Bei jedem Geometriesatz muss das Steuerprogramm für die adaptive Karte entscheiden können, ob er für den gewählten Massstab benötigt wird oder nicht.
Es können pro gewähltem Massstab immer nur Objektgruppen dargestellt werden, die dem gleichen Geometriesatz angehören.
4. Bei allen Objektgruppen muss das Steuerprogramm für die adaptive Karte entscheiden können, ob sie für den gewählten Massstab benötigt werden oder nicht.
Je mehr eingezoomt wird, desto mehr Objektgruppen müssen eingeblendet werden können, um die Informationsdichte konstant zu halten. Dies bedeutet, dass es möglich sein muss, jede Objektgruppe für sich anzusprechen.
5. Die Stile aller Objekte müssen zu jeder Zeit gut dimensioniert sein. Die Signaturmasse, also Stile wie Strichstärken (Strassen, Flüsse, Seekonturen, usw.) und Punktsignaturgrößen (Ortssignaturen, usw.), aber auch Schriftgrößen (Ortsbeschriftungen, usw.) müssen sich dynamisch dem Massstab anpassen, weil sie ohne Anpassung beim Einzoomen ungewollt gross würden.
6. Die Platzierung der Beschriftung muss zumindest teilweise dynamisch erfolgen, weil sich durch den Zoomvorgang die Schriftgrösse und somit auch die Textbreite und die Schrifthöhe ändern.

7.3.2 Funktionelle Ziele

Für die Qualität von interaktiven Karten sehr wichtig ist eine gut durchdachte Bedienung. Im Fall dieses Projektes beschränkt sich die Bedienung im Wesentlichen auf eine Möglichkeit, den Massstab zu verändern.

1. Der Benutzer soll mit einem möglichst einfach zu bedienenden Werkzeug selber einen Massstab festlegen können. Auf eine numerische Eingabe des Massstabs wird verzichtet, da dies vor allem für ungeübte Benutzer zu abstrakt ist. Ein Werkzeug im Sinne eines Schiebereglers, der die Höhe über Grund symbolisiert, kann geeignet sein.

2. Dem Benutzer soll es möglich sein, mit einer Lupe Ausschnitt und Vergrößerung direkt in der Karte zu bestimmen.
3. Der aktuelle Massstab der Karte soll jederzeit ablesbar sein, einerseits grafisch in einem sinnvollen Grössenvergleich, andererseits nummerisch.
4. Das Zoomzentrum soll durch den Benutzer festgelegt werden können. Per Default soll die Karte zentriert eingezoomt bzw. ausgezoomt werden, wenn der Benutzer aber ein anderes Zoomzentrum vorzieht, soll er das neue Zentrum selber beliebig wählen können.
5. Der aktuell gewählte Kartenausschnitt muss ausgedruckt werden können, wobei sowohl Farben als auch Massstab für den Druck optimiert sein sollen.

7.3.3 Inhaltliche Ziele

Folgende Objektgruppen sollen in der Karte enthalten sein:

Klasse	Objektgruppen
Flächenobjekte ohne Kontur	Siedlungsflächen
Flächenobjekte mit Kontur	Seen
Lineare Objekte	Strassennetz Eisenbahnnetz Flussnetz Landesgrenzen
Punktsignaturen	Ortschaften in mehreren Klassen
Beschriftungen	Ortschaften
Relief	

Tab. 5 Objektgruppen, die in der Karte enthalten sein sollen

7.3.4 Technische Ziele

Da mit interaktiven Karten ein breites Publikum angesprochen werden soll, müssen nicht nur Kartografie und Benutzeroberfläche strengen Anforderungen genügen. Auch die technischen und finanziellen Hürden, eine solche Karte zu produzieren und zu benutzen, dürfen nicht zu gross sein.

1. Die interaktive Karte soll weitestgehend plattformunabhängig, d. h. auf möglichst vielen Plattformen und unter allen verbreiteten Betriebssystemen funktionsfähig sein.
2. Die Ladezeit via Internet soll kurz sein.
3. Die Applikation soll so flexibel aufgebaut sein, dass sie ohne grossen Aufwand auch auf andere, ähnlich aufgebaute kartografische Datensätze angewandt werden kann.
4. Die Applikation und der adaptive Datensatz sollen so beschaffen sein, dass Wartung und Aktualisierung möglichst leicht und schnell vorgenommen werden können.
5. Alle Elemente der Applikation sollen mit kommerzieller und auch bezahlbarer Software entwickelt werden können. Softwareeigenentwicklungen sollen möglichst vermieden werden.
6. Die Entwicklung der Applikation soll mit einem vertretbaren Zeitaufwand erfolgen können. Der im Grafikprogramm notwendige Zusatzauf-

wand, um aus einer bereits bestehenden Karte eine adaptive Karte zu machen, muss daher möglichst gering sein.

7. Die Wahl der Flächen- und Linienfarben, der Signaturmasse wie Strichstärken und Grössen von Punktsignaturen und aller anderer sichtbarer Stile muss bereits im Grafikprogramm möglich sein, d. h. der kartografische Datensatz muss in WYSIWYG erstellt werden können.

7.4 Der adaptive Datensatz

Die Grundlage für die Funktionalität bildet der für eine adaptive Karte notwendige Datensatz. Seiner kartografischen und technischen Qualität kommt deshalb auch grosse Bedeutung zu. Je besser er ausgearbeitet, je konsistenter sein Aufbau ist, desto harmonischer läuft die Massstabsveränderung ab.

7.4.1 Kartografische Qualität

Neben der selbstverständlichen Berücksichtigung aller kartografischen Regeln ist besonders wichtig, dass der adaptive Datensatz konsistent und mit hoher Qualität generalisiert wurde. Der Datensatz sollte so flexibel generalisiert sein, dass er über einen grossen Massstabsbereich immer als geeignet bezeichnet werden kann.

7.4.2 Technische Qualität

Ein passender adaptiver Basisdatensatz muss Geometriesätze verschiedener Massstäbe umfassen. Diese Geometriesätze enthalten jeweils alle Objektgruppen des gleichen Massstabes und eignen sich über einen bestimmten Massstabsbereich; auf sie kann später je nach Massstab zugegriffen werden. Da die Berechnungen für die Darstellung neuer Massstäbe später von einem Grundmassstab aus gehen, müssen sämtliche Geometriesätze auf einen Massstab – den Grundmassstab – normalisiert sein.

Die einzelnen Geometriesätze unterschiedlicher Ausgangsmassstäbe sollten aufeinander abgestimmt sein und sich gegenseitig ergänzen. Besonders wichtig ist, dass es für alle Objekte, die in einem Geometriesatz vorkommen, im nächstgrösseren Massstab korrespondierende Objekte gibt. Der umgekehrte Fall gilt aber nicht: Objekte eines Geometriesatzes können zwar, müssen aber im nächsten kleineren Massstab nicht vorhanden sein.

Der adaptive Basisdatensatz muss so generalisiert sein, dass die Abhängigkeit von Signaturmassen sich auf ein Minimum reduziert: Grosse Toleranz gegenüber verschiedenen Stilgrössen ist sehr wichtig, da die Stile dynamisch behandelt werden.

Objekte der gleichen Art und mit gleichen Stilen müssen konsequent zu Objektgruppen zusammengefasst werden, da später nur noch ganze Objektgruppen und nicht Einzelobjekte angesprochen werden sollen.

Der adaptive Basisdatensatz muss ausserdem gemäss Kapitel 4.5 «Bildschirm-spezifische Vergrösserung der Kartengeometrie» auf Seite 24 um den Faktor 2,4 vergrössert sein, um eine geeignete Darstellung für die Bildschirm- ausgabe zu gewährleisten.

7.4.3 Auswahl des richtigen Geometriesatzes

Wenn durch Benutzerinteraktion die Karte in einem neuen Massstab dargestellt werden soll, dann wird durch die Applikation derjenige Geome-

Gut aufeinander abgestimmte Geometriesätze

triesatz gewählt, dessen Generalisierung für den aktuellen Massstab am besten geeignet ist.

Die Auswahl eines Geometriesatzes kann anhand von Grenzwerten geschehen, die für jeden Geometriesatz bestimmen, wie stark er skaliert werden darf. Gemäss Spiess liegen diese *Skalierungsgrenzwerte* bei ca. 0,5 und 2 (Spiess, 1996a, d. h. eine maximale Verkleinerung auf 50% des Ursprungsmassstabs bzw. eine maximale Vergrößerung auf 200% des Ursprungsmassstabs ist möglich. Die Skalierungsgrenzwerte können aber in Abhängigkeit von der Qualität der Generalisierung und der Komplexität der Darstellung bei unterschiedlichen Geometriesätzen stark variieren.

Da vor allem die formale Erfassung der Qualität der Generalisierung, aber auch der formale Ausdruck der Darstellungskomplexität nicht ohne weiteres möglich sind, können die Skalierungsgrenzwerte nicht auf einfache Art mathematisch erfasst werden. Sie beruhen auf Erfahrungswerten und müssen empirisch festgelegt werden.

7.4.4 Zeitlich versetztes Auswechseln von Objektgruppen

Weil ein harmonisches Bild mit sanften Übergängen zwischen verschiedenen Massstäben für eine adaptive Karte nötig ist, liegt die Idee nahe, anstatt eines ganzen Geometriesatzes aufs mal nur einzelne Objektgruppen auszuwechseln. Statt eines eindeutigen und möglicherweise sichtbaren Schnittes beim Wechsel des Geometriesatzes würde ein sanfterer Übergang von einem Geometriesatz zum nächsten entstehen.

Leider ist diese Lösung nicht realisierbar, weil sowohl Generalisierung als auch Wahl der Signaturen und Signaturmasse stark vom Kontext abhängt. Eine Linie, die innerhalb eines Geometriesatzes gut generalisiert zu sein scheint, passt innerhalb eines anderen Satzes normalerweise schon geometrisch nicht mehr in die umgebende Geometrie.

7.5 Informationsdichte und Informationsgehalt

Mit grösser werdendem Massstab vergrössert sich auch die zur Verfügung stehende Fläche, wenn, wie in Kapitel 7.2 «Konzept und Definitionen» auf Seite 45 kurz angetönt und weiter unten in Kapitel 7.6 «Anpassung der Signaturmasse: Liniensignaturen» ab Seite 51 detailliert beschrieben, die Signaturmasse relativ zur Umgebung verkleinert werden. Diese Fläche kann mit neuen und zusätzlichen Informationen gefüllt werden. Die Begriffe Informationsdichte und Informationsgehalt beschreiben den Sachverhalt um die Menge der Information und die benötigte Fläche formal.

Als *Informationsdichte* wird das Verhältnis der Menge der grafisch dargestellten Informationen zu der dafür zur Verfügung stehenden Fläche bezeichnet. Es wird angestrebt, diese Dichte über die ganze Spannweite der Massstabsänderung konstant zu halten, denn dies trägt stark dazu bei, ein sauberes, ruhiges und auch beim Zoomen konsistentes Kartenbild zu erhalten.

Der *Informationsgehalt* hingegen ist die Menge aller in der Karte enthaltenen Informationen – ohne Berücksichtigung der Fläche. Dieser soll mit zunehmend grösser werdendem Massstab auch grösser werden, und zwar in einem solchen Ausmass, dass die Informationsdichte konstant bleibt.

Bei einer Vergrößerung des Massstabes soll also der Informationsgehalt erhöht werden. Da der adaptive Datensatz so ausgelegt sein muss, dass verschiedene Objektgruppen wie z. B. Flüsse in unterschiedlichen Dichtestufen («locker», «mittel», «dicht») vorkommen, können durch den Zoomprozess der Darstellung kontinuierlich neue Dichtestufen hinzugefügt werden, um den Informationsgehalt zu erhöhen. Sie werden dann hinzugefügt, wenn

Def. Skalierungsgrenzwerte

Def. Informationsdichte

Def. Informationsgehalt

Def. Informationsgehalt-Grenzwert

Def. Schwere und leichte Objekte

*Festlegung des Darstellungsgewichtes
nur empirisch möglich*

der Massstab einen festgelegten Grenzwert, nämlich den *Informationsgehalt-Grenzwert*, überschreitet, und zwar so, dass die Informationsdichte konstant bleibt. Dieser Informationsgehalt-Grenzwert hängt ab von verschiedenen Faktoren, insbesondere vom Darstellungsgewicht der Objektgruppen und der Objekte.

7.5.1 Das Darstellungsgewicht von Objekten und Objektgruppen

Je nach Beschaffenheit der Signatur eines Objektes ist deren Wirkung auf die Kartendarstellung bei einer Veränderung des Massstabs sehr unterschiedlich. So haben beispielsweise feine, einfache, grobgliedrige Objektsignaturen wie Eisenbahnlinien relativ wenig Wirkung auf die Darstellung, wenn sie im Massstab verändert werden. Strassen, die als Doppellinien dargestellt werden und mit vielen Kurven häufig sehr feingliedrig sind, haben hingegen eine starke Wirkung, da sie verhältnismässig viel Raum beanspruchen und zur Umgebung ziemlich kontrastreich sind. Erstere Objekte haben ein geringes Darstellungsgewicht und sind daher *leicht*, letztere werden als *schwer* bezeichnet. Ebenso sind einzelne Punktsignaturen eher leicht.

Ganze Objektgruppen sind dann leicht, wenn die Anzahl der Objekte gering ist. Etwa eine Objektgruppe, die aus einer kleinen Menge von Punktsignaturen besteht, die weit voneinander entfernt sind, hat relativ wenig Gewicht. Hingegen ist eine grosse Anzahl von Objekten in einer Objektgruppe schwer.

7.5.2 Informationsgehalt-Grenzwerte

Schwere Objekte und Objektgruppen haben generell einen grösseren Informationsgehalt-Grenzwert als leichte, sie sollen also später eingebledet werden. Mathematisch formulierbar oder numerisch erfassbar ist dieser Zusammenhang aber leider nicht, da schon die Festlegung, ob ein Objekt oder eine Objektgruppe schwer oder leicht ist, fast nur aufgrund des persönlichen Empfindens vorgenommen werden kann:

- ◆ Zwar erscheint es auf den ersten Blick leicht, die Anzahl Objekte einer Objektgruppe numerisch zu erfassen. Da das hier beschriebene Konzept aber darauf ausgelegt ist, Karten aus gängigen Grafikprogrammen zu verwenden, und dort einzelne Objekte fast immer ihrerseits in (Unter-)Objekte aufgeteilt sind, wird eine Zählung verunmöglicht. So besteht ein Fluss (also ein Objekt) immer aus sehr vielen einzelnen Segmenten, die oft sogar je nach Strichstärke in verschiedene Ebenen aufgeteilt sind.
- ◆ Einzelne Objekte unterscheiden sich wiederum zu stark, als dass eine mathematische Erfassung des Gewichtes möglich wäre. Welcher Darstellungscharakteristik wird wieviel Wert zugemessen? Wie lassen sich Kontraste in der Darstellung numerisch erfassen?
- ◆ Informationsgehalt-Grenzwerte, die bei einer Karte zu guten Ergebnissen führen, müssen für andere Karten nicht zwangsläufig auch stimmen, da die Grenzwerte zu sehr vom Kontext der Darstellung abhängen.

Es bleibt also nur, die Informationsgehalt-Grenzwerte mit Hilfe von Tests und für jede adaptive Karte neu festzulegen.

7.6 Anpassung der Signaturmasse: Liniensignaturen

Wenn eine Karte ganz ohne Anpassungen eingezoomt wird, dann werden sämtliche Objekte proportional zum Massstab bzw. Zoomfaktor vergrössert. Eine Linie, die zuerst 1 px stark war, hat bei Vergrösserung um den Faktor 2 folglich die Stärke 2 px (siehe Abb. 15).

Rein geometrisch betrachtet würde dieses Vorgehen stimmen. Da Punkt- und Liniensignaturen aber nicht die Grösse der repräsentierten Objekte wiedergeben, sondern nur dessen Position und höchstens ihre relative Grösse im Vergleich zu anderen Objekten, muss für interaktive Massstabänderung eine andere Lösung gefunden werden (siehe Abb. 16).

Betrachtet man etwa eine Strasse in einer Karte, so hat sie schon im ursprünglichen Massstab eine Strichstärke, welche die geometrisch korrekte Grösse weit übersteigt. Z. B. die Signatur für Hauptstrassen in der Landeskarte 1 : 500 000 stellt masstäblich umgerechnet eine 250 m breite Strasse dar. Wollte man sie masstäblich richtig darstellen, so müsste man eine Linie mit der Strichstärke 0,012 mm zeichnen.

Man ist nun versucht, die Signaturmasse konstant zu halten. Diese Möglichkeit ist einfach und verhältnismässig schnell zu implementieren. Es scheint auch, als müsste sie bestens geeignet sein: Wenn man die Signaturmasse für eine Karte eines bestimmten Massstabs so wählt, dass sich ein gutes Kartenbild ergibt, sollte man davon ausgehen können, dass die gleichen Signaturmasse auch bei anderen Massstäben stimmen. Einige Blätter der Schweizer Landeskarten sind entsprechend aufgebaut. Etwa die Massstäbe 1 : 25 000, 1 : 50 000 und 1 : 100 000 machen grösstenteils von den gleichen Signaturen Gebrauch.

Auf eine adaptive Karte bezogen ist das Konstanthalten von Signaturmassen – das in der Landeskarte bestens funktioniert – leider nicht geeignet. Ein wichtiges Merkmal der adaptiven Karte ist es, dass die Veränderungen der Karte während dem Zoomvorgang möglichst unauffällig sein sollen. Insbesondere wird versucht, durch Anfügen bzw. Weglassen von Objekten und Objektgruppen die Informationsdichte konstant zu halten. Unter solchen Voraussetzungen wirken sich konstante Signaturmasse negativ auf das Kartenbild aus. Mit wachsendem Massstab verlieren Signaturen im Vergleich zur Umgebung aufgrund des allgemeinen Platzgewinns an Dominanz (siehe Abb. 17). Bei sich verkleinerndem Massstab geschieht das Umgekehrte: Die Betonung der Karte verlagert sich schnell in Richtung der Signaturen. Besonders augenfällig ist dieses Verhalten bei linearen Objekten, vor allem bei Signaturen mit Doppellinien wie Strassen oder feingliedrigen, stark kurvigen Objekten wie Flüssen.

Aus den bisherigen Erkenntnissen lässt sich also der Schluss ziehen, dass die Signaturen in irgend einer Weise vergrössert werden müssen. Doch wie kann diese Vergrösserung gerechnet werden? In der Literatur liessen sich kaum Hinweise finden.

Töpfer (Töpfer, 1962 und andere Artikel) beschreibt zwar mit dem Wurzelgesetz einen ähnlichen Fall, er bezieht seine Ausführungen aber auf Papierkarten und schliesst die komplette Generalisierung mit ein. Das Wurzelgesetz sagt ganz allgemein aus, dass die Generalisierung einer Karte von der Wurzel des Kartenmassstabs abhängt. Er hat das Gesetz auch auf Signaturmasse umformuliert. Liegt für einen Massstab $1 : m_0$ die richtige Signaturgrösse s_0 vor, so kann die richtige Signaturgrösse s_1 für den Massstab $1 : m_1$ berechnet werden (Töpfer, 1962:38):

$$s_1 = s_0 \sqrt{\frac{m_0}{m_1}} \quad (4)$$

Leider hat sich das Wurzelgesetz in der Kartografie oft als ungeeignet erwiesen (Brandenberger, 2000). Bezogen auf die massstabsabhängige Veränderung von Signaturmassen lässt sich zwar zeigen, dass eine Anpassung

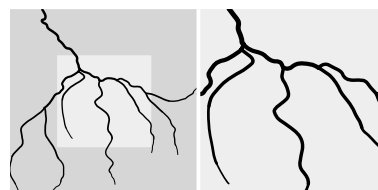


Abb. 15 Proportionale Vergrösserung eines Kartenausschnittes auf 200%

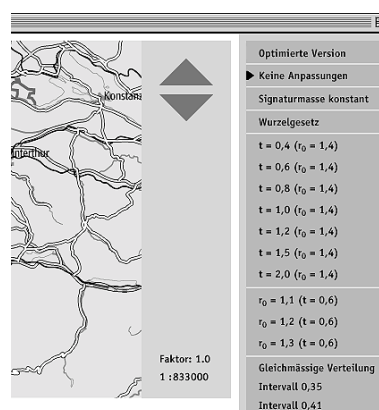


Abb. 16 Proportionale Vergrösserung der Signaturmasse

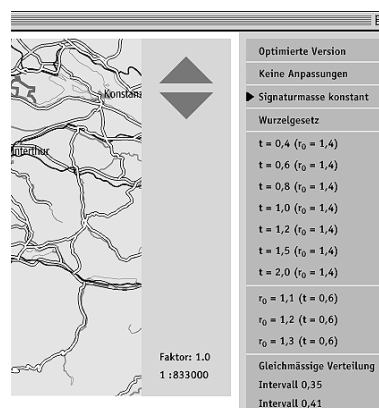


Abb. 17 Konstanthalten der Signaturmasse

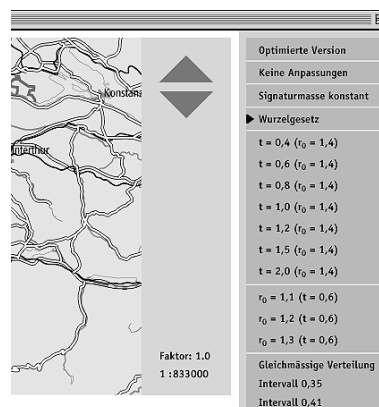


Abb. 18 Anpassung der Signaturmasse mit dem Wurzelgesetz

gemäss dem Wurzelgesetz zu deutlich besseren Resultaten führt als das Konstanthalten von Signaturmassen (siehe Abb. 18). Das Gesetz erlaubt aber weiterhin beliebig grosse Signaturmasse, wenn auch mit deutlich kleinerer Wachstumsgeschwindigkeit.

Es muss also eine andere Funktion als (4) zur Anpassung der Signaturen einer adaptiven Karte gefunden werden. Eine solche Funktion hat die durch die verschiedenen Massstäbe bewirkte Veränderung der Charakteristiken einer adaptiven Karte zu berücksichtigen. Zwar soll die Informationsdichte der Karte über den gesamten Massstabsbereich unverändert bleiben, da aber die absolute Menge der gezeigten Informationen und somit das Kartenbild variiert, ist es nötig, auch die kartografische Darstellung anzupassen.

Insbesondere muss miteinbezogen werden, dass sich die Art der Informationen, die beim Zoomprozess hinzukommen, mit grösser werdendem Massstab verändert. Dies hat bedeutende Auswirkungen auf die Signaturen. In den eher kleinmassstäblichen Bereichen sind Informationen vorwiegend als Punktsignaturen visualisiert, die Liniennetze sind eher locker und die flächenhaften Darstellungen ziemlich grob. Mittlere Bereiche der Massstabsreihe weisen dagegen eine zunehmende Dichte an linearen Objekten und grossmassstäbliche Darstellungen vermehrt auch detaillierte flächenhafte Darstellungen auf.

Solange die Liniennetze nicht sehr dicht sind und den Punktsignaturen eine verhältnismässig grosse Bedeutung zu kommt, dürfen die Signaturmasse für Linien und Punkte relativ stark wachsen. Sobald aber mehr platzintensive Objekte hinzukommen, ist ein schwächer werdendes Wachstum notwendig, um eine übertriebene Betonung der betroffenen Objekte zu verhindern.

An die gesuchte Funktion werden weiter die folgenden Anforderungen gestellt:

- ◆ Das Resultat der Funktion ist der Skalierungsfaktor für alle Signaturmasse.
- ◆ Berücksichtigung des aktuellen und des Ausgangsmassstabs: Ausgehend von einem Ausgangsmassstab sollen für jeden frei wählbaren Massstab die Signaturmasse berechnet werden können.
- ◆ Annäherung an ein bestimmtes Maximum: Die Signaturmasse sollen einen definierten Wert nicht überschreiten dürfen.
- ◆ Skalierungsfaktor 1 für die Karte (keine Vergrösserung) ergibt Skalierungsfaktor 1 für die Signaturen.

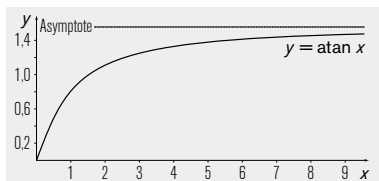


Abb. 19 Der Graph der Funktion (5)

Die Bedingung einer gleichmässigen Annäherung erfüllen asymptotische Funktionen. Gut geeignet zu sein scheint die Funktion (5) (siehe Abb. 19):

$$y = \operatorname{atan} x \quad (5)$$

Deren Asymptote liegt bei $\pi/2$, der y -Wert kann also nie grösser als $\pi/2$ werden. Diese Funktion ist für die Anpassung der Signaturmasse grundsätzlich ziemlich gut geeignet, wenn man davon ausgeht, dass x nicht direkt dem Skalierungsfaktor, sondern dessen Abweichung von 1 entspricht. Falls z. B. ein Signaturmass um den Faktor 1,4 vergrössert werden soll, so hat x den Wert 0,4.

Um den exakten Skalierungsfaktor zu erhalten, genügt es also, die Funktion (5) zur neuen Funktion

$$y = \operatorname{atan} x + 1 \quad (6)$$

zu erweitern, womit der Graph entlang der y -Achse nach oben geschoben wird (siehe Abb. 20).

Diese Funktion bietet noch nicht die Möglichkeit, über die genaue Lage der Asymptote und somit des maximalen Vergrößerungsfaktors zu bestimmen. Erreicht wird dies aber mit der Einführung eines Parameters r , der die Asymptote entlang der y -Achse verschiebt. Die neue Funktion lautet

$$y = r \operatorname{atan} x + 1 \quad (7)$$

r steuert nun die Position der Asymptote und kann in Abhängigkeit von $\pi/2$ festgelegt werden. Wenn sich die Asymptote von (5) bei $\pi/2$ befindet, dann kann sie gemäss

$$r = \frac{r_0 - 1}{\frac{\pi}{2}} = \frac{2(r_0 - 1)}{\pi} \quad (8)$$

entlang der y -Achse verschoben werden, wobei r_0 der gewünschte maximale Vergrößerungsfaktor ist. Mit der Verkleinerung um 1 wird der Vergrößerungsfaktor r_0 gemäss obiger Festlegung, dass x der Abweichung des Skalierungsfaktors von 1 entspricht, normalisiert.

Die vorläufige Funktion zur Ermittlung des Vergrößerungsfaktors lautet also (siehe Abb. 21):

$$y = \frac{2(r_0 - 1)}{\pi} \operatorname{atan} x + 1 \quad (9)$$

Der maximale Vergrößerungsfaktor r_0 kann mathematisch nicht ermittelt, sondern nur empirisch festgelegt werden, da er sich je nach Kartengrafik in Abhängigkeit von Informationsdichte, Wahl der Signaturen und Qualität der Darstellung stark verändern kann.

Unter Umständen ist es hilfreich, wenn man selber bestimmen kann, wie schnell die Signaturmasse wachsen bzw. wie schnell die Funktion sich an die Asymptote annähert. Z. B. sollten vielleicht bereits ab einem Kartenskalerungswert von 2 die Signaturmasse nicht mehr wesentlich grösser werden. Zusätzlich muss deshalb ein Parameter t eingeführt werden, der bestimmt, wie schnell die Kurve wie nahe an der Asymptote ist.

Die entsprechend modifizierte Funktion lautet (siehe Abb. 22):

$$y = \frac{2(r_0 - 1)}{\pi} \operatorname{atan} t x + 1 \quad (10)$$

Auch t ist ein empirisch zu ermittelnder Wert, der nur aufgrund von Erfahrungswerten festgelegt werden kann.

Ein Beispiel: Sollen bei einer Vergrößerung um den Faktor 1,6 ($x = 0,6$) die Signaturen angepasst werden, und hat man die maximale Vergrößerung r_0 auf 1,4 und das Signaturenwachstum t auf 2 festgesetzt, dann müssen die Signaturmasse um den Faktor 1,22 vergrössert werden.

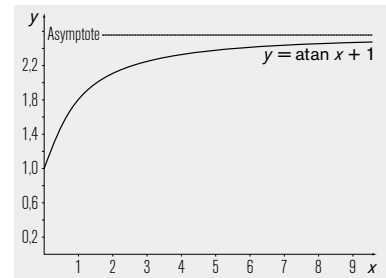


Abb. 20 Der Graph der Funktion (6)

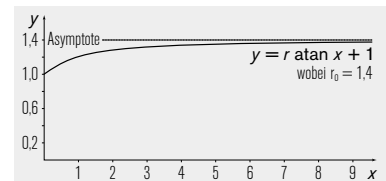


Abb. 21 Der Graph der Funktion (9)

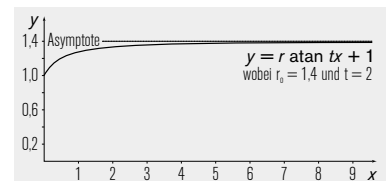


Abb. 22 Der Graph der Funktion (10)

Für jedes Signaturmass einer Karte gilt also

$$s_1 = s_0 \left(\frac{2(r_0 - 1)}{\pi} \operatorname{atan} \left(t \left(\frac{m_0}{m_1} - 1 \right) \right) + 1 \right) \quad (11)$$

wobei

s_0 das ursprüngliche Signaturmass,

s_1 das neue Signaturmass,

r_0 der gewünschte maximale Skalierungsfaktor,

m_0 der ursprüngliche Massstab,

m_1 der neue Massstab

und t die gewünschte Wachstumsgeschwindigkeit ist.

Diese Funktion hat vor allem bei Liniensignaturen grosse Bedeutung. Besonders gut sichtbar wird sie bei schweren Liniensignaturen wie Doppellinien oder feingliedrigen Linien. Die Funktion kann aber unter gewissen Umständen und mit Berücksichtigung anderer Faktoren auch für Punktsignaturen und die Schriftgrösse von Beschriftungen verwendet werden, wie dies in den folgenden Kapiteln beschrieben wird.

7.7 Anpassung der Punktsignaturen

Aus der Kartenproduktion mit Grafikprogrammen ist bekannt, dass der Umgang mit Punktsignaturen aufwändig und mühsam sein kann. Der oft gegangene Weg, eine Signatur zu zeichnen, sie mit *copy – paste* in der richtigen Anzahl zu kopieren und an der gewünschten Stelle zu platzieren, beinhaltet viele Nachteile. Schwierig wird es besonders dann, wenn z. B. alle Signaturen nach der Platzierung etwas kleiner gemacht werden sollen. Dieses Problem muss – zumindest in den gängigen Grafikprogrammen – gelöst werden, indem man jede Signatur einzeln auswählt und in der Grösse anpasst.

Punktsignaturen als Schrift

Viele Kartenautoren behelfen sich des Tricks, eine Punktsignatur wie eine Schrift zu behandeln. Dazu wird in einem Schriftbearbeitungsprogramm (z. B. *Macromedia Fontographer*) die Signatur gezeichnet und als Schrift abgespeichert. Wenn die Signatur jetzt platziert werden soll, wird einfach der Buchstabe in der entsprechenden Schrift abgesetzt, genau wie vorher kopiert und an die richtigen Orte platziert. Ein solches Vorgehen hat zwei grosse Vorteile:

- ◆ Die Grösse der Signaturen kann in sehr kurzer Zeit durch Änderung der Schriftgrösse angepasst werden.
- ◆ Die Form einer Signatur kann verändert werden, indem die Signaturenschrift im Schriftbearbeitungsprogramm geöffnet, die entsprechende Signatur verändert und die Schrift neu abgespeichert wird. Beim nächsten Öffnen der Karte im Grafikprogramm wird die Signatur automatisch angepasst.

Nachteilig wirkt sich hingegen aus, dass Signaturen – genau wie Schriften – nur einfarbig sein können. Man kann diesen Nachteil aber umgehen, indem man mehrere einfarbige Signaturen exakt übereinander legt, wobei pro Farbe eine Signatur benötigt wird.

Die Vorteile dieses Konzeptes lassen sich auch für adaptive Karten nutzen. Insbesondere die Grössenveränderung durch Anpassung der Schriftgrösse passt gut für adaptive Karten. Anstatt jede Punktsignatur einzeln zu skalieren, genügt es, nur einmal die Schriftgrösse mit Hilfe der Funktion (11) für die Anpassung der Signaturmasse neu zu berechnen und auf alle Punktsignaturen anzuwenden. Ein solcher Ansatz wirkt sehr elegant, birgt aber dennoch ein Problem.

In SVG wird Text nämlich ähnlich wie Grafik behandelt, zumindest nach der hier verwendeten Spezifikation 1.0 und mit dem *Adobe SVG-Viewer 1.0* (vgl. zu Text Kapitel 6.2.7 *Text-Elemente* auf Seite 39 und zum *Adobe SVG-Viewer* Kapitel 8.2 *Der Adobe SVG-Viewer mit Netscape Communicator als Ausgabepattform* auf Seite 60). Text ist zwar in einem SVG-Bild wie in HTML jederzeit frei selektierbar und sogar – je nach Applikation – interaktiv veränderbar, er kann aber stets nur einzellig sein.

Der Ursprung des Textes ist immer der am weitesten links gelegene Punkt der Grundlinie eines Textes (siehe Abb. 23), ganz unabhängig davon, ob der Text links- oder rechtsbündig oder gar zentriert abgesetzt wurde. Wenn nun eine Punktsignatur, die ja auch Text ist, verkleinert wird, geschieht dies nicht vom Mittelpunkt der Signatur aus – was der Fall wäre, wenn Text zentriert abgesetzt werden könnte. Der Mittelpunkt der Signatur und mit ihm die ganze Signatur verschiebt sich also etwas nach links und gegebenenfalls nach unten. Um die Signatur am ursprünglichen Ort zu halten, ist demzufolge eine Verschiebung der Signatur nach rechts bzw. oben notwendig.

7.7.1 Positionskorrektur entlang der x-Achse

Der Betrag der Positionskorrektur nach rechts entspricht der Differenz der Signaturgrößenradien, also der halben Differenz der Signaturgrößen:

$$v_x = \frac{s_0 - s_1}{2} \quad (12)$$

wobei v_x die notwendige Verschiebung entlang der x -Achse beschreibt, s_0 die ursprüngliche Signatur- bzw. Schriftgröße und s_1 die neue Signaturgröße (siehe Abb. 24).

7.7.2 Positionskorrektur entlang der y-Achse

Würde die Punktsignatur während der Schriftgestaltung willkürlich im zur Verfügung stehenden Raum positioniert, so wäre jetzt auch noch eine Verschiebung entlang der y -Achse notwendig, und zwar um die Differenz zwischen den Strecken von der Grundlinie zum Mittelpunkt.

Der Rechen- und der Entwicklungsaufwand können aber verkleinert und die Handhabung vereinfacht werden, wenn der Mittelpunkt einer Signatur bereits während der Schriftgestaltung auf die Grundlinie gelegt wird. So wird erreicht, dass die Differenz immer null ist und eine Verschiebung nur entlang der x -Achse erfolgen kann (siehe Abb. 24).

7.8 Beschriftung von Punktsignaturen

Ein ganz eigenes Problem stellt die Kartenbeschriftung dar. Kartenbeschriftung ist ein besonderer Bestandteil einer Karte, da sie unter allen Gestaltungsmitteln die geringste geometrische Aussagemöglichkeit besitzt. Sie ist dafür aber das wichtigste erläuternde Element der Karte. Sie ist deutlich komplexer in der Handhabung als die oben beschriebenen Konzepte und teilt sich in drei Bereiche auf.

Einerseits soll in diesem Kapitel die Anpassung der Schriftgröße an den Massstab untersucht werden. Andererseits ist die Schriftplatzierung von Bedeutung, die ähnlich erfolgen kann wie die Punktsignaturplatzierung, aber einiges komplexer ist. Ein weiteres Thema ist die Schriftfreistellung. Diese Aufgabe ist allerdings derart komplex, dass eine Untersuchung den Rahmen



Abb. 23 Der Ursprung eines Textes in SVG

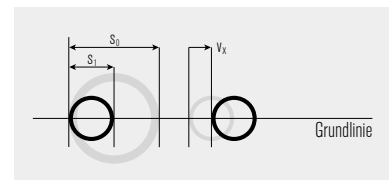


Abb. 24 Positionskorrektur von Punktsignaturen entlang der x -Achse

dieser Arbeit sprengen würde. Deshalb wird hier auf Schriftfreistellung verzichtet – wenn auch leider auf Kosten der Qualität.

Aus dem gleichen Grund wird nur die Beschriftung von Punktssignaturen untersucht. Die Beschriftung von Flächen und Linien wird in dieser Arbeit nicht berücksichtigt.

7.8.1 Masstabsabhängige Optimierung der Schriftgrösse

Genau wie bei Punkt- und Liniensignaturen darf auch die Kartenbeschriftung nicht proportional zum Massstab vergrössert werden. Auch sie würde überproportionale Bedeutung erlangen. Da die Kartenbeschriftung wie die Signaturen im Kontext der Umgebung ihre Bedeutung nicht ändern soll, ist eine Schriftgrössenveränderung gemäss der Funktion (11) für die Anpassung der Signaturmasse die beste Wahl, wobei die beiden Parameter r_0 und t grundsätzlich die gleichen Werte annehmen sollen wie bei Linien- und Punktssignaturen.

Neben Informationsdichte und anderen grafischen und kartografischen Faktoren, welche die Schriftfamilie und die Schriftgrösse bestimmen, kommen erschwerend typografische Eigenheiten hinzu, die es auf Bildschirmen zu beachten gilt. In Kapitel 4.4.2 «Wahl der Schrift für Bildschirmkarten» auf Seite 24 wurde auf die Wahl von geeigneten Schriften für die Bildschirmausgabe kurz eingegangen. Eine für die Bildschirmausgabe gut geeignete Schrift darf nur in gewissen Grenzen skaliert werden, um die Qualität der Darstellung und somit die Lesbarkeit zu wahren.

Bezogen auf die Funktion (11) bedeutet dies, dass die Position der Asymptote maximal einen Wert annehmen darf, welcher der oberen Skalierungsgrenze der benützten Schrift entspricht. Bei den auf Bildschirmen meistens verwendeten Schriften ist in der Regel die typografisch sinnvolle Skalierungsgrenze höher als der maximale Skalierungsfaktor von Signaturmassen und muss deshalb nicht berücksichtigt werden, da diese Schriften eher einfach gehalten sind. Dennoch gilt es, diese Einschränkung vor allem bei bezüglich Grössenveränderungen besonders empfindlichen, feineren Schriften zu beachten. Übrigens lässt sich auch der Wert der maximalen Skalierbarkeit einer Schrift nicht ohne weiteres numerisch festlegen, sondern muss durch Versuche ermittelt werden.

7.8.2 Platzierung der Kartenbeschriftung

Wie die als Schrift gesetzten Punktssignaturen werden auch bei der Kartenbeschriftung durch die Masstabsveränderung und die dadurch entstehende Änderungen der Schriftgrösse gewisse Anpassungen in der Positionierung notwendig, da Text in SVG grundsätzlich nur linksbündig und einzeilig gesetzt werden kann (vgl. Kapitel 7.7 «Anpassung der Punktssignaturen» auf Seite 54). Durch die Schriftgrössenunterschiede ändern sich sowohl die Textbreite als auch die Buchstabenhöhe.

Der hier beschriebene Lösungsweg geht nur auf die Platzierung der Labels von Punktssignaturen, z. B. Ortschaften ein. Alle anderen Beschriftungen, etwa für Flächenobjekte und lineare Objekte werden nicht berücksichtigt.

Zudem geht man davon aus, dass die Kartenbeschriftung im Ausgangsmassstab bereits platziert ist und nur noch an neue Massstäbe angepasst werden muss. Auf den ersten Blick scheint eine solche Lösung zwar «quick and dirty» zu sein, doch zeigt die genauere Betrachtung einer vollständig automatisierten Lösung, dass eine solche allein ein Thema für ein neues Forschungsprojekt abgäbe.

Ein automatisierter Ansatz würde beinhalten, dass jeder Punktssignatur genau ein Label zugewiesen wird. Je nach Art der umgebenden Signaturen

*Nur Platzierung der Labels von
Punktssignaturen werden untersucht*

erfolgte eine entsprechende Platzierung, und zwar so, dass sich das Label nicht mit anderen Texten überschneidet. Ausserdem müsste die Position auf gute Lesbarkeit optimiert gewählt werden und daher allen in Kapitel 4.2.8 «Anordnung der Kartenbeschriftung» auf Seite 20 beschriebenen Regeln folgen. Allein schon der Umstand, dass viele dieser Regeln nicht oder nur eingeschränkt in Formeln fassen lassen, erschwert die Sache. Es gibt aber noch mehr Probleme: Eine Verknüpfung der Punktsignaturen und der zugehörigen Label wäre zwar sowohl mit *Adobe Illustrator* seit Version 9 als auch mit SVG möglich, leider geht die Verknüpfung aber beim Export aus *Illustrator* verloren. Man könnte die Zusammengehörigkeit von Signatur und Label natürlich von Hand ergänzen, doch der zeitliche Aufwand wäre nicht praxistgerecht. Insgesamt gesehen stehen Aufwand und Ertrag also bei diesem Ansatz in keinem nutzbaren Verhältnis.

7.8.3 Adaptive Platzierung der Kartenbeschriftung

Ein Label bleibt relativ zur Punktsignaturen dann auf dem richtigen Platz, wenn bei der Massstabsveränderung weder der horizontale, noch der vertikale Abstand zwischen dem Label und dem Rand der Punktsignaturen verändert werden. Da aber die Grösse sowohl der Schrift als auch der Punktsignaturen modifiziert werden, sind Verschiebungen des Labels notwendig, die beides korrigieren.

Die Grössenänderung der *Punktsignaturen* wirkt in alle Richtungen und hat daher immer Einfluss auf die Labelposition. Ein Label muss also immer um die Differenz zwischen altem und neuem Radius der Punktsignaturen in deren Richtung geschoben werden (siehe Abb. 25).

Die Grössenänderung des *Labels* hat hingegen nur dann Einfluss, wenn sich das Label links und/oder unterhalb der Punktsignaturen befindet (siehe Abb. 26). Ein Label links von der Punktsignaturen wird mit veränderter Schriftgrösse und somit veränderter Labelbreite näher bzw. weiter entfernt von der Punktsignaturen sein, da sein Ursprung nicht verschoben wird (vgl. Kapitel 7.7 «Anpassung der Punktsignaturen» auf Seite 54). In diesem Fall ist eine Verschiebung entlang der x-Achse um den Betrag der *Labelbreitendifferenz* notwendig. Auch die Labelhöhenveränderung geht vom auf der Grundlinie gelegenen Ursprung aus. Kleinere, also weniger hohe Schrift ist etwas weiter von der Punktsignaturen entfernt als ihr Original. Sie muss entlang der y-Achse um den Betrag der *Versalhöhendifferenz* verschoben werden, wobei die Versalhöhe derjenige Teil eines Buchstabens ist, der sich oberhalb der Grundlinie befindet.

Es gibt bezogen auf die Position einer Punktsignaturen nur vier mögliche Standorte für ein Label (siehe Abb. 26). Teilt man den eine Punktsignaturen umgebenden Raum in vier Quadranten auf, so muss sich der Schwerpunkt eines Labels rechts oben (Quadrant A), rechts unten (Quadrant B), links unten (Quadrant C) oder links oben (Quadrant D) befinden. Der Label-schwerpunkt ist hierbei das Zentrum der Grundlinie. Seine genaue Position innerhalb des Quadranten ist nicht definiert und hängt von der umgebenden Grafik ab. Ein Label kann also ohne weiteres auch fast in der Mitte oberhalb bzw. unterhalb einer Punktsignaturen positioniert sein – falls dies kartografisch Sinn macht.

Bei einer Massstabsveränderung wird ein Label nur innerhalb seines Quadranten angepasst und kann nie aus dem Quadranten hinausgeschoben werden. Alle Labels, die dem gleichen Quadranten zugeordnet sind, müssen in die gleiche Richtung und um den gleichen Betrag verschoben werden. Mit diesem System ist von vornherein klar, wie und um welchen Betrag die Position eines Labels korrigiert werden muss.

In SVG befindet sich der Nullpunkt aller Transformationen links oben, Verschiebungen nach rechts und unten sind also positiv, nach links und

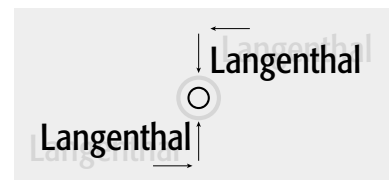


Abb. 25 Wirkung der Grössenänderung von Punktsignaturen

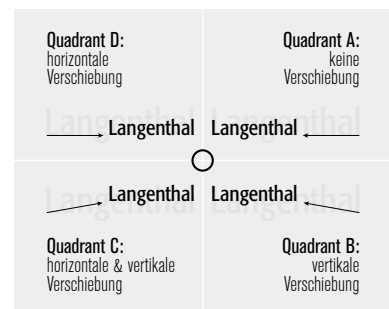


Abb. 26 Wirkung der Grössenänderung von Labels und die Raumaufteilung in Quadranten

oben negativ. Tab. 6 enthält sämtliche mögliche Verschiebungsbeträge in x- und y-Richtung.

Labelposition	x-Korrektur	y-Korrektur
Quadrant A	– Signaturgrössendifferenz	+ Signaturgrössendifferenz
Quadrant B	– Signaturgrössendifferenz	– (Signaturgrössendifferenz + Versalhöhendifferenz)
Quadrant C	+ Signaturgrössendifferenz + Labelbreitendifferenz	– (Signaturgrössendifferenz + Versalhöhendifferenz)
Quadrant D	+ Signaturgrössendifferenz + Labelbreitendifferenz	+ Signaturgrössendifferenz

Tab. 6 Die Beträge der Positionskorrektur für Kartenbeschriftung

7.8.4 Platzierung der Kartenbeschriftung im Grafikprogramm

Damit die adaptive Platzierung der Kartenbeschriftung später einwandfrei vor sich gehen kann, müssen im adaptiven Basisdatensatz die Positionen sämtlicher Labels nach einem konsistenten System festgelegt werden.

Alle Labels sollen, unabhängig davon, ob sie ein- oder ausgeblendet sind, bereits im Ausgangsmassstab so platziert werden, dass ihre Position später zur Berechnung der angepassten Position verwendet werden kann. Anders ausgedrückt, müssen alle Labels im Ausgangsmassstab bezüglich der zugehörigen Signaturen richtig platziert werden, und zwar mit denjenigen Schriftgrössen, die später zur Berechnung der angepassten Schriftgrössen als Ausgangswert verwendet werden.

8 Entwicklungsumgebung und Ausgabeplattform

Die bisherigen Kapitel dieser Arbeit haben die Grundlage für die Entwicklung einer adaptiven Karte geschaffen. Für eine solche Karte werden ein entsprechend präparierter adaptiver Datensatz und ein Steuerprogramm, das diesen Datensatz adaptiv modifiziert, benötigt. Die nächsten Kapitel setzen sich mit der Entwicklung dieser beiden Teile auseinander.

Das vorliegende Kapitel beschreibt zuerst die Entwicklungsumgebung; es wird ausgeführt, welche Applikationen wofür benutzt werden. Danach werden die Funktionsweise und die Schwächen des *Adobe SVG-Viewers* als Ausgabeplattform beschrieben.

8.1 Die Entwicklungsumgebung und deren Möglichkeiten

Die Entwicklung der adaptiven Karte und der dazu benötigten Programme geschieht auf einem *Apple PowerMac G4/400* unter *Mac OS 9.04*. Alternativ hätte die Entwicklung auch unter *Microsoft Windows* geschehen können, da alle verwendeten Applikationen auch für *Windows* existieren. *Unix*-basierte Entwicklungsplattformen wie *Linux* sind mangels genügender SVG-Unterstützung noch nicht geeignet.

8.1.1 Produktion eines SVG-Dokumentes mit Illustrator

Für die grafischen Arbeiten wird das neu erschienene Vektorgrafikprogramm *Adobe Illustrator 9* verwendet, das als einziges Grafikprogramm zur Zeit *native support* für SVG bietet. Allerdings basiert, wie früher schon erwähnt, die SVG-Implementation nicht auf der finalen Version der SVG-Spezifikation.

Mit den in Kapitel 6 ‹Technische Grundlagen von SVG› auf Seite 31 beschriebenen Methoden ist es natürlich möglich, Grafiken nur in einem Texteditor zu produzieren. Dies ist aber eine mühsame Arbeit und in der Praxis kaum umsetzbar. Der unter anderem von der Firma *Adobe* vorgezeichnete Weg sieht deutlich komfortabler aus.

Adobe richtet ihre Produktpalette nach und nach auf SVG aus. So unterstützt das weitverbreitete Grafikprogramm *Illustrator* schon seit der Version 8.01 (mit Plug-In) den Export im SVG-Format, die neue Version 9 kommt ohne Plug-In aus und verfügt gar über eine *SVG-interactivity*-Palette. Trotzdem ist noch immer viel Handarbeit angesagt, und Kenntnisse in *JavaScript* sind Voraussetzung.

Im Prinzip kann jede beliebige Grafik aus *Illustrator* ins SVG-Format exportiert werden. Einige Einschränkungen müssen dennoch gemacht werden. Sie haben aber ihren Ursprung darin, dass der *SVG-Viewer*, das Plug-In für Browser, bisher trotz der Versionsnummer 1.0 eigentlich noch nicht komplett fertig ist (vgl. Kapitel 8.2 ‹Der Adobe SVG-Viewer mit Netscape Communicator als Ausgabeplattform› auf Seite 60). Einige Elemente, wie z. B. Muster, oder verschiedene Funktionen, etwa das Laden der Inhalte aus anderen SVG-Anwendungen, werden leider noch nicht unterstützt.

Eine Grafik wird ins SVG-Format exportiert, indem unter *File > Export* der entsprechende Dialog aufgerufen und das richtige Format ausgewählt wird. Der Benutzer hat hier mehrere Entscheidungen zu treffen:

- ◆ *Font subsetting*: Je nach Grafik ist es manchmal sinnvoll, nur die benutzten Buchstaben, alle Buchstaben ausser der Sonderzeichen oder den ganzen Zeichensatz der verwendeten Schrift zu exportieren.

- ◆ *Embedded font location*: Schriften können entweder in einem externen Dokument oder innerhalb der SVG-Anwendung abgespeichert werden.
- ◆ *Raster image location*: Wie Schriften können auch Rasterbilder intern oder extern abgelegt sein.
- ◆ *Decimal places*: Die Genauigkeit der Vektorpositionen kann zwischen einer und sieben Stellen nach dem Komma festgelegt werden.
- ◆ *Encoding*: Der Autor kann zwischen ASCII und *Unicode* wählen.
- ◆ *CSS property location*: Es stehen drei Möglichkeiten zur Verfügung, mit Objektstilen umzugehen. Die wichtigste, bezüglich Ladezeit schnellste Methode ist *style attributes (entity references)*. Dabei werden die Objektstile in der DTD in *entities* abgelegt und per *reference* in das Dokument eingefügt (vgl. Kapitel 6.1.1 «Die Konzepte von XML» auf Seite 32). Mit der Option *style attributes* werden die *attributes in line*, d. h. unmittelbar beim Objekt, abgespeichert. Die Option *style elements* produziert ein eigentliches CSS-Dokument.

8.1.2 Adobe GoLive als JavaScript-Editor

Einfache SVG-Grafiken, z. B. Buttons für Web-Seiten, können direkt in *Illustrator* mit Interaktivität versehen werden. Mit Hilfe der *SVG-interactivity*-Palette können *mouse rollovers* und andere Dinge relativ schnell erstellt werden. Solche *JavaScripts* werden direkt in der SVG-Anwendung gespeichert und gehören zu den eigentlichen Objekten.

Soll eine Funktion etwas komplexer werden oder ein *JavaScript* für verschiedene Dokumente (andere SVG-Grafiken oder auch HTML-Dokumente) zugänglich sein, ist es einfacher, einen guten *JavaScript*-Editor zu benutzen.

Als *JavaScript*-Editor geeignet ist das *Web-Authoring-Tool Adobe GoLive 4.01*. SVG wird zwar in der Version 4 nicht unterstützt (in der auf Herbst 2000 angekündigten Version 5 soll aber SVG implementiert sein), da das Testen der Anwendung aber sowieso in einem Webbrowser geschehen soll, muss dieser Mangel nicht berücksichtigt werden. Der *JavaScript*-Editor von *GoLive* ist übersichtlich, schnell und einfach zu bedienen. Der HTML-Code der zugrundeliegenden Homepage kann im gleichen Programm und im gleichen *GoLive*-internen Projekt weitgehend grafisch bearbeitet werden.

8.1.3 BBEdit Lite als SVG-Editor

Für das Editieren des SVG-Codes wird der leistungsfähige, sehr schnelle und kostenlose Texteditor *BBEdit Lite 4.6* verwendet. Betont werden muss insbesondere die hervorragende Suchfunktion des Editors (*find* und *find/replace*), die vor allem das schnelle, dokumentübergreifende Löschen von Stildefinitionen sehr erleichtert (vgl. Kapitel 9.5 «Repetitive Anpassung der Kartengeometrie» auf Seite 72).

8.2 Der Adobe SVG-Viewer mit Netscape Communicator als Ausgabepattform

Verschiedene Softwarehersteller arbeiten zur Zeit an Lösungen, um SVG in ihren Produkten zu implementieren. Auf dem Markt sind auch bereits einige Programme erhältlich, mit denen SVG-Grafik betrachtet werden kann. Die Softwarehersteller stehen allerdings vor dem Problem, dass SVG noch nicht fertig spezifiziert ist, es sind also noch Änderungen möglich. Da

zugleich jeder mit seinem Produkt als erster auf dem Markt sein will, sind die heute erhältlichen *SVG-Viewer* durchwegs unvollständig und oft sogar falsch bzw. nach einer früher richtigen Spezifikation aufgebaut.

Der *Adobe SVG-Viewer 1.0*, der für diese Arbeit verwendet wurde, scheint von allen erhältlichen Produkten die meisten Vorteile zu haben. Insbesondere ist zu erwähnen, dass der als Plug-In für Webbrowser ausgelegte Viewer sowohl unter *Windows* als auch unter *Mac OS* funktioniert. Unter *Windows* können die Browser *Netscape Communicator* und *Internet Explorer* von *Microsoft* verwendet werden, unter *Mac OS* wird nur *Netscape Communicator* unterstützt.

8.2.1 Bedienung des Adobe SVG-Viewer

Der *Adobe SVG-Viewer 1.0* ist ein in *Java* geschriebenes Plug-In, das beim Laden einer *SVG*-Grafik aufgerufen wird. Für die Benutzerin sichtbar ist im Wesentlichen ein kontextsensitives Menü (siehe Abb. 27). Standardmässig gibt es die Möglichkeit, vom Zentrum der Grafik aus zu zoomen (*zoom in*, *zoom out*) und den Ursprungszustand wieder herzustellen (*original view*). Ausserdem kann das *Antialiasing* ein- und ausgeschaltet werden (*higher quality*), eine Animation kann gestoppt (*pause*) oder deren Sound ausgeschaltet (*mute*) werden. Für kartografische Anwendungen sehr interessant und nützlich sind die Suchfunktionen (*find...* und *find again*).

Durch Drücken der *option*-Taste wechselt das Bild des Cursors vom Standardpfeil zu einer Hand (*pan* oder *scroll*). Mit dieser Funktion kann der sichtbare Ausschnitt im *viewport* herumgeschoben werden.

Mit gedrückter *command*-Taste wird aus dem Mauszeiger eine Lupe, mit der sowohl Zoomausschnitt als auch Zoomstärke grafisch bestimmt werden können.

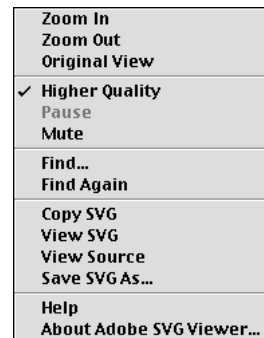


Abb. 27 Das Kontextmenü des *Adobe SVG-Viewer 1.0*

8.2.2 Einschränkungen des Adobe SVG-Viewer

Der *Adobe SVG-Viewer 1.0* unterstützt *SVG* nach der *last call W3C draft specification* vom 3. März 2000, die im August 2000 bereits durch die *candidate recommendation* abgelöst wurde (vgl. Kapitel 6 «Technische Grundlagen von *SVG*» auf Seite 31). Die Bezeichnung «1.0» scheint also wohl etwas verfrüht zu sein, die Auflistung aller bekannten Probleme ist auch entsprechend lang (Adobe, 2000b).

Die Liste mit den augenblicklich nicht unterstützten *SVG*-Funktionen hat sogar noch deutlich grössere Ausmasse (Adobe, 2000d). Da laut *Adobe* vor allem im Bereich des *DOM* (vgl. Kapitel 6.1 «XML im Überblick» auf Seite 31) die Wahrscheinlichkeit beträchtlich ist, dass Teile der Spezifikation geändert werden, wurde fast vollständig auf *DOM*-Implementationen verzichtet. Implementiert sind nur *DOM 1* (dieses dafür vollständig), geringe Teile von *DOM 2* und ein absolut minimaler Set des *SVG*-spezifischen *DOM*.

Damit kann schon recht viel erreicht werden, unglücklicherweise sind aber gerade für adaptive Karten nützliche Funktionen wie etwa *SVGZoom* oder *SVGScroll*, mit denen durch die Standardfunktionen Zoomen oder Scrollen bestimmte Ereignisse ausgelöst werden können, Teil des noch nicht unterstützten *SVG-DOM* (Ferraiolo, 2000). Für adaptive Karten kann also zur Zeit nicht auf die im *Adobe SVG-Viewer* implementierten Standardwerkzeuge zurückgegriffen werden, sondern sie müssen selber entwickelt werden.

Eine weitere nützliche *SVG*-Funktion liessen die *Adobe*-Entwickler bisher beiseite: Es besteht zwar die Möglichkeit, in einer *SVG*-Anwendung Inhalte aus anderen Dokumenten zu platzieren, leider gilt dies aber zur Zeit nur für Bilder in den Formaten *JPEG*, *GIF* und *PNG*, nicht aber für *SVG*- oder andere *XML*-Anwendungen oder Teile davon. Dies hat zur Konsequenz, dass

DOM sehr eingeschränkt implementiert

alle benötigten Objekte in ein und demselben Dokument abgelegt sein müssen. Dynamisches Hinzuladen von Objekten dann, wenn sie benötigt werden, ist noch nicht möglich, was sich in unnötig grossen Dokumenten widerspiegelt.

9 Aufbau des adaptiven Datensatzes

Der adaptive Datensatz muss entsprechend den Anforderungen in Kapitel 7.4 ‹Der adaptive Datensatz› ab Seite 48 so aufgebaut werden, dass er vor allem zwei sehr wichtige Bedingungen erfüllt:

- ♦ Alle Objektgruppen sollen später einzeln durch das Steuerprogramm für die adaptive Karte angesprochen werden können. Deshalb müssen alle mit einer eindeutigen, einmaligen ID versehen werden.
- ♦ Der SVG-Export aus dem Grafikprogramm soll schnell und wiederholbar geschehen können, wenn die Karte aktualisiert werden muss. Der kartografische Datensatz soll also möglichst unabhängig von der Form der Implementation aufgebaut sein.

Aus dem adaptiven Datensatz wird eine SVG-Anwendung entstehen, die so vorbereitet ist, dass sie durch das Steuerprogramm für die adaptive Karte angesprochen und gesteuert werden kann.

9.1 Ausgangsmassstab und maximale Vergrößerung

Der Benutzer einer adaptiven Karte soll bei seinem ersten Kontakt mit der Anwendung eine möglichst gute Übersicht über das vorliegende Gebiet und gegebenenfalls die gezeigten Themen erhalten. Sein nächster Schritt wird es sein, mit den bereitstehenden Werkzeugen das Gebiet und/oder das Thema genauer zu untersuchen.

Es ist daher sinnvoll, dass die Karte oder der zu zeigende Ausschnitt im Grundzustand den kleinsten möglichen Massstab, also den *Ausgangsmassstab*, einnimmt. Hätte die Karte einen anderen als diesen Ausgangsmassstab, dann befände sich der Benutzer bereits von Beginn weg zu sehr im fokussierten Gebiet. Seine erste Reaktion wäre es, sofort auszuzoomen, um sich einen Überblick zu verschaffen.

Da der Ausgangsmassstab derjenige ist, welcher der Benutzer zuerst und tendenziell am längsten betrachtet, ist hier der Kartenqualität besondere Bedeutung zuzumessen. Vor allem die Signaturmasse und die Labelplatzierung müssen stimmen. Ausserdem soll das Kartenbild bezüglich Informationsdichte und Informationsgehalt ausgewogen gestaltet sein. Darum ist auch die Entscheidung von Bedeutung, welche Objektgruppen bereits im Ausgangsmassstab eingeblendet werden und welche erst später hinzukommen.

In Kapitel 7.4.3 ‹Auswahl des richtigen Geometriesatzes› auf Seite 48 wurde erwähnt, dass ein kartografischer Datensatz ohne grossen Qualitätsverlust etwa auf 50% seiner Originalgrösse verkleinert werden kann. Dieser Wert variiert natürlich je nach Komplexität des Datensatzes, und zwar tendenziell eher nach oben. Als provisorischer Ausgangsmassstab scheint er aber recht gut geeignet zu sein, deshalb soll der Herstellungsprozess der kartografischen Datenbasis einschliesslich Abdigitalisieren von der Grundlage und Generalisieren in diesem Massstab erfolgen.

In Kapitel 7.4.3 wurde auch festgehalten, dass ein kartografischer Datensatz auf maximal 200% seiner Originalgrösse skaliert werden darf – wiederum in Abhängigkeit seiner Komplexität. Unter Berücksichtigung der obigen Festlegung kann ein Datensatz also insgesamt auf 400% seines Ausgangsmassstabes vergrössert werden (vgl. Abb. 28).

Def. Ausgangsmassstab



Abb. 28 Schematische Darstellung der Skalierungsgrenzen

9.2 Bearbeitung des adaptiven Datensatzes

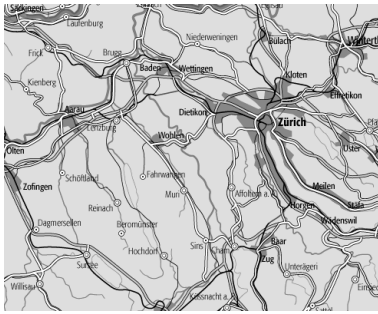


Abb. 29 Der Datensatz s10

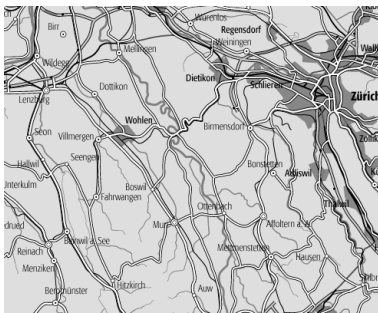


Abb. 30 Der Datensatz s05

Zusammenkopieren beider
Datensätze

Für die adaptive Karte werden zwei möglichst repräsentative kartografische Datensätze verwendet. Der erste Datensatz wird von der Landeskarte 1:1 Mio abdigitalisiert (siehe Abb. 29) und hat im Wesentlichen deren Inhalt. Er wird «s10» genannt, um seine Abstammung von der Landeskarte 1:1 Mio zu kennzeichnen. Der zweite Datensatz soll den ersten im grösseren Massstabereich durch höheren Informationsgehalt ergänzen und wird von der Landeskarte 1:500 000 abdigitalisiert. Er erhält die Bezeichnung «s05» (siehe Abb. 30).

Die Qualität der Kartengeometrie beider Datensätze entspricht, abgesehen von Generalisierungsungenauigkeiten, derjenigen der Landeskarten. Punktsignaturen werden konsequent als Schrift gesetzt (vgl. Kapitel 7.7 «Anpassung der Punktsignaturen» auf Seite 54). Alle anderen Objekte entstehen mit Hilfe konventioneller Vektorgrafik. Objektstile sind provisorisch vergeben und Labels erst ungefähr platziert.

Objekte der gleichen Art sind zu Objektgruppen zusammengefasst. Jede Objektgruppe befindet sich auf mindestens einer eigenen Ebene, kann aber auch auf mehrere aufgeteilt sein, etwa dann, wenn ein Fluss aus mehreren Liniensegmenten mit unterschiedlicher Strichstärke besteht. Die Ebenen sind nach grafischer Priorität geordnet: Die grafisch zuoberst angesiedelte Ebene befindet sich auch «physisch» in der Ebenenpalette an oberster Stelle. Strassen bestehen jeweils aus zwei übereinander liegenden, geometrisch exakt gleichen Ebenen. Die untere, welche aus schwarzen Linien besteht, bildet die Kontur, und die obere, welche weisse, etwas dünnere Linien hat, dient als Füllung.

Bevor die Datensätze weiter bearbeitet werden können, müssen sie gemäss Formel (1) auf Seite 25 für die Bildschirmdarstellung vergrössert werden. Dieser Vergrösserungsfaktor hat einen fixen Wert von 2,4 und gilt für alle Bildschirme und alle Karten, daher muss der Datensatz schon jetzt, bevor er weiter bearbeitet wird, skaliert werden. Neu befinden sich die Datensätze also in den Ausgangsmassstäben 1:833 333 (s10) bzw. 1:416 667 (s05). Beide Datensätze verfügen nun über eine für den jeweiligen Ausgangsmassstab optimierte Kartengeometrie. Die Stile, vor allem die Signaturmasse, blieben bisher unberücksichtigt.

Schon aus bearbeitungstechnischen Gründen, aber auch, weil der *Adobe SVG-Viewer* den Import von Objekten aus anderen SVG-Anwendungen nicht zulässt, werden die beiden Datensätze im nächsten Schritt in ein Dokument zusammenkopiert und so zu Geometriesätzen eines vollständigen kartografischen Datensatzes (vgl. Kapitel 7.2 «Konzept und Definitionen» auf Seite 45). Der Geometriesatz s05, der im Massstab 1:416 667 vorliegt, wird durch einfaches numerisches Skalieren dem Massstab des Geometriesatzes s10 angepasst und so verschoben, dass beide Koordinatensysteme deckungsgleich werden. Die vollständige Kartengeometrie befindet sich nun im Ausgangsmassstab 1:833 333.

9.2.1 Anpassungen für die Karte im Ausgangsmassstab

Der Entscheidung, welche Objektgruppen im Zustand des Ausgangsmassstabes dargestellt werden sollen – wie schon erwähnt – einige Bedeutung zu. Diese Entscheidung muss man jetzt treffen, bevor im nächsten Schritt die Signaturmasse und die anderen Objektstile festgelegt werden. Die Objektstile können nur im Kontext der vollständigen Kartengeometrie – d. h. des richtigen Informationsgehaltes – festgelegt werden, weil deren Betonungen durch Grösse oder Farbe und deren anderen grafischen Variablen wie Form oder Art von Signaturen sehr stark von den umgebenden Objekten abhängen.

Für den Ausgangsmassstab ausgewählt werden diejenigen Objektgruppen, die dem Benutzer helfen, sich einen schnellen Überblick über die geografische Lage und gegebenenfalls das Kartenthema zu verschaffen. Wenn die Karte rein topografischer Natur ist, beinhaltet dies ein einfaches Gewässernetz, grobe politische Grenzen wie Landesgrenzen, grosse Siedlungsflächen, Autobahnen, wichtige Haupt- und Nebenstrassen und wichtige Eisenbahnlinien. Alle Objektgruppen entstammen natürlich dem Geometriesatz s10. Die Ebenen aller anderen Objektgruppen werden zu diesem Zeitpunkt ausgeblendet.

Bei allen eingeblendeten Objektgruppen werden jetzt die Objektstile überarbeitet und so angepasst, dass sich ein gutes Kartenbild ergibt. Betroffen sind die Wahl der Grösse der Signaturmasse, die Wahl von Schrift und Schriftgrösse für Labels, die Schriftplatzierung und die Wahl von Farben. Als Schrift kommt die *Berthold Formata* in den Schriftschnitten *Medium Condensed*, *Condensed* und *Light Condensed* zum Einsatz.

Sämtliche Anpassungen folgen den in Kapitel 4 «Gestaltung von Bildschirmkarten» auf Seite 17 beschriebenen Regeln. Das Resultat – ein möglichst harmonisches Kartenbild – ist das Bild, welches der Benutzer später im Ausgangsmassstab zu sehen bekommen wird. Es bleibt aber zu beachten, dass sowohl Signaturmasse als auch (und vor allem) die Labelplatzierung aufgrund der nachfolgenden Anpassungen später evt. wieder verändert werden müssen.

9.2.2 Festlegung der Skalierungsgrenzwerte

Um die Stile aller anderen Objektgruppen festlegen zu können, muss zumindest ungefähr bekannt sein, bei welchem Massstab sich der Informationsgehalt um wieviel erhöht. Es muss festgelegt werden, welche Objektgruppen welches Darstellungsgewicht haben (vgl. Kapitel 7.5.1 «Das Darstellungsgewicht von Objekten und Objektgruppen» auf Seite 50). Für jede Objektgruppe wird folglich ein Informationsgehalt-Grenzwert benötigt (vgl. Kapitel 7.5.2 «Informationsgehalt-Grenzwerte» auf Seite 50). Zu diesem Zeitpunkt ist noch nicht entscheidend, dass diese Grenzwerte absolut exakt stimmen. Wichtig ist, dass deren Grössenordnung vernünftig ist, die Feinjustierung kann später erst aufgrund des fertigen Kartenbildes vorgenommen werden.

Die Grenzwerte werden provisorisch nach rein mathematischen Gesichtspunkten gewählt. Die fixen Werte liegen bei 1:833 333 (Ausgangsmassstab) und 1:104 167 (maximale Vergrösserung). Den Wert der maximalen Vergrösserung erhält man, indem der Geometriesatz s05, der ja eigentlich einen Ausgangsmassstab von 1:416 667 (also $2 \times 1:833\ 333$) hat, um den Faktor 4 vergrössert wird. Die adaptive Karte soll zwischen diesen Grenzwerten jeden beliebigen Massstab annehmen können.

Der Wechsel von einem Geometriesatz zum nächsten erfolgt vorläufig bei 1:312 500, obwohl der Ausgangsmassstab von s05 bei 1:416 667 liegt. Da s10 auf maximal 1:208 333 vergrössert werden kann, überschneiden sich die Massstabsbereiche beider Geometriesätze. Der Grenzwert kann also irgendwo zwischen 1:416 667 und 1:208 333 gewählt werden, 1:312 500 ist der Mittelwert (siehe Abb. 31).

Hätte man sich statt für den gewählten Geometriesatz s05 für einen anderen entschieden, dessen Ausgangsmassstab sich direkt an den maximalen Massstab von s10 anschliesst (es wäre ein Originalmassstab von ca. 1:175 000 nötig gewesen), so wäre zwar ein grösserer Zoombereich bis zum Massstab 1:50 000 möglich gewesen, also eine deutliche Ausweitung des Anwendungsbereiches. Der Vorteil der Entscheidung für s05 ist aber, dass weder s10 noch s05 bis zu den Extremwerten skaliert werden müssen, da «Extremwert» bedeutet, dass die Qualität gerade noch toleriert werden kann, aber eigentlich schon nicht mehr wirklich gut ist. Mit s05 befindet man sich



Abb. 31 Provisorische Zoombereiche und Skalierungsgrenzwerte

immer auf der sicheren Seite. Ein fließender Übergang ist dadurch gewährleistet, dass der geometrische Charakter und die Anzahl der grafischen Elemente nur unmerklich ändern und vom Benutzer nicht als störend empfunden werden. Ein Wechsel auf 1:200 000 (als nächster ‹gerader› Massstab zu 1:175 000) hätte alle diese Nachteile mit sich gebracht.

9.2.3 Provisorische Festlegung der Informationsgehalt-Grenzwerte

Der Informationsgehalt wird durch Hinzufügen neuer Objektgruppen grösser, wobei erst leichte, dann schwerere Objektgruppen dazukommen (vgl. Kapitel 7.5 ‹Informationsdichte und Informationsgehalt› auf Seite 49). Die Auflistung in Tab. 7 gibt einen Überblick über die vorhandenen Objektgruppen, ihre Merkmale, ihre relative Dichte und ihr Darstellungsgewicht. Sie wird weiter unten ausführlich erläutert.

Objektgruppe	Rel. Dichte	Art	s10	s05	Merkmal
Ortschaften, gross	locker	Punkt	2	2	Objektgeometrie unverändert
Ortschaften, gross	dicht	Punkt	–	6	Neue Information
Ortschaften, mittelgross	locker	Punkt	3	3	Objektgeometrie unverändert
Ortschaften, mittelgross	dicht	Punkt	–	7	Neue Information
Ortschaften, klein	–	Punkt	–	8	Neue Information
Siedlungsflächen	locker	Label	0	0	Objektgeometrie unverändert
Siedlungsflächen	mittel	Label	1	1	Objektgeometrie unverändert
Siedlungsflächen	dicht	Label	–	5	Neue Information
Ortschaften, gross	locker	Label	2	2	Objektgeometrie unverändert
Ortschaften, gross	dicht	Label	–	6	Neue Information
Ortschaften, mittelgross	locker	Label	3	3	Objektgeometrie unverändert
Ortschaften, mittelgross	dicht	Label	–	7	Neue Information
Ortschaften, klein	–	Label	–	8	Neue Information
Hauptstrassen	–	Linie	0	4	Information detaillierter
Nebenstrassen	locker	Linie	1	4	Information detaillierter
Nebenstrassen	dicht	Linie	–	5	Neue Information
Hauptbahnlinien	–	Linie	0	4	Information detaillierter
Nebenbahnlinien	–	Linie	1	4	Information detaillierter
Autobahnen	–	Linie	0	4	Information detaillierter
Gewässernetz	locker	Linie	0	–	
Gewässernetz	mittel	Linie	1	–	
Gewässernetz	dicht	Linie	2	4	s05 mit locker, mittel, dicht
Gewässernetz	–	Linie	–	5	Neue Information
Landesgrenzen	–	Linie	0	4	Information detaillierter
Siedlungsflächen	locker	Fläche	0	–	
Siedlungsflächen	dicht	Fläche	1	4	s05 mit locker, mittel

Tab. 7 Die vorhandenen Objektgruppen und ihre Merkmale (Erläuterungen im Text)

Def. Relative Dichte, Objektklasse

Die *relative Dichte* (in Tab. 7 in der Spalte ‹Rel. Dichte›) beschreibt die Dichte einer Objektgruppe im Kontext ihrer verwandten Objektgruppen. Mehrere verwandte Objektgruppen, also diejenigen, die von der gleichen Art sind (in Tab. 7 repräsentiert durch den gleichen Namen), sich aber in ihrer relativen Dichte unterscheiden, bilden zusammen eine *Objektklasse*. Die re-

lative Dichte gibt also die Dichte einer Objektgruppe innerhalb ihrer Objektklasse wieder. Z. B. ist die Objektklasse Gewässernetz in s10 in den drei relativen Dichtestufen ‹locker›, ‹mittel› und ‹dicht› vorhanden, sie besteht also aus drei unterschiedlich dichten Objektgruppen.

Die relative Dichte hat keinen direkten Einfluss auf die absoluten Informationsgehalt-Grenzwerte, weil sie unabhängig vom Objektgewicht ist. Das Attribut ‹dicht› bedeutet also nicht, dass alle Objektgruppen, die mit ‹dicht› klassifiziert sind, gleichzeitig eingeblendet werden müssen.

Es wurde früher schon festgehalten, dass jede Objektgruppe von s10 in s05 ein Pendant haben muss (vgl. Kapitel 7.4.2 ‹Technische Qualität› auf Seite 48), da immer neue Information hinzukommt, aber nie bereits bestehende Information entfernt, sondern nur anders, also für den Masstab optimiert, dargestellt werden darf. Das s05-Pendant zu einer s10-Objektklasse muss gezwungenermassen immer eine einzelne Objektgruppe sein, welche alle Informationen der s10-Objektklasse zusammengefasst enthält: Alle Objektgruppen einer s10-Objektklasse müssen in s05 die gleiche relative Dichte haben und gleichzeitig dargestellt werden. Das Pendant zum Gewässernetz, das sich als s10-Objektklasse in die Objektgruppen ‹locker›, ‹mittel›, ‹dicht› aufteilt, besteht in s05 aus einer einzelnen Objektgruppe, die alle drei Dichtestufen vereint enthält. Sie wird bei grösserem Masstab durch eine weitere Objektgruppe ergänzt, die das Gewässernetz mit der relativen Dichte ‹sehr dicht› enthält (siehe Abb. 32) und bildet somit eine neue, aus zwei Objektgruppen bestehende Objektklasse.

Das Darstellungsgewicht wird in einem *Informationsgehalt-Index* ausgedrückt, der in den Spalten s10 bzw. s05 dargestellt ist:

- ◆ Eine Null bedeutet, dass diese Objektgruppe bereits im Ausgangsmasstab dargestellt werden soll.
- ◆ Jede Erhöhung des Index um 1 hat eine Zunahme des Informationsgehaltes zur Folge. Die Objektgruppen werden in dieser Reihenfolge eingeblendet.
- ◆ Hat die Spalte s10 keinen Eintrag, dann existiert die Objektgruppe in der entsprechenden Dichte nicht, in diesem Fall kommt in s05 neue Information dazu.
- ◆ Hat die Spalte s05 keinen Eintrag, so wird von s10 nach s05 eine Objektklasse zu einer Objektgruppe zusammengefasst.
- ◆ Haben die Indizes in s10 und s05 den gleichen Wert, dann kann in beiden Fällen die gleiche Objektgeometrie verwendet werden, die Objekte müssen also nicht mehrfach vorhanden sein. Dies ist immer dann der Fall, wenn es sich bei den betroffenen Objekten um Punktsignaturen oder Labels handelt, die nicht generalisiert, sondern nur in der Dichte verändert werden können.
- ◆ Lineare oder flächenhafte Objekte, die von s10 auf s05 geometrisch verändert werden, können dagegen nie den gleichen Index haben. Ihre Objektgeometrie muss stets ausgewechselt werden und in einer detaillierteren Version vorhanden sein.

Die Informationsgehalt-Grenzwerte der einzelnen Objektgruppen hängen direkt mit dem Informationsgehalt-Index zusammen. Da diese Grenzwerte nicht mathematisch begründet quantifiziert werden können, müssen sie aufgrund der Resultate und durch Probieren letztendlich genau ermittelt werden (vgl. Kapitel 7.5.2 ‹Informationsgehalt-Grenzwerte› auf Seite 50).

An dieser Stelle erfolgt provisorisch eine rein arithmetische Festlegung:

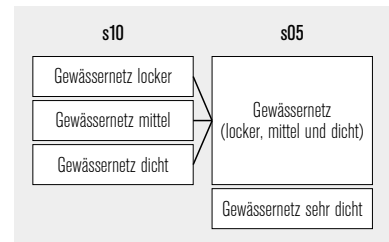


Abb. 32 Zusammenfassung von Objektgruppen einer Objektklasse

Def. Informationsgehalt-Index

s10	s05	
0		1:833 333 – 1.0
1		1.41
2		1.82
3		2.23
	4	1:312 500 – 2.65
	5	3.72
	6	4.79
	7	5.86
	8	6.93
		1:104 167 – 8.0

Abb. 33 Provisorische Festlegung der Informationsgehalt-Grenzwerte

- ◆ Bei insgesamt neun Informationsdichtestufen (neun Indizes) werden vier für den Bereich bis zum Wechsel von s10 auf s05 gewählt. Dieser Wechsel erfolgt beim Massstab 1:312 500 (vgl. Kapitel 9.2.2 «Festlegung der Skalierungsgrenzwerte» auf Seite 65), also bei einem Skalierungsfaktor von 2.65. Entsprechend muss der Skalierungsfaktor bei 4 Grenzwerten jeweils um 0,41 (1,65/4) erhöht werden. In vier Abschnitte aufgeteilt, liegen die Grenzwerte bei 1,41, 1,82 und 2,23.
- ◆ Im Bereich von s05 werden fünf Grenzwerte benötigt, die einen Skalierbereich von 2,65 bis 8 abdecken. Die Grenzwerte werden jeweils um 1,07 erhöht und bei 3,72, 4,79, 5,86 und 6,93 festgesetzt (siehe Abb. 33).

Mit Hilfe der provisorischen Festlegung für die Informationsgehalt-Grenzwerte kann die Karte weiter bearbeitet werden. Die nächsten wichtigen Schritte umfassen die Anpassung der Signaturmasse auf alle Massstäbe und die im Hinblick auf die dynamische Platzierung optimierte Kartenbeschriftung.

9.2.4 Anpassung der Signaturmasse auf alle Massstäbe

Die Signaturmasse in ihrer Grösse anzupassen, ist mit beträchtlichem Aufwand verbunden. Für jeden Massstab, bei dem der Informationsgehalt grösser wird, also für jeden Informationsgehalt-Grenzwert, muss die Darstellung, wie sie der Benutzer am Schluss zu sehen bekommt, simuliert werden.

In *Illustrator* muss das Bild so gezoomt werden, dass die Darstellung gerade einem Informationsgehalt-Grenzwert entspricht. Soll z. B. der Grenzwert 1,41 simuliert werden, so muss die Darstellung in der Vergrösserung 141% erfolgen.

Um abschätzen zu können, wie die Signaturmasse im Kontext der ganzen Abbildung wirken, werden jetzt alle Ebenen eingeblendet, die Objektgruppen enthalten, welche bei diesem Grenzwert sichtbar sein müssen.

Die Signaturmasse aller Objektgruppen, die bereits zum Zeitpunkt des letzten tieferen Grenzwerts eingeblendet waren, sind zwar schon optimiert. Da sie aber proportional zum Zoomfaktor vergrössert wurden, sind sie jetzt zu gross und müssen gemäss den Ausführungen in Kapitel 7.6 «Anpassung der Signaturmasse: Liniensignaturen» auf Seite 51 temporär, d. h. für die Dauer der Anpassungen, mit neuen Signaturmassen versehen werden. Die Signaturmasse werden also nach Funktion (11) neu berechnet und durch den Zoomfaktor dividiert, weil dieser sich ja auf die ganze Darstellung bezieht. Die ursprünglichen Signaturmasse müssen aber unbedingt festgehalten werden, da die neuen nur vorübergehend benötigt werden.

Jetzt können die Signaturmasse für die neu hinzugekommenen Objektgruppen so gewählt werden, damit ein ausgewogenes Kartenbild entsteht. Dieser Prozess muss für jeden Informationsgehalt-Grenzwert wiederholt werden und ist entsprechend aufwändig.

9.2.5 Anpassung der Kartenbeschriftung

Gleichzeitig mit der Anpassung der Signaturmasse kann auch die Kartenbeschriftung optimiert werden. Dieser Vorgang ist noch um einiges arbeitsintensiver, da man die adaptive Platzierung berücksichtigen muss und Fehler im Resultat sehr gut sichtbar sein werden.

Wie in Kapitel 7.8.3 «Adaptive Platzierung der Kartenbeschriftung» auf Seite 57 beschrieben, muss jedes Label einem Quadranten zugeordnet sein. Am einfachsten und schnellsten kann dies geschehen, wenn für jeden Quadranten eine eigene Ebene angelegt wird. Alle Labels auf einer Ebene werden dann um den gleichen Betrag in x- und y-Richtung verschoben.

Zum Zeitpunkt der Erreichung jedes Informationsgehalt-Grenzwertes kann die Platzierung der Labels nach den bekannten kartografischen Regeln angepasst werden (vgl. Kapitel 4.2.8 ‹Anordnung der Kartenbeschriftung› auf Seite 20). Gleichzeitig werden sie nach Quadranten auf die vier vorher erstellten Ebenen verteilt. Die Schriftgrösse wird – genau wie in Kapitel 9.2.4 die Signaturmasse – dem Massstab entsprechend gewählt.

Nachdem die Platzierung der Kartenbeschriftung aber erfolgt ist, müssen die Labels um die negativen Beträge von Tab. 6 auf Seite 58 verschoben werden, weil die Verschiebung der Labels – gleichgültig, zu welchem Zeitpunkt sie eingeblendet werden – immer vom Ausgangsmassstab der adaptiven Karte aus geht. Die Labels befinden sich jetzt in ihren jeweiligen Ausgangspositionen.

Die Kartenbeschriftung hängt noch viel stärker vom Kontext der umgebenden Grafik ab als die Signaturmasse; insbesondere dürfen einzelne Labels sich nicht überlappen. Bei jedem Informationsgehalt-Grenzwert müssen also die schon bestehenden Labels gemässe Tab. 6 auf Seite 58 provisorisch an die neue Position verschoben werden, damit die hinzukommenden Labels entsprechend positioniert werden können. Nach Beendigung der Platzierung müssen die Labels aber in jedem Fall – wie oben schon beschrieben – wieder in ihre Ausgangsposition und Ausgangsgrösse gebracht werden.

9.3 Bezeichnung von Ebenen

Wird eine Karte aus *Illustrator* in das SVG-Format exportiert, dann werden die Ebenennamen in SVG als ID übernommen. Grundsätzlich können beliebige Zeichen für den Namen verwendet werden, Sonderzeichen wie Leerzeichen oder Umlaute werden aber nicht so umgewandelt, dass sie gut lesbar sind. Deshalb werden bei der Wahl von Ebenennamen nur Buchstaben, Zahlen und *Underbars* verwendet.

Einzelne Objekte werden in SVG mit Hilfe der ID angesprochen. Es ist daher äusserst wichtig, dass die Namensgebung der Ebenen mit einem einheitlichen System erfolgt. Die ID soll folgende Informationen enthalten:

- Zugehörigkeit zum Geometriesatz
- relative Dichte
- Name der Objektgruppe

Die Zugehörigkeit zum Geometriesatz und die relative Dichte heissen zusammen genommen *absolute Dichte* und ergeben die gleiche Information, die auch im Informationsgehalt-Index enthalten ist (vgl. Kapitel 9.2.3 ‹Provisorische Festlegung der Informationsgehalt-Grenzwerte› auf Seite 66). Man könnte daher einfach diesen Index verwenden. Beide Informationen einzeln, aber zusammengefasst zu behandeln, hat aber den Vorteil höherer Flexibilität: Wird später ein weiterer Dichtegrenzwert eingefügt oder soll einer weggelassen werden, dann muss nicht alles neu nummeriert werden.

Def. absolute Dichte

Die Zugehörigkeit zum Geometriesatz muss ausserdem die Information enthalten, ob die betroffene Ebene nur ein Element eines Geometriesatzes ist (z. B. Gewässernetz), von mehreren Geometriesätzen verwendet werden kann (z. B. Ortssignaturen) oder sogar immer eingeblendet werden muss (z. B. Hintergrund). Auch dies wäre nicht möglich, würde man den Index verwenden.

Um später die Handhabung in der Programmierumgebung zu vereinfachen und Missverständnisse zu vermeiden, werden konsequent nur englische Bezeichnungen verwendet.

Die in den Ebenennamen verwendeten Bezeichnungen und ihre Bedeutung sind in Tab. 8 aufgelistet.

Informationsteil	Bez.	Bedeutung
Zugehörigkeit zum Geometriesatz	s10 s05 s00 s__	Geometriesatz s10 Geometriesatz s05 beide Geometriesätze immer eingeblendet
Relative Dichte	l m d x y	locker (<i>light</i>) mittel (<i>middle</i>) dicht (<i>dense</i>) extradicht 1 (<i>extra dense 1</i>) extradicht 2 (<i>extra dense 2</i>)
Name der Objektgruppe	f... o...	Signaturfüllung (<i>filling</i>) Signaturkontur (<i>outline</i>)

Tab. 8 In den Ebenennamen verwendete Bezeichnungen und ihre Bedeutung

Ein Ebenenname bzw. eine ID kann also z. B. so aussehen:

s10_l_fMainroad.

Diese Ebene gehört dem Geometriesatz `<s10>` an, hat die relative Dichte `<l>` (*light*) und den Namen `<fMainroad>`. Das `<f>` steht für *filling* und deutet darauf hin, dass es sich um die Signaturenfüllung handelt.

9.4 Aufbau der SVG-Anwendung

Der präparierte kartografische Datensatz kann nicht einfach exportiert werden. Auch an der SVG-Anwendung selber muss man einige Anpassungen vornehmen. Der Datensatz wird zu diesem Zweck provisorisch ins SVG-Format exportiert. Das produzierte Dokument kann nun in einem Texteditor geöffnet und abgeändert werden.

9.4.1 Objektstile in der Document Type Definition DTD

Illustrator bietet die Möglichkeit, ein Dokument so zu exportieren, dass Objektstile als *entity reference* exportiert werden (vgl. Kapitel 6.1.3 `<Physische Struktur einer XML-Anwendung>` auf Seite 33). Alle Objektstile werden dann in der DTD gesammelt, die sich physisch zwar in der SVG-Anwendung, aber unmittelbar vor dem eigentlichen SVG-Dokument befindet (das SVG-Dokument besteht eigentlich nur aus dem Teil, der durch die *tags* `<svg>` und `</svg>` ausgezeichnet ist; vgl. Kapitel 6.2.3 `<Struktur eines SVG-Dokumentes>` ab Seite 38).

Der Vorteil einer solchen Methode ist, dass das Dokument schneller geladen wird und kleiner ist, als wenn die Objektstile *in line* abgelegt würden. Objektstile werden von den jeweiligen Objektgruppen aus referenziert; das SVG-Dokument besteht im Wesentlichen nur aus der Objektgeometrie.

Da Objektgeometrie und Objektstile bei einer adaptiven Karte stets voneinander getrennt behandelt werden, ist die physische Trennung beider sehr sinnvoll. Die Definitionen für alle Objektstile sollen deshalb in der DTD des SVG-Dokumentes abgelegt werden. Die Exportfunktion von *Illustrator* produziert aber leider generische und nicht sehr aussagekräftige Namen für die *entities*, die für das vorliegende Projekt in dieser Form nicht brauchbar sind. Eine Objektgruppe ist ausserdem standardmässig oft auf mehrere Stile referenziert, die sich ergänzen, womit der Überblick verloren geht.

*Physische Trennung von
Objektgeometrie und Objektstilen*

Die *entities* müssen also in Handarbeit zusammengestellt werden. An die Stilsammlung in der DTD werden daher die folgenden wichtigen Anforderungen gestellt:

- ◆ Jede Objektgruppe darf nur einen Objektstil verwenden, der alle notwendigen Definitionen enthält.
- ◆ Ein Objektstil muss von mindestens einer, kann aber von beliebig vielen Objektgruppen verwendet werden.
- ◆ Der Name des Objektstils muss aus dem Namen der Ebene ableitbar sein, d. h. er muss dem Namen der Objektgruppe entsprechen.

Eine *entity* ist aus mindestens einem, kann aber aus beliebig vielen Stileigenschaften aufgebaut sein. Die für Bildschirmkarten wichtigsten sind in Tab. 9 zusammengestellt, weitere findet man in der SVG-Spezifikation (Bowler et al., 2000). Farben werden als RGB-Werte im Hexadezimalsystem beschrieben, Strichstärken in px oder, falls spezifiziert, in einer beliebigen anderen Einheit angegeben, die Transparenz kann Werte zwischen 0 (vollständig transparent) und 1 (vollständig opak) annehmen.

Eine Stildefinition kann z. B. so aussehen:

```
<!ENTITY mainRailway "fill:none;stroke:#FF0000;stroke-width:1.5;">
```

Dieser Objektstil hat den Namen `mainRailway`, keine Füllfarbe (also nur Kontur), die Konturfarbe `FF0000` (Rot) und die Strichstärke 1,5 px.

Ein anderes Beispiel beschreibt einen Teil der Kartenbeschriftung:

```
<!ENTITY labelSettlement1 "font-family:'Formata-Cond';font-size:14;">
```

Dieser Objektstil für die Objektgruppe `labelSettlement1` benutzt die Schrift Formata-Condensed in der Grösse 14 px. Die Schrifteinbindung wird im Kapitel 9.4.3 beschrieben.

9.4.2 Attributes des SVG-Tag

Ein SVG-Dokument hat *attributes*, welche die Grösse der Grafik (*width*, *height*) und den sichtbaren Bereich (*viewport*, vgl. Kapitel 6.2.5 `⌈Koordinatensysteme und Transformationen⌋` auf Seite 39) festlegen. Das `<svg>`-tag für die beschriebene Anwendung muss also mit entsprechenden *attributes* erweitert werden:

```
<svg width="528" height="423" viewBox="18 12 500 400">
```

Die Kartengeometrie ist 528 px auf 423 px gross. Davon sichtbar ist der durch die *viewBox* definierte Ausschnitt.

9.4.3 Inhalt des SVG-Dokumentes

Der grösste Teil des SVG-Dokumentes macht die Kartengeometrie aus. Um diesen Teil später einfach und eindeutig ansprechen zu können, wird die ganze Kartengeometrie zu einer einzigen Gruppe zusammengefasst, die den Namen *mapPart* enthält. Die vollständige Kartengeometrie des adaptiven Datensatzes wird also durch die tags `<g id="mapPart">` und `</g>` eingerahmt.

Neben dem Geometrieteil besteht das SVG-Dokument aus einem *style-Element*, das die Schriftdefinitionen beinhaltet. Schriften, die sich in externen Dokumenten befinden (vgl. Kapitel 8.1.1 `⌈Produktion eines SVG-Dokumentes mit Illustrator⌋` auf Seite 59), können mit Hilfe des *style-Elementes* eingebettet werden. Die entsprechenden Schriftdokumente wurden durch

Stileigenschaft	Funktion
<i>fill</i>	Füllfarbe
<i>fill-opacity</i>	Transparenz
<i>stroke</i>	Strichfarbe
<i>stroke-width</i>	Strichstärke
<i>font-family</i>	Schriftfamilie
<i>font-size</i>	Schriftgrösse

Tab. 9 Die wichtigsten Stileigenschaften für Bildschirmkarten

⌈style-Element

den provisorischen Export produziert. Zwischen den *tags* `<style type="text/css">` werden Schriften wie folgt eingebettet:

```
@font-face{font-family:'Formata-Regular';src:url(FormRg.cef)}
```

Die Schriftfamilie heisst `Formata-Regular`, und das Dokument mit ihrer Definition hat den Namen `FormReg.cef`.

9.5 Repetitive Anpassung der Kartengeometrie

Die abgeänderte SVG-Anwendung erfüllt jetzt formal alle notwendigen Anforderungen; nur die Kartengeometrie wurde bisher nicht berücksichtigt. Es ist wichtig, dass diese immer wieder verändert werden kann, und zwar mit verhältnismässig geringem Aufwand.

Der kartografische Datensatz muss also einfach und schnell immer wieder in die SVG-Anwendung eingefügt werden können, und dabei soll nur die Kartengeometrie berücksichtigt werden. Die durch den Export entstehenden *style-attributes*, die alle SVG-Elemente erhalten, müssen im Texteditor entfernt werden. Am einfachsten geschieht dies, wenn jeweils möglichst viele SVG-Elemente (also Objektgruppen) die gleichen Stile verwenden. Es ist dann möglich, mit den Standardbefehlen *find/replace*, die alle Texteditoren kennen, jede Stildefinition zu entfernen.

9.5.1 Export aus Illustrator

Um zu erreichen, dass möglichst viele Objektgruppen die gleichen Stile verwenden, müssen in *Illustrator* alle Objekte – gleichgültig, ob Flächenobjekt, lineares Objekt, Punktsignatur oder Schrift – mit der gleichen Flächenfarbe versehen werden. Wird die Flächenfarbe auf schwarz gestellt, hat dies den Vorteil, dass der Stil `fill` gleich ganz weggelassen wird, da Objekte, deren Füllfarbe nicht definiert ist, immer schwarz dargestellt werden.

Genau aus dem gleichen Grund soll die Kontur für alle Objekte entfernt werden: Objekte ohne Kontur erhalten in SVG keine Konturdefinition, die Stileigenschaften *stroke* und *stroke-width* werden einfach weggelassen.

Schrifteigenschaften dürfen hingegen nicht verändert werden, da die Veränderung sowohl der Schriftfamilie als auch der Schriftgrösse Einfluss auf die Kartengeometrie haben.

Der entsprechend präparierte Datensatz kann in dieser Form ins SVG-Format exportiert werden.

9.5.2 Bearbeitung im Texteditor

Mit *find/replace* müssen alle Stildefinitionen, also alle *tag-attributes*, in denen `<style=>` vorkommt, entfernt werden. Ausserdem müssen alle Teile des Dokumentes, die sich ausserhalb der *tags* `<svg>` und `</svg>` befinden, gelöscht werden. Übrig bleibt die reine Kartengeometrie.

Mit *copy/paste* kann die neue Kartengeometrie nun jederzeit in den `mapPart`-Teil der SVG-Applikation eingefügt werden.

10 Das Steuerprogramm für die adaptive Karte

Damit die adaptive Karte interaktiv gesteuert werden kann, wird ein Programm benötigt, welches die Steuerung der Karte aus einem Webbrowser ermöglicht. Das Steuerprogramm muss die folgenden Abläufe kontrollieren:

Beim Starten der adaptiven Karte:

- ◆ Die richtigen Objektgruppen für den Ausgangsmassstab müssen eingeblendet werden.
- ◆ Die richtigen Objektstile müssen den Objektgruppen zugewiesen werden.

Beim Zoomvorgang:

- ◆ Die richtigen Objektgruppen müssen ein- bzw. ausgeblendet werden.
- ◆ Die Signaturmasse müssen dem Massstab angepasst werden.
- ◆ Die Labels und Punktsignaturen müssen richtig positioniert werden.

10.1 Notation

Auf den nächsten Seiten ist erklärt, wie dieses Steuerprogramm aufgebaut ist. Es ist in der Sprache *JavaScript* geschrieben (vgl. Kapitel 6.1.7 *Interaktivität mit JavaScript* auf Seite 36) und direkt in den HTML-Code der Homepage eingebunden. Die Struktur des Steuerprogrammes ist nach der Methode des *Jackson Structured Programming* JSP notiert, auf Angabe des Codes wird zumeist verzichtet.

Beim Softwareentwurf nach JSP werden logische Abhängigkeiten in Form von attributierten Bäumen dargestellt. Solche Bäume werden aus atomaren Komponenten (Blätter des Baumes) mit den drei Strukturtypen *Sequenz* (Folge, Reihung), *Iteration* (Wiederholung, Schleife) und *Selektion* (Auswahl, Verzweigung) aufgebaut (Kilberth, 1989).

Diese Abläufe können auch in Worte gefasst werden, wobei man bestimmte Ausdrücke verwendet, die jeweils zu einer der Komponenten passen.

- ◆ Eine *Sequenz* umfasst eine oder mehrere Komponenten, die genau einmal in der angegebenen Reihenfolge auftreten (siehe Abb. 34). Die Reihenfolge ist von links nach rechts festgelegt.

Wörter und Wendungen, die zu Sequenzen passen, sind:

- zuerst, dann, danach
- Folge von
- vorausgehend, folgend
- vorher, nachher
- erster, letzter

- ◆ Eine *Iteration* besteht aus einer Komponente, die nullmal, einmal oder mehrmals auftreten kann (siehe Abb. 35). Die atomare Komponente einer Iteration (nicht die Iteration selber) wird durch einen Stern markiert, wobei der Stern immer nach oben wirkt.

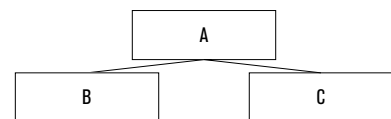


Abb. 34 Sequenz (A ist die Sequenz. B und C sind atomare Komponenten dieser Sequenz)

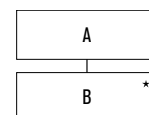


Abb. 35 Iteration (A ist die Iteration. B ist die atomare Komponente dieser Iteration)

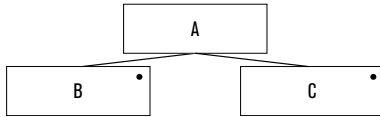


Abb. 36 Selektion (A ist die Selektion. B und C sind die Komponenten dieser Selektion)

Wörter und Wendungen, die zu Iterationen passen, sind:

- für alle gilt
- alle, viele, einige
- eine Anzahl von
- eine Gruppe, eine Menge
- wiederholt, mehrfach

- ◆ Eine *Selektion* besteht aus zwei oder mehreren Komponenten, von denen genau eine vorkommt (siehe Abb. 36). Die atomaren Komponenten einer Selektion (nicht die Selektion selber) werden durch einen Kreis hervorgehoben, wobei der Kreis immer auf die Komponente darüber wirkt.

Wörter und Wendungen, die zu Selektionen passen, sind:

- wenn, falls, andernfalls
- entweder, oder
- kann sein, tritt auch auf, ist auch erlaubt
- alternativ, an der Stelle von
- möglicherweise, kann fehlen, manchmal

Für das Steuerprogramm wird auf den nächsten Seiten jeder Programmteil sowohl ausformuliert, als auch als JSP-Baum dargestellt. Die ausformulierten Elemente werden durch etwas kleinere Schrift jeweils grafisch hervorgehoben.

Zur Verknüpfung der bereits erwähnten zu benutzenden Wörter und Wendungen werden die Wörter `<und>` und `<oder>` im boole'schen Sinn verwendet. Ein `<und>` bedeutet, dass die Bedingungen auf beiden Seiten des `<und>` erfüllt sein müssen. Immer, wenn mehrere Bedingungen aneinander gereiht werden, müssen sie durch ein `<und>` verknüpft werden. Es kann daher manchmal zu recht langen Sätzen mit mehreren `<und>` hintereinander kommen.

JSP-Bäume werden konsequent in englisch beschriftet, da dies die spätere Umsetzung in *JavaScript*-Code erleichtert.

10.2 Starten des Steuerprogramms für die adaptive Karte

Beim Starten des Steuerprogramms für die adaptive Karte muss das Programm vor allem dafür sorgen, dass die richtigen Objekte eingblendet und mit Objektstilen versehen werden. Vorher muss das SVG-Dokument aber so bereinigt sein, dass das Steuerprogramm auch fehlerfrei darauf zugreifen kann.

10.2.1 White Space Handling

Wenn XML-Anwendungen editiert werden, ist es meistens üblich, das Dokument mit Leerschlägen und Tabulatoren so zu gliedern, dass es besser lesbar wird. Diese Leerschläge, sogenannte *white spaces*, sollen typischerweise für die Benutzerin nicht sichtbar sein, da sie nur dem Autoren zur besseren Übersicht dienen. Deshalb werden mehrere Leerschläge hintereinander in der Regel durch den XML-Parser automatisch auf nur einen einzelnen Leerschlag reduziert (Bray et al., 1998).

White spaces hat die vorliegende SVG-Anwendung immer zwischen einzelnen Objektgruppen (siehe Abb. 37): Mindestens vor jedem `<g>`-tag steht ein Tabulator zur optischen Gliederung. Diese Tabulatoren werden zwar durch den Parser in einen einzelnen *white space* umgewandelt, aber nicht vollständig entfernt.

```
<svg>
...
  <g id="s05_l_riverClass1">
    <path d="..." />
    <path d="..." />
  </g>
  <g id="s05_l_riverClass2">
    <path d="..." />
    <path d="..." />
  </g>
...
</svg>
```

Abb. 37 Ausschnitt aus dem SVG-Dokument

Ein *white space* ist immer ein *node* und wird sowohl vom DOM als auch vom Parser als solcher behandelt. Ein *white space* hat somit den gleichen Stellenwert wie eine Objektgruppe. Aus diesem Grund müssen, bevor das Programm korrekt ablaufen kann, während dem Ladevorgang sämtliche *white spaces* zwischen den Objektgruppen entfernt werden.

In Worten gefasst, macht dieser Programmteil also folgendes:

(Abb. 38) Für alle *nodes* gilt (Iteration): Falls der *node* vom Typ *text-node* ist (Selektion), dann muss er entfernt werden (Sequenz).

Abb. 38 zeigt, wie das *white space handling* in der JSP-Notation strukturiert dargestellt werden kann. Der grosse Vorteil der JSP-Notation ist, dass man den Ablauf mehr oder weniger problemlos in Code umsetzen kann:

```
for (var i=0;i<mapGeometry.getChildNodes().getLength();i++)
{
  if (mapGeometry.getChildNodes().item(i).getNodeTypes() == 3)
    mapGeometry.getChildNodes().item(i).getParentNode().removeChild(mapGeometry.~
                                                                getChildNodes().item(i));
}
```

`mapGeometry` ist eine Referenz auf die Kartengeometrie, `getChildNodes()`, `item()`, `getNodeTypes()`, usw. sind Methoden des SVG-DOM, mit deren Hilfe auf das SVG-Dokument zugegriffen wird.

10.2.2 Einblenden der Objektgruppen für den Ausgangsmassstab

Das notwendige Kriterium zum Einblenden der Objektgruppen für den Ausgangsmassstab ist die absolute Dichte (vgl. Kapitel 9.3 «Bezeichnung von Ebenen» auf Seite 69).

(Abb. 39) Für alle Objektgruppen des Kartenteiles gilt (Iteration): Falls die absolute Dichte die Objektgruppe als sichtbar deklariert (Selektion): Weise ihr Objektstile zu (Sequenz).

Andernfalls (Selektion): Mache die Objektgruppe unsichtbar (Sequenz).

Sowohl das Zuweisen der Objektstile (*append styles*) als auch das Unsichtbarmachen einer Objektgruppe (*remove visibility*) geschehen sinnvollerweise in eigenen Funktionen, die später wieder verwendet werden können.

10.2.3 Zuweisen von Objektstilen: Die Funktion `appendStyle`

Die Funktion zur Zuweisung der Objektstile hat den Namen `appendStyle`. Sie kann mit diesem Namen jederzeit von anderen Funktionen aus aufgerufen werden.

Im Prinzip könnten die Objektstile immer wieder aus den Stil-*entities* in der DTD ausgelesen werden (vgl. Kapitel 9.4.1 «Objektstile in der Document Type Definition DTD» auf Seite 70). Dazu wird jedesmal, wenn eine solche Stilinformation benötigt wird, ein *entityReference-node* produziert, der dann die entsprechenden Angaben liefert. Mit der Zeit entstehen aber beliebig viele *nodes*, welche das SVG-Dokument vergrössern und langsam machen. Um dies zu verhindern, ist es sinnvoll, einmal ausgelesene Objektstile in einem *comment-node* abzulegen, welcher der Objektgruppe zugeordnet ist, und auf den immer wieder zugegriffen werden kann.

(Abb. 40) Falls die Objektgruppe noch keinen *comment node* mit der Stildefinition enthält:

- † Extrahiere zuerst den Namen der *entity* aus der ID;
- † konstruiere dann mit diesem Namen einen *entityReference-node*;
- † lies dann den Wert der *entity* aus;
- † produziere dann einen neuen *comment-node* mit dem Wert der *entity*;

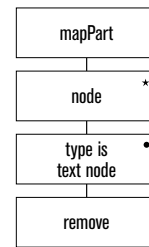


Abb. 38 Teilfunktion zur Entfernung überflüssiger *text nodes*

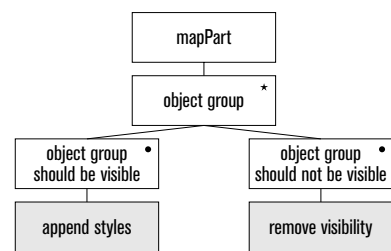


Abb. 39 Teilfunktion zur Einblendung der richtigen Objekte

- ♦ hänge ihn dann an die Objektgruppe an;
- ♦ füge dann die Objektstile der Objektgruppe an.

Andernfalls (falls die Objektgruppe bereits einen *comment-node* mit der Stildefinition besitzt) lies den Wert der *entity* aus dem *comment-node* aus.

Danach füge die Objektstile der Objektgruppe an.

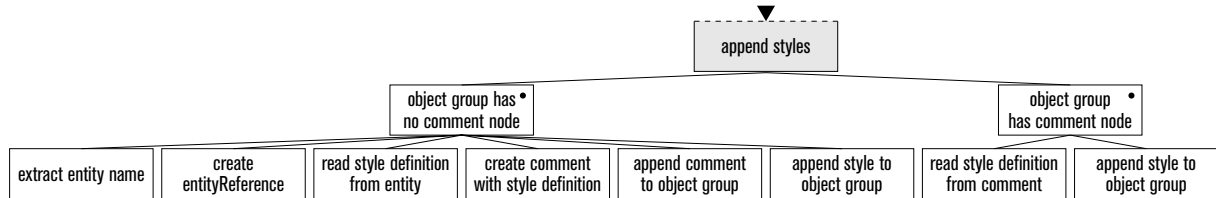


Abb. 40 Die Funktion *appendStyle*

10.2.4 Unsichtbarmachen von Objektgruppen: Die Funktion *removeVisibility*

Einzelne Objektgruppen werden mit der Funktion *removeVisibility* unsichtbar gemacht. SVG kennt zwei Möglichkeiten, um die Sichtbarkeit von Objekten zu kontrollieren, nämlich *display* und *visibility* (vgl. Kapitel 6.2.8 «Kontrolle über die Sichtbarkeit» auf Seite 40). *Display* hat vor allem den Vorteil sehr viel kürzerer Ladezeit, weil Objekte mit dem *attribute* `<display:none>` beim Rendern nicht berücksichtigt werden, und wird deshalb hier verwendet.

(Abb. 41) Füge der Objektgruppe das *attribute* an, welches sie als nicht sichtbar deklariert.

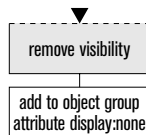


Abb. 41 Die Funktion *removeVisibility*

Eine weitere Möglichkeit, Objektgruppen auszublenden, wäre es, die einzelnen *nodes* im SVG-Dokument physisch zu verschieben. Dabei würden alle *nodes* in einem `<defs>`- oder `<symbol>`-Element (vgl. Kapitel 6.2.3 «Struktur eines SVG-Dokumentes» auf Seite 38) abgelegt und bei Gebrauch in den (sichtbaren) *mapPart* verschoben. Der Nachteil dieser Methode ist aber, dass ziemlich viel Rechenleistung benötigt wird.

10.3 Der Zoomprozess

Bei der Änderung des Zoomfaktors müssen alle Objektgruppen grundsätzlich auf zwei Arten angepasst werden: Einerseits sollen sämtliche Signaturmasse an den Zoomfaktor angepasst werden, und andererseits wird die Objektgruppe proportional zum Zoomfaktor vergrößert. Jede Objektgruppe durchläuft diesen Prozess einzeln.

Die Anpassung der Signaturmasse lässt sich in drei von der Informationsdichte abhängige Fälle gliedern, die das Steuerprogramm jeweils unterschiedlich behandeln muss:

1. Die Dichte wird nicht verändert.
2. Die *relative* Dichte wird verändert, d. h. ohne Wechsel des Geometriesatzes (vgl. Kapitel 9.3 «Bezeichnung von Ebenen» auf Seite 69).
3. Der Geometriesatz wird ausgewechselt.

Drei einzelne Abläufe müssen also entwickelt werden, welche, jeweils für ihr Dichteverhalten, das Hinzufügen bzw. Weglassen von Objektgruppen und die Anpassung der Signaturmasse übernehmen. Sie werden bedingt, d. h. je nach Verhalten der Dichte, aufgerufen.

(Abb. 42) Für alle Objektgruppen gilt:

Passe die Signaturmasse an.

- † Falls die Dichte nicht verändert wird, und falls der Objektstil der Objektgruppe eines der adaptiven Signaturmasse (also Strichstärke oder Schriftgröße) enthält: Passe die adaptiven Signaturmasse an.
- † Falls die relative Dichte verändert wird, und falls die Objektgruppe die richtige relative Dichte hat, und falls die Objektgruppe nicht bereits eingeblendet ist: Weise der Objektgruppe die Objektstile zu und mache sie somit sichtbar; andernfalls (d. h. falls die Objektgruppe nicht die richtige relative Dichte hat), aber nur dann, wenn der aktuelle Skalierungsfaktor kleiner als der letzte ist (also wenn ausgezoomt wird, denn nur dann darf Information ausgeblendet werden): Mache die Objektgruppe unsichtbar.

Falls der Objektstil der Objektgruppe eines der adaptiven Signaturmasse (also Strichstärke oder Schriftgröße) enthält: Passe die adaptiven Signaturmasse an.

Anders ausgedrückt: Wenn beim Einzoomen die relative Dichte verändert wird, kommen immer neue Objektgruppen hinzu, alle bereits sichtbaren Objektgruppen bleiben auch eingeblendet. Beim Auszoomen werden hingegen einzelne Objektgruppen, deren Dichte zu hoch ist, ausgeblendet.

- † Falls der Geometriesatz ausgewechselt wird, und falls die Objektgruppe den richtigen Geometriesatz hat und falls die Objektgruppe die richtige relative Dichte hat oder der Skalierungsfaktor verkleinert wird (bei Verkleinerung des Skalierungsfaktors müssen alle Objektgruppen des neuen Geometriesatzes eingeblendet werden): Weise der Objektgruppe die Objektstile zu und mache sie somit sichtbar; andernfalls (falls die Objektgruppe nicht dem richtigen Geometriesatz angehört): Mache die Objektgruppe unsichtbar.

Falls der Objektstil der Objektgruppe eines der adaptiven Signaturmasse (also Strichstärke oder Schriftgröße) enthält: Passe die adaptiven Signaturmasse an.

Anders ausgedrückt: Wenn beim Einzoomen der Geometriesatz ausgewechselt wird, dann müssen alle Objektgruppen, die zur Zeit eingeblendet sind (mit Ausnahme derjenigen, die Teil beider Geometriesätze sind), ausgeblendet werden. Neu eingeblendet werden diejenigen Objektgruppen, deren relative Dichte für den neuen Massstab passt. Beim Auszoomen müssen hingegen alle Objektgruppen mit der Dichte des neuen Massstabs eingeblendet werden.

Nach der Anpassung der Signaturmasse einer Objektgruppe wird diese mit Transformationen im Massstab verändert:

Transformiere die Objektgruppe proportional zum Skalierungsfaktor.

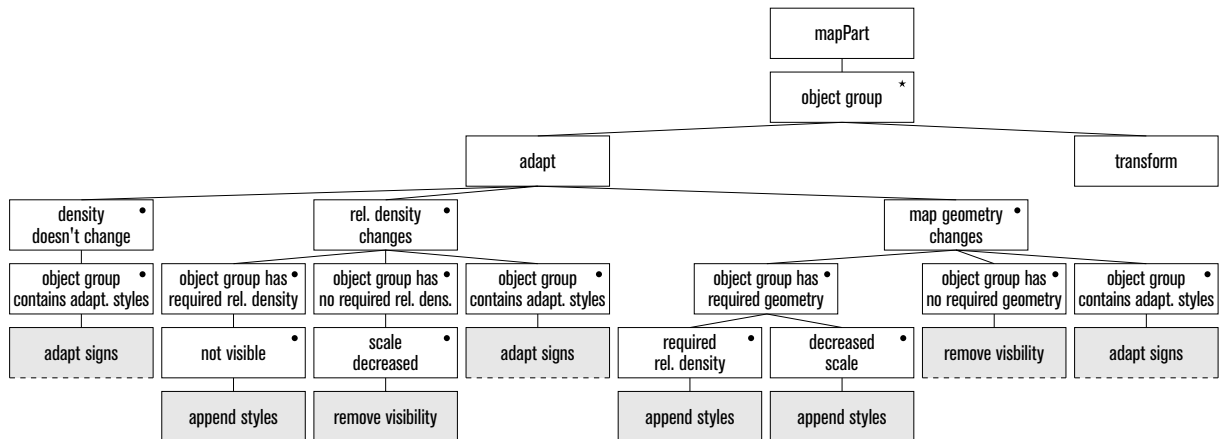


Abb. 42 Die Funktion zoomMap

Funktionen für die Zuweisung von Objektstilen (append stiles) und das Unsichtbarmachen von Objektgruppen (remove visibility) wurden bereits entwickelt. Sie können hier einfach wieder verwendet werden. Neu hinzu kommt die Funktion adaptSigns zur Anpassung der Objektsignaturen, die im nächsten Kapitel erläutert wird.

10.3.1 Anpassung der Objektsignaturen: Die Funktion `adaptSigns`

Mit der Funktion `adaptSigns` werden sowohl die Signaturmasse als auch die Labelplatzierung einzelner Objektgruppen angepasst. Sie wird direkt aus der Funktion `zoomMap` aufgerufen und daher für jede Objektgruppe angewendet, die adaptive Signaturmasse enthält.

(Abb. 43) Für alle betroffenen Objektgruppen gilt also:
 Extrahiere zuerst aus allen Stilen einer Objektgruppe das adaptive Signaturmass.
 Berechne dann mit der Funktion (11) auf Seite 54 das neue Signaturmass.
 Füge dann das neue Signaturmass der Objektgruppe an.

Falls die Objektgruppe Schrift enthält (bzw. falls das adaptive Signaturmass eine Schriftdefinition enthält):

- ♦ Falls die Objektgruppe aus Labels besteht, und falls sich das Label im Quadrant A befindet (vgl. Kapitel Tab. 6 auf Seite 58):
 - Ermittle zuerst den Signaturentyp;
 - berechne dann die Verschiebung entlang der x-Achse um die negative Signaturgrössendifferenz;
 - berechne dann die Verschiebung entlang der y-Achse um die Signaturgrössendifferenz;
 - verschiebe dann die Objektgruppe.
- ♦ Falls die Objektgruppe aus Labels besteht, und falls sich das Label im Quadrant B befindet:
 - Ermittle zuerst den Signaturentyp;
 - berechne dann die Verschiebung entlang der x-Achse um die negative Signaturgrössendifferenz;
 - berechne dann die Verschiebung entlang der y-Achse um die negative Summe von Signaturgrössendifferenz und Versalhöhendifferenz;
 - verschiebe dann die Objektgruppe.
- ♦ Falls die Objektgruppe aus Labels besteht, und falls sich das Label im Quadrant C befindet:
 - Ermittle zuerst den Signaturentyp;
 - berechne dann die Verschiebung entlang der x-Achse um die Signaturgrössendifferenz;
 - berechne dann die Verschiebung entlang der y-Achse um die negative Summe von Signaturgrössendifferenz und Versalhöhendifferenz;
 - verschiebe dann die Objektgruppe;
 - verschiebe dann jedes einzelne Objekt um die Labelbreitendifferenz.
- ♦ Falls die Objektgruppe aus Labels besteht, und falls sich das Label im Quadrant D befindet:
 - Ermittle zuerst den Signaturentyp;
 - berechne dann die Verschiebung entlang der x-Achse um die Signaturgrössendifferenz;
 - berechne dann die Verschiebung entlang der y-Achse um die Signaturgrössendifferenz;
 - verschiebe dann die Objektgruppe;
 - verschiebe dann jedes einzelne Objekt um die Labelbreitendifferenz.
- ♦ Falls die Objektgruppe aus Punktsignaturen besteht:
 - Ermittle zuerst den Signaturentyp;
 - berechne dann die Verschiebung entlang der x-Achse um die Signaturgrössendifferenz;
 - verschiebe dann die Objektgruppe.

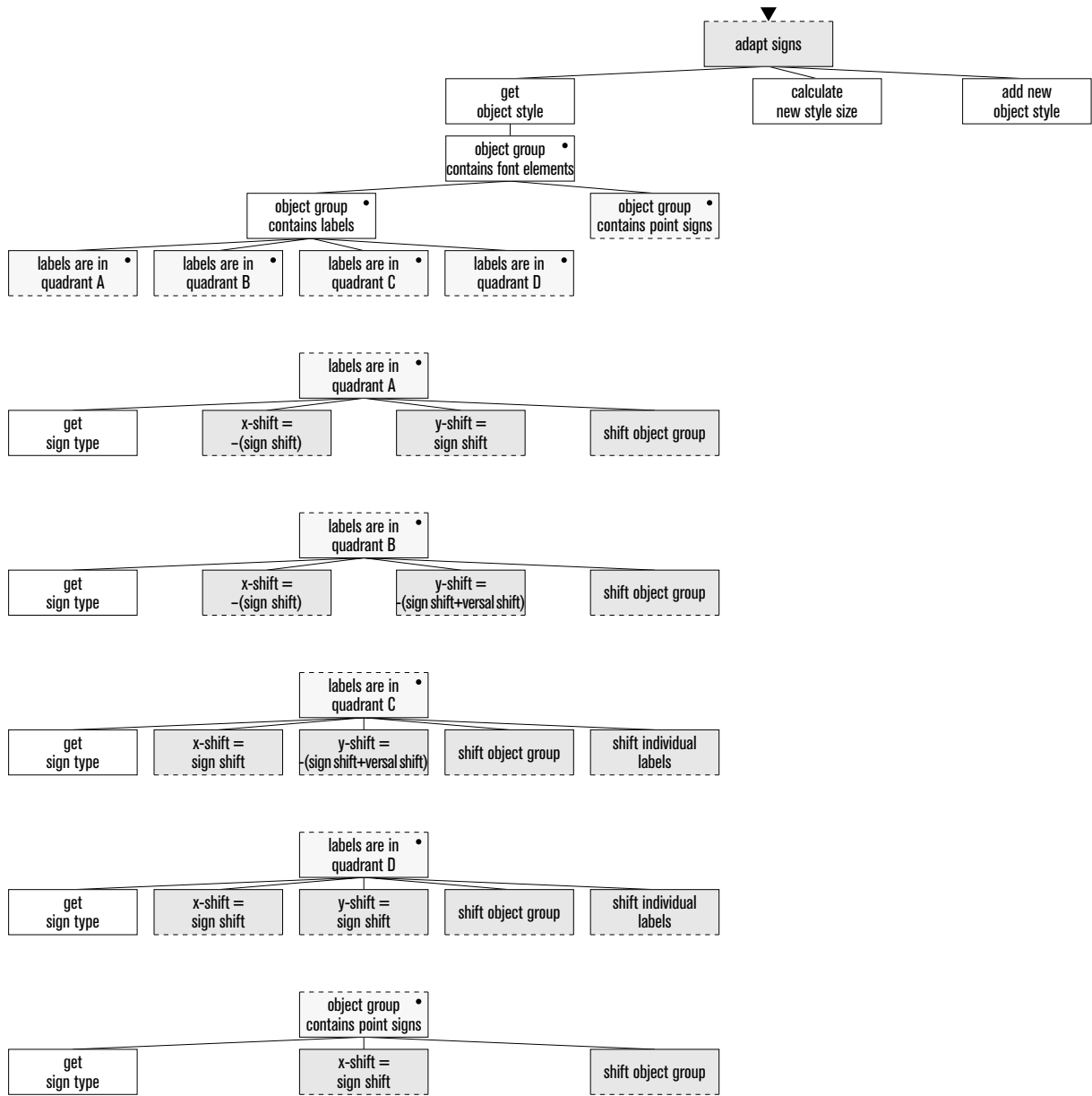


Abb. 43 Die Funktion adaptSigns

Die Funktion adaptSigns benötigt für die Berechnung der einzelnen Verschiebungswerte ihrerseits die Funktionen signShift, versalShift, shiftGroup und shiftIndividuals.

- ⊙ Ortschaft mit 5000 bis 10 000 Einwohnern
- ⊙ Ortschaft mit 2000 bis 5000 Einwohnern
- Ortschaft mit weniger als 2000 Einwohnern

Abb. 44 Die im Projekt vorkommenden Punktsignaturen

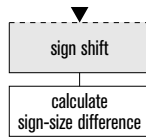


Abb. 45 Die Funktion signShift

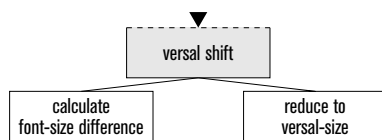


Abb. 46 Die Funktion versalShift

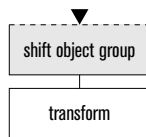


Abb. 47 Die Funktion shiftGroup

10.3.2 Berechnung der Punktsignaturendifferenz: Die Funktion signShift

Die Funktion signShift berechnet, um wieviel ein Label oder eine Punktsignature aufgrund der Grössenänderung der Punktsignature verschoben werden muss (vgl. Kapitel 7.7 ‹Anpassung der Punktsignaturen› auf Seite 54).

In diesem Projekt kommen nur drei Arten bzw. Grössen von Punktsignaturen vor (siehe Abb. 44). Für jede der Punktsignaturen sind die Ursprungsgrössen bekannt. Der Verschiebungswert wird nun mit Hilfe der Funktion (12) auf Seite 55 berechnet.

(Abb. 45) Für alle betroffenen Objektgruppen gilt: Berechne den durch die Signaturgrössendifferenz verursachten Verschiebungswert der Objektgruppe.

10.3.3 Berechnung der Versalhöhendifferenz: Die Funktion versalShift

Die Funktion versalShift berechnet, um wieviel ein Label aufgrund der Grössenänderung der Schrift entlang der y-Achse verschoben werden muss (vgl. Kapitel 7.8.3 ‹Adaptive Platzierung der Kartenbeschriftung› auf Seite 57). Die für diese Rechnung benötigte Versalhöhe entspricht etwa 85% der Schriftgrösse und kann daher aus der Schriftgrösse abgeleitet werden.

(Abb. 46) Für alle betroffenen Objektgruppen gilt: Berechne den durch die Versalhöhendifferenz verursachten Verschiebungswert der Objektgruppe.

10.3.4 Verschiebung der Objektgruppen: Die Funktion shiftGroup

Mit der einfachen Transformfunktion shiftGroup wird die Objektgruppe jetzt anhand der mit signShift und versalShift ausgerechneten Werte verschoben.

(Abb. 47) Für alle betroffenen Objektgruppen gilt: Verschiebe die Objektgruppe um den berechneten Verschiebungswert.

10.3.5 Berücksichtigung der Labelgrössen: Die Funktion shiftIndividuals

Die Verschiebung von Labels entlang der x-Achse um den Wert der Labelbreitendifferenz ist ein etwas komplexerer Vorgang. Da die Labels je nach Anzahl und Art der verwendeten Buchstaben sehr unterschiedliche Breiten haben, müssen alle Labels einzeln verschoben werden. Dies geschieht in der Funktion shiftIndividuals.

(Abb. 48) Für jedes einzelne Label (nicht für die ganze Objektgruppe) gilt: Falls der *node*, der das Label enthält, ein *element node* ist (*white space handling* muss aus den früher genannten Gründen auch bei dieser Funktion unbedingt berücksichtigt werden, vgl. Kapitel 10.2.1 ‹White Space Handling› auf Seite 74), dann gilt für jeden Buchstaben des Labels:

- ✦ Finde zuerst die Buchstabengrösse heraus;
- ✦ addiere sie dann zur Labelbreite, die bisher ermittelt wurde.

Danach rechne die neue Labelbreite aus.

Danach rechne den Verschiebungsbetrag, also die Labelbreitendifferenz aus.

Falls bisher kein *comment node* vorhanden ist, der die ursprüngliche Position des Labels festhält: Ermittle die ursprüngliche Position aus dem *transform attribute* und produziere dann einen neuen *comment node*, der die Position speichert.

Andernfalls (falls die ursprüngliche Position bereits in einem *comment node* festgehalten wurde): Ermittle die ursprüngliche Position des Labels aus dem *comment node*.

Addiere dann die Labelbreitendifferenz zur Position auf der x-Achse.

Weise dann dem Label den neuen Transformwert zu.

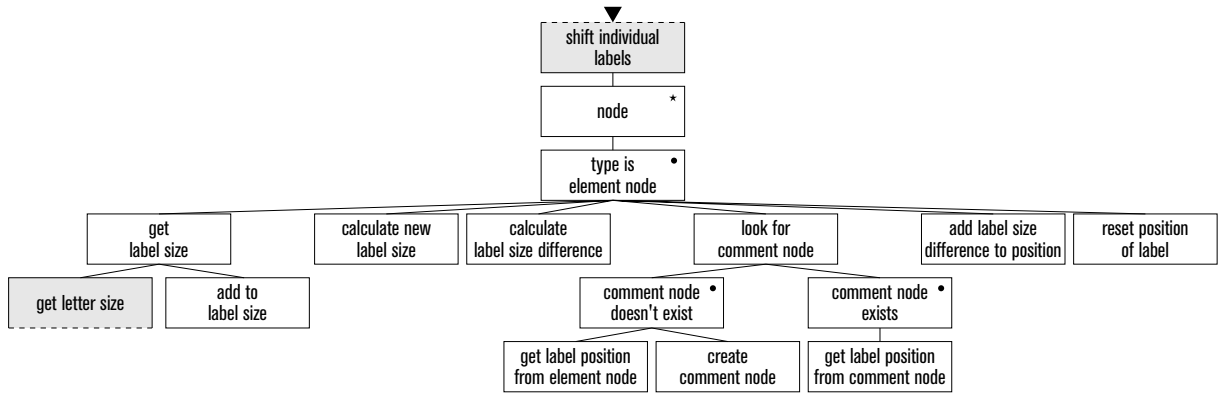


Abb. 48 Die Funktion *shiftIndividuals*

10.3.6 Ermitteln der Buchstabengröße: Die Funktion *letterSize*

Zuletzt fehlt noch eine Funktion, mit der die Grösse einzelner Buchstaben ermittelt werden kann. SVG kennt leider keine Standardfunktion, die für diese Aufgabe geeignet ist. Es bleibt also nichts anderes übrig, als die Grösse sämtlicher Buchstaben in einer Liste zusammenzustellen. Je nachdem, welche individuelle Buchstabengrösse gerade benötigt wird, wird der entsprechende Wert zurückgegeben. Die Funktion:

Falls der Buchstabe ein `<A>` ist, gib die Grösse zurück.

muss also pro Buchstabe einmal vorkommen. Sie gilt aber nur für jeweils eine Schrift. Im vorliegenden Projekt wird die Schrift *Berthold Formata Light Condensed* verwendet, die angegebenen Buchstabengrößen beziehen sich also alle darauf.

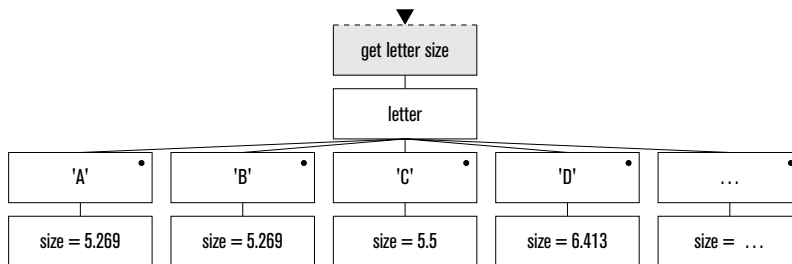


Abb. 49 Die Funktion *letterSize*

Problematisch ist, dass Umlaute und Sonderzeichen nicht erkannt werden. Bei Kartenbeschriftungen kommen Sonderzeichen glücklicherweise praktisch nie vor. Um aber für die Umlaute einen Verschiebungswert zu bekommen, wird die Funktion so modifiziert, dass sämtliche Zeichen, die nicht erkannt werden, (also bei Karten nur die Umlaute) eine einheitliche Grösse zugewiesen bekommen, die dem Durchschnittswert der drei Umlaute entspricht.

Schon zu Beginn der Implementationsphase der adaptiven Karte wurde wiederholt festgestellt, dass vor allem die Festlegung einiger Grenzwerte nur empirisch erfolgen kann. Betroffen sind insbesondere

- die Art des Wachstums von Signaturmassen, d. h. die Festlegung des maximalen Skalierungsfaktors und der Wachstumsgeschwindigkeit von allen Signaturmassen (vgl. Kapitel 7.6 ‹Anpassung der Signaturmasse: Liniensignaturen› ab Seite 51),
- der Wechsel von einem Geometriesatz zum nächsten (vgl. Kapitel 9.2.2 ‹Festlegung der Skalierungsgrenzwerte› auf Seite 65), und damit auch die Festlegung der Informationsgehalt-Grenzwerte (vgl. Kapitel 7.5 ‹Informationsdichte und Informationsgehalt› auf Seite 49).

In diesem Kapitel wird die adaptive Karte aufgrund von diversen Tests und optischen Eindrücken optimiert. Die Art des Wachstums von Signaturmassen wird zuerst untersucht, da sie unabhängig von den Informationsgehalt-Grenzwerten erfolgt.

11.1 Festlegung der Art des Wachstums von Signaturmassen

Die Art des Wachstums von Signaturmassen ist in Funktion (11) auf Seite 54 beschrieben:

$$s_1 = s_0 \left(\frac{2(r_0 - 1)}{\pi} \operatorname{atan} \left(t \left(\frac{m_0}{m_1} - 1 \right) \right) + 1 \right)$$

wobei

s_0 das ursprüngliche Signaturmass,

s_1 das neue Signaturmass,

r_0 der gewünschte maximale Skalierungsfaktor,

m_0 der ursprüngliche Massstab,

m_1 der neue Massstab

und t die gewünschte Wachstumsgeschwindigkeit ist.

Die empirisch anzupassenden Werte sind der gewünschte maximale Skalierungsfaktor r_0 und die gewünschte Annäherungsgeschwindigkeit an die Asymptote, also die Wachstumsgeschwindigkeit t .

Eine Versuchsreihe ergibt, dass die subjektiv besten optischen Eindrücke mit $r_0=1,2$ und $t=0,6$ entstehen (siehe Abb. 50).

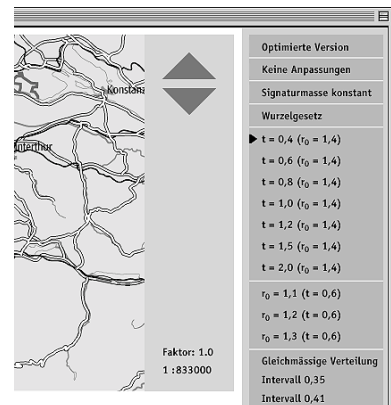


Abb. 50 Festlegung von r_0 und t

11.2 Festlegung der Informationsgehalt-Grenzwerte

Es ist offensichtlich, dass die in Abb. 33 auf Seite 68 festgelegten Informationsgehalt-Grenzwerte nicht geeignet sind. Der Geometriesatz s_{10} wird zwar befriedigend dargestellt, die Informationsdichte stimmt zumindest in den kleineren Massstäben ziemlich gut. Schon der Wechsel zu s_{05} scheint aber eher spät zu erfolgen, die Informationsdichte der letzten Stufe von s_{10} ist zu gering. Auch nach dem Wechsel verhält sich die Informationsdichte nicht wie erhofft. Das Kartenbild erscheint zusehends ausgedünnt.

Als erste Verbesserungsmaßnahme werden, trotz der Tatsache, dass der Grenzwert zum Wechsel der Geometriesätze noch nicht optimal gewählt ist, die Informationsgehalt-Grenzwerte so gewählt, dass sie in einem regelmässigen Abstand wachsen. Als Intervall wird 0,41 gewählt, also derjenige Wert,

s10	s05		
0		1:833 333	1.0
1			1.38
2			1.76
3			2.14
	4	1:330 688	2.52
	5		2.9
	6		3.28
	7		3.66
	8		4.04
		1:88 537	4.42

Abb. 51 Überarbeitete Informationsgehalt-Grenzwerte

welcher bei s10 zu einer befriedigenden Darstellung geführt hat. Es zeigt sich bereits ein klar verbessertes Resultat, das aber immer noch eine geringe Tendenz zur Ausdünnung aufweist.

Um die Ausdünnung noch weiter einem Minimum anzunähern, werden in einem nächsten Schritt mit verschiedenen Intervallgrößen Versuche durchgeführt. Ein kleineres Intervall bedeutet, dass einerseits die Ausdünnung geringer wird, und andererseits der Wechsel des Geometriesatzes etwas früher erfolgt. Das Intervall darf dabei aber nicht zu klein sein, denn dies hätte ein dichter werdendes Kartenbild zur Folge.

Mehrere Versuche ergeben, dass das Kartenbild bei einer Intervallgröße von 0,38 am besten ist. Die genauen Werte können der Abb. 51 entnommen werden.

Werden die Informationsgehalt-Grenzwerte in gleichmässigen Abständen, d. h. mit gleich bleibendem Intervall gewählt, dann ist die Qualität der kartografischen Darstellung sehr viel besser, als wenn die Abstände ungleichmässig gewählt würden. Das Kartenbild erscheint ruhiger, und die Karten der unterschiedlichen Massstäbe passen besser zueinander.

Auf den ersten Blick nachteilig scheint, dass nicht mehr der ganze Zoombereich ausgenutzt werden kann. Sobald der Zoomfaktor den Wert des letzten Informationsgehalt-Grenzwertes überschreitet, wird der Informationsgehalt nicht mehr erhöht, obwohl der Wert der maximalen Vergrößerung noch nicht erreicht ist. Bezieht man sich aber auf die Aussage in Kapitel 9.2.2 «Festlegung der Skalierungsgrenzwerte» auf Seite 65, dass Extremwerte vermieden werden sollten, so kann dieser Umstand sogar eher als Vorteil betrachtet werden.

Die vorliegende Arbeit hatte zum Ziel, eine Bildschirmkarte zu entwickeln, die von einer Benutzerin ohne Expertenwissen interaktiv im Massstab verändert werden kann; mit der Änderung des Massstabs sollte sich auch die Detailgenauigkeit der Karte anpassen (vgl. Kapitel 1 ‹Zielsetzung und Überblick› ab Seite 1).

Gleichzeitig war es die Absicht, die Entwicklung weitgehend unabhängig von der Form der Implementation zu halten; die Methoden und Konzepte, die hier entwickelt wurden, sollen als Grundlage für andere, ähnliche Projekte dienen.

Auf den nächsten Seiten werden die Resultate der Arbeit zusammengetragen. Es wird auch untersucht, inwiefern und mit welchen Einschränkungen die für die adaptive Karte gesteckten Ziele erreicht werden konnten. Überdies wird eine mögliche Weiterführung des Projektes skizziert, und die zukünftig denkbaren Erweiterungen werden ausgelotet. Anschliessend wird der persönliche Nutzen der Arbeit kurz festgehalten.

Bevor die Resultate der Arbeit analysiert werden, ist anzumerken, dass die aus dem Projekt entstandene interaktive Karte dem eingangs formulierten Hauptziel entspricht. Die Karte kann auf Knopfdruck im Massstab verändert werden und passt sich dabei grafisch und im Detaillierungsgrad an; die Bedienung ist äusserst einfach, von der Benutzerin werden nicht mehr Kenntnisse verlangt, als dies konventionelles Surfen im Internet auch tut.

Die Kernaussagen dieser Arbeit – also die für eine adaptive Karte notwendigen Methoden und Konzepte – sind ausführlich in Kapitel 7 ‹Interaktive Massstabsveränderungen durch adaptives Zoomen› ab Seite 43 beschrieben. Die implementationsbezogenen Problemlösungen sind, auf den Resultaten aus Kapitel 7 aufbauend, in Kapitel 9 ‹Aufbau des adaptiven Datensatzes› ab Seite 63 und Kapitel 10 ‹Das Steuerprogramm für die adaptive Karte› ab Seite 73 zu finden.

12.1 Allgemeine Resultate

Im ersten Teil der Arbeit werden die Rahmenbedingungen der Bildschirmausgabe von Karten abgesteckt. Die Analyse beschränkt sich aber auf diejenigen Themen, die später für adaptives Zoomen nützlich sind. Die technischen Grundlagen, die Vor- und Nachteile der Bildschirmausgabe, die Regeln der Kartengestaltung und die Interaktion mit Bildschirmkarten werden untersucht.

Ab Kapitel 2 ‹Der Bildschirm als Ausgabemedium für Grafik› auf Seite 5

12.1.1 Eignung von Bildschirmen zur Ausgabe von Karten

Bildschirme haben offenkundig im Vergleich mit Papier grosse Nachteile als Ausgabemedium für Karten. Insbesondere sind die schlechtere Auflösung und die dem Kartenautoren nicht bekannte Qualität des von der Benutzerin verwendeten Computersystems zu erwähnen. Die reine kartografische Darstellung von Inhalten ist auf Papier also besser und übersichtlicher.

Kapitel 3.2 ‹Unterschiede zwischen Papierkarten und Bildschirmkarten› ab Seite 11

Die auf Bildschirmen möglichen Mittel der Interaktion bieten aber dem Kartenautoren ein ganz neues Potenzial der Informationsvermittlung. Die Vermittlung komplexer räumlicher Daten ist mit interaktiven Anwendungen möglich, ohne dass von der Benutzerin Expertenwissen verlangt werden muss. Der Umgang mit räumlichen Daten kann sogar spielerisch geschehen.

und Kapitel 5 ‹Interaktion mit Bildschirmkarten› ab Seite 27

Kapitel 3.3 «Eine geeignete Entwicklungsumgebung für Bildschirmkarten» ab Seite 13

und Kapitel 6.3 «SVG für Bildschirmkarten» ab Seite 40

Kapitel 4 «Gestaltung von Bildschirmkarten» ab Seite 17

und Kapitel 2.1.5 «Antialiasing» ab Seite 7

Ab Kapitel 7 «Interaktive Massstabsveränderungen durch adaptives Zoomen» auf Seite 43

Kapitel 7.2 «Konzept und Definitionen» ab Seite 45

12.1.2 SVG als Entwicklungsumgebung für Bildschirmkarten

Eine Aufstellung aller Anforderungen der Bildschirmkartografie an eine Entwicklungsumgebung ergibt, dass die neue, XML-basierte Technologie SVG für interaktive Karten wohl am besten geeignet ist: Nicht nur durch den grossen Funktionsumfang zeichnet sich SVG aus, sondern auch durch Unterstützung von Softwareherstellern auf breiter Front.

Die Möglichkeiten, die SVG der Kartografie zu bieten hat, sind riesig. Dynamische Kartenproduktion, mediumsorientierte Ausgabe, unterschiedliche Sprachversionen, dynamische Legendenfunktion und natürlich adaptive Karten sind einige Stichworte.

12.1.3 Verbesserung der Darstellungsqualität von Bildschirmkarten

Die Grundsätze zur Kartengestaltung, wie sie für die Papierausgabe gelten, haben mit Einschränkungen auch für Bildschirmkarten ihre Gültigkeit. Im Hinblick auf adaptives Zoomen müssen aber insbesondere die grafischen Mindestgrössen und die Kriterien für die Schriftwahl modifiziert werden.

Die wohl wichtigste, eher technisch orientierte Möglichkeit, die Qualität der Kartendarstellung auf Bildschirmen zu verbessern, ist *Antialiasing*, das aber auch einige Probleme birgt; insbesondere müssen Flächenzwischenräume etwas grösser gewählt werden.

12.2 Projektspezifische Resultate

Was grafische GIS-Produkte von interaktiven Karten unterscheidet, wie genau adaptives Zoomen definiert werden kann, welche Probleme für adaptives Zoomen zu lösen sind, und wie eine adaptive Karte entwickelt wird: Das sind diejenigen Fragen, die im zweiten Teil der Arbeit beantwortet werden.

Aufgrund ihrer höheren Bedeutung werden die projektspezifischen Resultate auf den nächsten Seiten etwas ausführlicher geschildert, als die allgemeinen Resultate im letzten Kapitel.

12.2.1 Das Konzept «adaptives Zoomen»

«*Adaptives Zoomen*» kann wie folgt definiert werden: «Ein Zoomprozess ist dann adaptiv, wenn sich die Darstellung einer Karte so an den Kartenmassstab respektive den Zoomfaktor anpasst, dass die Kartenqualität immer als gut empfunden wird. Dies gilt insbesondere für den Informationsgehalt, die Informationsdichte und die grafische Darstellung aller Signaturen.»

Eine adaptive Karte besteht immer aus der Kartengeometrie, die im adaptiven Datensatz enthalten ist, und einem Programm, das die Steuerung der notwendigen Prozesse übernimmt.

Der adaptive Datensatz setzt sich aus mehreren, sich massstäblich unterscheidenden Geometriesätzen zusammen, die jeweils in bestimmten Grenzen skalierbar sind. Jeder dieser Geometriesätze deckt einen bestimmten Massstabsbereich ab; die Massstabsbereiche zweier aneinander anschliessender Geometriesätze überlappen sich im Optimalfall, können aber auch direkt aneinander anschliessen. Jeder Geometriesatz kann pro Objektgruppe verschiedene Dichtestufen haben.

Beim Zoomprozess wird, ausgehend von der äussersten Zoomstufe, ein Geometriesatz so lange kontinuierlich vergrössert, bis sein

Vergrößerungslimit erreicht ist. Gleichzeitig mit der Vergrößerung des Massstabs müssen neue Dichtestufen einzelner Objektgruppen hinzukommen, und zwar so, dass die Informationsdichte konstant bleibt. Wenn das Vergrößerungslimit erreicht ist, wird der Geometriesatz aus- und ein neuer, auf den anschliessenden Zoombereich optimierter Geometriesatz eingeblendet.

Um die Qualität der kartografischen Darstellung beibehalten zu können, müssen Signaturmasse bei der Vergrößerung des Massstabs in der Dimension relativ zu ihrer Umgebung kleiner gewählt werden können als im Vorläufermassstab. Die Signaturmasse müssen jeweils so bemessen sein, dass alle kartografischen Regeln eingehalten werden können.

12.2.2 Der adaptive Datensatz

Der adaptive Datensatz bildet die Grundlage für die Funktionalität der adaptiven Karte. Er muss sowohl kartografisch als auch technisch hohen Anforderungen genügen, da alle folgenden Arbeiten darauf aufbauen.

Die verschiedenen Geometriesätze müssen gut aufeinander abgestimmt sein. Diese Bedingung ist besonders schwierig zu erfüllen, da sich die Geometriesätze, weil sie für verschiedene Massstäbe gelten, in der Generalisierung unterscheiden.

Die Entscheidung, welcher Geometriesatz welche Skalierungsgrenzwerte hat, kann nicht ohne weiteres und vor allem nicht mit mathematisch erfassbaren Kriterien getroffen werden, da die Qualität der Generalisierung und der formale Ausdruck der Darstellungskomplexität nicht möglich sind. Die Skalierungsgrenzwerte liegen zwar ungefähr in den Bereichen von 50% bzw. 200% des Originalmassstabs, können aber je nach Karte recht unterschiedlich sein.

*Kapitel 7.4 «Der adaptive Datensatz»
ab Seite 48*

12.2.3 Informationsdichte und Informationsgehalt

Mit grösser werdendem Massstab vergrössert sich auch die zur Verfügung stehende Fläche, wenn die Signaturmasse angepasst werden. Diese Fläche kann mit neuen und zusätzlichen Informationen gefüllt werden. Der Informationsgehalt muss dabei in einem solchen Ausmass vergrössert werden, dass die Informationsdichte konstant bleibt.

Neue Dichtestufen kommen dann hinzu, wenn der Informationsgehalt-Grenzwert überschritten wird. Der Informationsgehalt-Grenzwert hängt vom Darstellungsgewicht einzelner Objekte und Objektgruppen ab, wobei kontrastreiche und massive Liniensignaturen eher schwer und kleine, feine Punktsignaturen leicht sind.

Die Informationsgehalt-Grenzwerte sind mathematisch nicht erfassbar und müssen mittels Versuchen festgelegt werden.

Kapitel 7.5 «Informationsdichte und Informationsgehalt» ab Seite 49

Kapitel 7.6 «Anpassung der Signaturmasse: Liniensignaturen» ab Seite 51

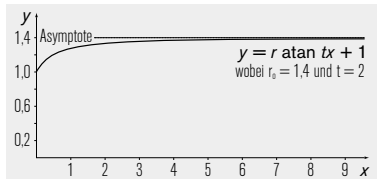


Abb. 52 Der Graph der Funktion für die Anpassung der Signaturmasse

Kapitel 7.7 «Anpassung der Punktsignaturen» ab Seite 54

Kapitel 7.8 «Beschriftung von Punktsignaturen» ab Seite 55

Ab Kapitel 9 «Aufbau des adaptiven Datensatzes» auf Seite 63

Kapitel 10 «Das Steuerprogramm für die adaptive Karte» ab Seite 73

12.2.4 Anpassung der Signaturmasse

Die Signaturmasse (sowohl Linien-, als auch Punktsignaturen und Schriftgrößen) können mit der folgenden Funktion an den Masstab angepasst werden:

$$s_1 = s_0 \left(\frac{2(r_0 - 1)}{\pi} \operatorname{atan} \left(t \left(\frac{m_0}{m_1} - 1 \right) \right) + 1 \right)$$

wobei

s_0 das ursprüngliche Signaturmass,

s_1 das neue Signaturmass,

r_0 der gewünschte maximale Skalierungsfaktor,

m_0 der ursprüngliche Masstab,

m_1 der neue Masstab

und t die gewünschte Wachstumsgeschwindigkeit ist.

Diese Funktion berücksichtigt sowohl den maximalen Skalierungsfaktor der Signaturmasse, als auch die Annäherungsgeschwindigkeit an diesen Maximalwert.

12.2.5 Anpassung von Punktsignaturen

Punktsignaturen werden als Texte behandelt, die immer linksbündig gesetzt sind. Deshalb werden sie beim Skalieren relativ zu ihrem Mittelpunkt etwas verschoben. Diese Verschiebung muss mit der Funktion (12) auf Seite 55 korrigiert werden.

12.2.6 Anpassung der Beschriftung von Punktsignaturen

Durch die Veränderung der Schriftgröße verschieben sich Labels von Punktsignaturen relativ zu ihrer ursprünglichen Position. Je nachdem, ob ein Label sich links, rechts, oberhalb oder unterhalb der zugehörigen Punktsignatur befindet, muss es anders behandelt werden.

Ein Label wird entsprechend seiner Position so in die richtige Richtung geschoben, dass bei der Masstabsänderung weder der horizontale, noch der vertikale Abstand zwischen dem Label und dem Rand der Punktsignatur verändert wird. Natürlich muss dabei auch die Größenänderung der Punktsignatur berücksichtigt werden.

12.3 Analyse der Implementationsphase

Zur eigentlichen Implementationsphase des Projektes gehören der Aufbau des adaptiven Datensatzes und das Steuerprogramm für die adaptive Karte. Beide Teile sind stark praktisch orientiert.

Es wird einerseits beschrieben, auf welche Systematiken beim Aufbau des adaptiven Datensatzes im Grafikprogramm Wert zu legen ist. Dies schliesst die Anpassung der Signaturmasse und der Labelpositionen in Abhängigkeit von den Skalierungsgrenzwerten einzelner Geometriesätze mit ein, aber auch die Bezeichnung der Ebenen, die später als Identifikation einzelner Objektgruppen verwendet werden. Ausserdem ist der Aufbau der SVG-Anwendung ein wichtiges Thema.

Andererseits wird die logische Abfolge des Steuerprogramms ausführlich geschildert. Das Steuerprogramm muss grundsätzlich zwei Abläufe steuern: hauptsächlich den Zoomprozess, daneben aber auch die Prozesse, die ablaufen, wenn die Karte geladen wird.

Im Lauf der Entwicklung der adaptiven Karte stellte sich immer mehr heraus, welche Konzepte und Methoden wie gut umsetzbar sind. Das Kapitel 7.3 «Projektspezifische Ziele» ab Seite 46 enthält eine Auflistung aller Ziele, die in diesem Projekt erreicht werden sollten. Im Abgleich dazu und als Ergänzung wird auf den nächsten Seiten untersucht, welche Konzepte und Methoden implementiert wurden, und welche nicht oder nur schlecht umgesetzt werden konnten.

12.3.1 Kartografische Darstellung

Fast alle Ziele, die sich in der kartografischen Darstellung äussern, konnten erreicht werden. Das Steuerprogramm für die adaptive Karte übernimmt sowohl die Auswahl von Objektgruppen und Geometriesätzen, als auch die Anpassung von Signaturmassen an den Massstab – und beides in der angestrebten Qualität. Auch die dynamische Anpassung der Schriftplatzierung funktioniert vollständig.

Kapitel 7.3.1 «Kartografische Ziele» ab Seite 46

Die Vergrösserung der Kartengeometrie wurde nur teilweise implementiert. Der technische Vergrösserungsfaktor (vgl. Kapitel 4.5 «Bildschirmspezifische Vergrösserung der Kartengeometrie» ab Seite 24) konnte vorläufig nicht berücksichtigt werden.

12.3.2 Benutzerinteraktion

Die Massstabsveränderung geschieht über ein einfach gestaltetes Werkzeug, das der Benutzerin wenig Möglichkeiten lässt. Ausser dem schrittweisen Ein- und Auszoomen ist nichts möglich.

Kapitel 7.3.2 «Funktionelle Ziele» ab Seite 46

Dieser Funktionsumfang muss als mager bezeichnet werden. Erfüllt ist zwar die Anforderung, dass die Bedienung äusserst einfach ist. Auch das Zoomzentrum kann selber festgelegt werden. Aber die vom *Adobe SVG-Viewer* bereitgestellten Funktionen wie die Lupe oder die Kontextmenü-Befehle *zoom in* und *zoom out* können nicht verwendet werden. Diese Einschränkung muss gemacht werden, da *Adobe* die freie Programmierung beider Funktionen zur Zeit noch nicht zulässt (vgl. Kapitel 8.2.2 «Einschränkungen des Adobe SVG-Viewer» auf Seite 61).

Das Steuerprogramm für die adaptive Karte ist so modular ausgelegt, dass eine Implementation weiterer Navigationsmöglichkeiten mit verhältnismässig kleinem Aufwand und ohne strukturelle Veränderungen geschehen kann, sobald der *SVG-Viewer* diese Möglichkeiten unterstützt.

12.3.3 Karteninhalt

Die adaptive Karte enthält

- die Flächenobjekte Siedlungsflächen und Seen (in den vier Dichtestufen «locker», «mittel», «dicht», «sehr dicht»)
- die linearen Objekte Flussnetz (in den vier Dichtestufen «locker», «mittel», «dicht», «sehr dicht»), Strassennetz (Autobahnen, Hauptstrassen, Nebenstrassen), Eisenbahnnetz (Hauptlinien, Nebenlinien) und Landesgrenzen
- die Punktsignaturen Ortschaften (kleine, mittelgrosse, grosse Ortschaften, jeweils mit eigener Signatur)
- die Beschriftungen für Ortschaften

Kapitel 7.3.3 «Inhaltliche Ziele» ab Seite 47

Von den in Tab. 5 auf Seite 47 aufgeführten Objektgruppen ist nur das Relief nicht im Projekt enthalten. Die Einbindung des Reliefs wurde im Rahmen dieser Arbeit nicht näher untersucht, da bereits die Ladezeit der SVG-An-

wendung über Internet sehr hoch ist. Wäre zusätzlich ein Bitmapbild enthalten, würde die Ladezeit vervielfacht.

12.3.4 Implementation

Auf Implementationsstufe wurden fast alle Ziele erreicht. Die Anwendung ist plattformunabhängig, leicht zu warten, wurde vollständig mit kommerziell erhältlicher Software entwickelt, und die Karte konnte in einer handelsüblichen Grafikapplikation erstellt werden.

Nicht erreicht wurde das Ziel kurzer Ladezeit. Die SVG-Anwendung ist aufgrund der Komplexität der Karte einfach zu gross, um schnell via Internet übertragen zu werden. Da eine Aufteilung in kleinere Teile zur Zeit nicht möglich ist (vgl. Kapitel 8.2.2 ‹Einschränkungen des Adobe SVG-Viewer› auf Seite 61), lässt sich diese Einschränkung leider nicht umgehen. Der kommerzielle Einsatz einer adaptiven Karte von dieser Komplexität ist damit leider etwas in Frage gestellt, bis *Adobe* oder ein anderer Hersteller diese Funktion implementiert hat.

12.4 Weiterführung des Projektes

Wenn auch der Umfang der implementierten Methoden und Konzepte im vorliegenden Projekt schon recht hoch ist, so gibt dennoch einiges, das noch fehlt. Mit dazu gehören

- die Implementation weiterer Information wie z. B. Wald
- die Beschriftung weiterer Objekte
- die Schriftfreistellung
- das Verhalten von Farbe
- die Einbindung des Reliefs
- eine auf interaktive Karten ausgelegte Legende

Die Implementation weiterer räumlicher Informationen ist mit grosser Arbeit verbunden, hat aber auf Entwicklungsstufe wenig Neues zu bieten. Für die Entwicklung fast genau so wenig interessant – wenn auch grafisch äusserst wertvoll – ist die Einbindung eines Reliefs; neu hinzu kommt dabei immerhin die Transparenz von Objekten, die aber in SVG problemlos möglich ist.

12.4.1 Beschriftung weiterer Objekte

Ein interessanter Forschungsbereich könnte die Beschriftung weiterer Objekte sein. Labels für Flächen, die horizontal und auf einer geraden Grundlinie liegen, können einfach und auf die in Kapitel 7.8 ‹Beschriftung von Punktsignaturen› ab Seite 55 beschriebene Art gehandhabt werden. Labels hingegen, die entlang einer gekrümmten Grundlinie laufen, etwa Flusslabels oder teilweise auch Flächenlabels, bedürfen einer speziellen Untersuchung.

12.4.2 Schriftfreistellung

Die Schriftfreistellung in einer digital produzierten Karte für die Papierausgabe ist mit riesigem Aufwand verbunden, besonders dann, wenn ein Label über mehrere verschiedenfarbige Flächen läuft. Alle unter dem La-

beliegenden Objektteile, welche die gleiche Farbe wie der Text haben, müssen mit der Flächenfarbe versehen werden.

Sollen sich in adaptiven Karten Labels dynamisch verschieben können, dann muss sich auch die Schriftfreistellung für jede Labelposition neu anpassen, da immer wieder andere Objektteile überdeckt werden.

12.4.3 Verhalten von Farbe

Unterschiedlich grosse Flächen mit der gleichen Flächenfarbe scheinen nicht immer die gleiche Farbe zu haben. Besonders dann, wenn eine Fläche sehr klein ist, entsteht der Eindruck, dass die Farbe etwas dunkler sei als diejenige der grösseren Fläche.

Solche Effekte, aber auch diejenigen, die in Kapitel 2.2 ‹Wahrnehmung von Grafik auf dem Bildschirm› ab Seite 7 beschrieben sind, sind Teil eines Zoomprozesses und sollten auch berücksichtigt werden.

12.4.4 Legenden für adaptive Karten

Eine adaptive Karte ist so ausgelegt, dass sie zu jedem Zeitpunkt möglichst gut lesbar sein soll; Expertenwissen wird von der Benutzerin nicht erwartet. Dies verlangt auch nach einer Kartenlegende, die schnell und einfach verständlich ist. Natürlich kann dafür eine konventionelle – aber aufgrund des beschränkten Platzes nicht sehr übersichtliche – Legende gestaltet werden, die sich z. B. am rechten Kartenrand befindet: Jeweils ein Kästchen, das die Signatur enthält, wird durch einen Text erläutert.

Eine dynamische Legende kommt den Anforderungen weit besser entgegen. In Kapitel 6.3 ‹SVG für Bildschirmkarten› auf Seite 40 wurde deren mögliche Funktionsweise bereits kurz vorgestellt. Ob sie für eine Benutzerin die erwarteten Vorteile bringt, kann ohne ausführliche Tests zwar erwartet, nicht aber vorausgesetzt werden.

12.5 Persönlicher Nutzen der Arbeit

Die vorliegende Arbeit kann als Grundlage zur Entwicklung anderer interaktiver Karten, die auf SVG basieren, dienen. Nicht die technische Seite von SVG und XML wird dabei in den Vordergrund gestellt (dafür gibt es diverse Tutorials, Spezifikationen und Lehrbücher), sondern viel mehr der kartografische Aspekt.

Neben den spezifisch kartografischen Themen auch wichtig war für mich, zwei Ziele zu erreichen: Einerseits wollte ich neue, zukunftsorientierte Technologien kennen lernen, und andererseits wollte ich mir anhand dieser Arbeit Kenntnisse in Projektmanagement aneignen.

12.5.1 Neue Technologien

Der Umgang mit XML und vor allem SVG stand für mich fast von Anfang an im Vordergrund. Ich hatte mit dieser Arbeit die Chance, mit einer Technologie zu arbeiten, die noch so jung ist, dass sie noch nicht einmal fertig spezifiziert ist, aber sehr gute Chancen hat, sich zu etablieren. Es gibt zwar – vor allem zu SVG – noch so gut wie keine gedruckte Literatur, mit Hilfe von verschiedenen Onlinetutorials und Benutzerforen im Internet konnten aber die meisten Fragen beantwortet werden.

Zusammen mit der Anwendung von SVG musste ich auch den Umgang mit der Skriptsprache *JavaScript* lernen, zu der aber glücklicherweise deutlich mehr Literatur existiert.

12.5.2 Projektmanagement

Da es das erste Mal war, dass ich an einem Projekt dieser Dimension arbeitete, bemühte ich mich, von Anfang an strukturiert vorzugehen. Für alle Teilaufgaben, seien sie nun entwicklungsbezogen oder dokumentationsbezogen, versuchte ich Werkzeuge auszuwählen, die sich eignen. Die verwendeten Entwicklungswerkzeuge habe ich bereits in Kapitel 8 «Entwicklungsumgebung und Ausgabeplattform» ab Seite 59 erwähnt, für die Dokumentation griff ich auf eine Reihe von Applikationen zurück, die nachfolgend kurz beschrieben sind.

www.adobe.com

- ◆ *Adobe FrameMaker 6*: Erfassung von Text und Layout sind in FrameMaker besonders praktisch, weil die Software sowohl eine hervorragende Buchfunktion zur Verwaltung grosser Dokumente hat, als auch Eigenschaften wie alle möglichen Querverweise, Randspalten für Marginalien und anderen Inhalt, ausgezeichnete Typografie und anderes bietet. Auch die Möglichkeit, interaktive PDF-Dateien zu erstellen, ist sehr nützlich.

www.seven-r.ch

- ◆ *7r Templates for FrameMaker*: Eine gut ausgebaute und konsistente Sammlung von Stilvorlagen für Absätze und Zeichen, bereits vorbereitete Inhalts-, Tabellen- und Abbildungsverzeichnisse und andere Eigenschaften sorgen für ein konsistentes Erscheinungsbild der Arbeit.

www.niles.com

- ◆ *Niles EndNote*: Die durchgearbeitete Literatur wird in einer speziellen Applikation verwaltet, welche die Verknüpfung von Text mit Literatur unterstützt. Ein selber definierbares Literaturverzeichnis wird automatisch erstellt.

www.inspiration.com

- ◆ *Inspiration 6*: Für Projektplanung und Brainstorming eignet sich diese Applikation; sie verhilft zu besserer Übersicht und erlaubt die Darstellung aller *items* eines Projektes sowohl in Diagramm- als auch in Listenform.

Zur Recherche diente mir in weiten Teilen das Internet. Nicht nur Tutorials und andere informative Homepages waren nützlich, sondern auch die Zusammenarbeit mit anderen Entwicklern, die mit SVG arbeiten.

- Adobe (2000a): *Adobe Illustrator Version 9.0*. Adobe Systems GmbH, Unterschleissheim.
- Adobe (2000b): Adobe SVG Format Plug-In for Illustrator – Release Notes. Website, Zugriff am 5.6.2000. www.adobe.com/svg/indepth/releasenotes.html.
- Adobe (2000c): Adobe SVG Viewer for Macintosh – Release Notes. Website, Zugriff am 4.5.2000. www.adobe.com/svg/indepth/releasenotes.html.
- Adobe (2000d): Current Support for SVG. Adobe SVG Viewer, Adobe SVG Format Plug-In for Illustrator. Website, Zugriff am 23.8.2000. www.adobe.com/svg/viewer/install.
- Adobe (2000e): SVG: Scalable Vector Graphics. Website, Zugriff am 9.8.2000. www.adobe.com/svg/.
- Apparao, V., Byrne, S., Champion, M., Isaacs, S., Jacobs, I., Le Hors, A., Nicol, G., Robie, J., Sutor, R., Wilson, C. und Wood, L. (1998): Document Object Model (DOM) Level 1 Specification Version 1. W3C Recommendation. Website, Zugriff am 10.5.2000. www.w3.org/TR/REC-DOM-Level-1/.
- Arleth, M. (1999): Problems in Screen Map Design. *19th International Cartographic Conference of ICA*, Ottawa.
- Ballhaus, A. (1998): *Bildbearbeitung am PC*. teWi Verlag GmbH, München.
- Bidoshi, K., Ramirez, J. P. und Caelli, T. (1999): Multimedia Visualization for Maps of the Future. *19th International Cartographic Conference of ICA*, Ottawa.
- Bos, B. (1999): XML in 10 Points. Website, Zugriff am 4.5.2000. www.w3.org/XML/XML-in-10-points.
- Bos, B. und Lie, H. W. (1996): Cascading Style Sheets, Level 1. Website, Zugriff am 10.5.2000. www.w3.org/pub/WWW/TR/REC-CSS1.
- Bos, B., Lie, W. H., Lilley, C. und Jacobs, I. (1998): Cascading Style Sheets, Level 2. CSS2 Specification. Website, Zugriff am 10.5.2000. www.w3.org/TR/REC-CSS2.
- Bowler, J., Capsimalis, M., Cohn, R., Dodds, D., Donoho, A., Duce, D., Evans, J., Ferraiolo, J., Furman, S., Graffagnino, P., Graham, R., Hardy, V., Henderson, L., Hester, A., Hopgood, B., Jolif, C., Lawrence, K., Lilley, C., Mansfield, P., McCluskey, K., Nguyen, T., Sandal, T., Santangeli, P., Sheikh, H., State, G., Stevahn, R., Thompson, T. und Zhou, S. (2000): Scalable Vector Graphics (SVG) 1.0 Specification. W3C Candidate Recommendation 2 August 2000. Website, Zugriff am 10.4.2000. www.w3.org/TR/2000/03/WD-SVG-20000303.
- Bradley, N. (2000): *The XML Companion*. Addison-Wesley. An imprint of Pearson Education, Harlow, England.
- Brandenberger, C. (2000): Re: Wurzelgesetz, 23.5.2000, persönliche Mitteilung. eMail: brandenberger@karto.baug.ethz.ch.
- Bray, T., Paoli, J. und Sperberg-McQueen, C. M. (1998): Extensible Markup Language (XML) 1.0. W3C Recommendation. Website, Zugriff am 10.5.2000. www.w3.org/TR/REC-xml.
- Brown, A. (1993): Map Design for Screen Displays. In: *The Cartographic Journal*, Ausgabe 2/1993, S. 129–135.
- Bungartz, H.-J., Griebel, M. und Christoph, Z. (1996): *Einführung in die Computergraphik*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden.
- Cartwright, W., Peterson, M. P. und Gartner, G. (1999): Multimedia Cartography Approaches to the Presentation of Geographical Information. *19th International Cartographic Conference of ICA*, Ottawa.
- Dickmann, F. und Zehner, K. (1999): *Computerkartographie und GIS*. Westermann, Braunschweig.
- Eberl, M. und Jacobsen, J. (2000): *Macromedia Director 7*. Markt&Technik Verlag, München.
- Ferraiolo, J. (2000): Re: zoom & pan, 14.8.2000, persönliche Mitteilung. eMail: jferraio@adobe.com.
- Hake, G. und Grünreich, D. (1994): *Kartographie*. Walter de Gruyter, Berlin.
- Kiehn, A. und Titzmann, I. (1998): *Typographie interaktiv! Ein Leitfaden für gelungenes Screen-Design*. Springer-Verlag, Berlin Heidelberg.
- Kilberth, K. (1989): *JSP – Einführung in die Methode des Jackson Structured Programming*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden.
- Lexico.com (1997): Dictionary.com. Website, Zugriff am 25.4.2000. www.dictionary.com.

- 94 MacEachren, A. M., Edsall, R., Haug, D., Baxter, R., Otto, G., Masters, R., Fuhrmann, S. und Liujian, Q. (1999): Virtual Environments for Geographic Visualization: Potential and Challenges. Website, Zugriff am 7.3.2000. www.geovista.psu.edu.
- Miller, S. (1999): Design of Multimedia Mapping Products. *Multimedia Cartography*. William Cartwright, Michael P. Peterson und Georg Gartner. Berlin, Springer Verlag.
- Musciano, C. und Kennedy, B. (1999): *HTML. Das umfassende Referenzwerk*. O'Reilly Verlag GmbH, Köln.
- Netscape Communications Corporation (1999a): Cascading Style Sheets Developer Central. Website, Zugriff am 11.5.2000. developer.netscape.com/tech/css/css.html.
- Netscape Communications Corporation (1999b): JavaScript Developer Central. Website, Zugriff am 11.5.2000. developer.netscape.com/tech/javascript/javascript.html.
- Netscape Communications Corporation (1999c): XML Developer Central. Website, Zugriff am 11.5.2000. developer.netscape.com/tech/xml/xml.html.
- Neumann, A. (2000): Vienna – Social patterns and structures. Website, Zugriff am 22.10.2000. www.karto.ethz.ch/~an/cartography/vienna.
- Popa, A. (1997): Image Resolution Limits. Website, Zugriff am 17.4.2000. madsci.wustl.edu/posts/archives/may97/864446241.Ph.r.html.
- Raskin, J. (2000): *The Humane Interface*, Reading, Massachusetts.
- Real3D (1998): Antialiasing. Website, Zugriff am 10.3.1999. www.real3d.com/about/antialiasing.html.
- Sharon, A., Anders, B., Jeff, C., Stephen, D., Paul, G., Eduardo, G., Alex, M., Scott, P., Jeremy, R. und Steve, Z. (2000): Extensible Stylesheet Language (XSL) Version 1.0. Website, Zugriff am 10.5.2000. www.w3.org/TR/2000/WD-xsl-20000327/.
- Sieber, R. und Bär, H. R. (1997): Atlas der Schweiz – Multimedia Version. Adaptierte GIS-Techniken und qualitative Bildschirmgraphik. *GIS und Kartographie im multimedialen Umfeld*. Deutsche Gesellschaft für Kartographie. Bonn, Kirschbaum Verlag.
- Spiess, E. (1996a): Digitale Technologie und graphische Qualität von Karten und Plänen. In: *Vermessung, Photogrammetrie, Kulturtechnik*, Ausgabe 9/1996, S. 467-472.
- Spiess, E. (1996b): *Kartentechnik und Kartenentwurf. Vorlesungsskript*. Zürich.
- Timpf, S. (1997): Cartographic Objects in a Multi-Scale Data Structure. In: *Geographic Information Research: Bridging the Atlantic*, Ausgabe 1/1997, S. 224-234.
- Töpfer, F. (1962): Das Wurzelgesetz und seine Anwendung bei der Reliefgeneralisierung. In: *Vermessungstechnik*, Ausgabe 1/1962, S. 37–42.
- VillageTronic (1998): *MacPicasso 540 User's Manual*. Village Tronic Marketing GmbH, Sarstedt.
- Weibel, R. (1995): Three Essential Building Blocks for Automated Generalization. *GIS and Generalization*. Jean-Claude Müller, Jean-Philippe Lagrange und Robert Weibel. London, Taylor & Francis Ltd. 1.
- Weinman, L. (1998): *WebDesign*. Midas Verlag AG, Zürich.
- Winter, A. M. und Neumann, A. (2000): carto:svg. Website, Zugriff am 21.3.2000. www.carto.net/papers/svg/.