Institut für Geodäsie und Geoinformation

Automatisierte Erzeugung personalisierter ad-hoc-Karten in einem Service-basierten GIS (Mapping on Demand)

Inaugural-Dissertation

zur

Erlangung des Grades

Doktor-Ingenieur

(Dr.-Ing.)

der

Hohen Landwirtschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität

zu Bonn

vorgelegt am 2. April 2009

von

Dipl.-Ing. Jörg Steinrücken

aus Winterberg

Referent: Prof. Dr. L. Plümer
Korreferent: Prof. Dr.-Ing. W. Förstner

Tag der mündlichen Prüfung: 06. August 2009

Erscheinungsjahr: 2009
Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert.

Kurzfassung

In den letzten Jahren wurden mehr und mehr raumbezogene Daten über das World Wide Web zur Verfügung gestellt. Karten und Luftbilder bekannter Anwendungen (z.B. Google Maps) lassen sich direkt nutzen oder mit Daten anderer Angebote verknüpfen und in ihrem Informationsgehalt erweitern. Ein Nutzer hat so die Möglichkeit, unabhängig von Ort und Zeit, zu einer aktuellen Fragestellung beliebige Daten zu integrieren und eigene Karten zu erstellen (Mapping on demand). Benötigte Daten werden zunehmend im Zusammenhang mit Diensten – Funktionen, die den Zugriff auf Daten und deren sachgerechte Verwendung unterstützen – zur Verfügung gestellt. Dies entspricht dem allgemeinen Trend des Internets hin zu einer Service-orientierten Architektur; deren Implementierung durch Web Services löst Daten aus dem Kontext spezifischer Anwendungen und macht sie über definierte Schnittstellen verfügbar.

Herausforderung in einer Service-orientierten Architektur ist zunächst, dass die genutzten Daten weltweit verteilt sind, zu verschiedensten Zwecken erhoben wurden und in unterschiedlichen Formaten vorliegen. Diese Heterogenität wird durch den Einsatz standardisierter Dienste und Formate überwunden. In der vorliegenden Arbeit wird gezeigt, dass im World Wide Web Technologien verfügbar sind, die das notwendige Potential für ein Mapping on demand besitzen. Beim Auffinden geeigneter Dienste und der Nutzung verfügbarer Technologien wird ein Nutzer durch Webportale unterstützt. Diese bündeln Dienste, machen sie strukturiert zugänglich und ermöglichen die Erschließung durch geeignete Werkzeuge. Eine konkrete Umsetzung von Webportalen wird in dieser Arbeit anhand dreier Beispiele aus dem Bereich der Freizeitplanung gezeigt.

Für das Mapping on demand ist von Bedeutung, dass Daten verfügbarer Dienste nicht nur als fertige Karten vorliegen, sondern auch als Kartenbausteine in Form raumbezogener Objekte oder thematischer Layer. Bei der Integration der graphischen Ausgaben dieser Dienste wird offensichtlich, dass die graphische Repräsentation, die jedes Angebot von sich aus mitbringt, isoliert von anderen Angeboten festgelegt ist. Die Kombination beliebiger Quellen führt damit häufig zu Darstellungen, die schon den Mindestanforderungen graphischer Gestaltungsregeln und erst recht den Erfordernissen der Prägnanz nicht genügen, so dass der Inhalt schwer erfassbar ist. Nur eine prägnante Darstellung gibt dem Nutzer die Möglichkeit, Objekte visuell zu differenzieren und effektiv Wesentliches von Unwesentlichem zu unterscheiden. Eine Voraussetzung hierfür ist der Einsatz visuell gut unterscheidbarer Farben.

Kern dieser Arbeit ist der Nachweis, dass die Bestimmung gut unterscheidbarer Farben als Optimierungsproblem formuliert und mit mathematischen Methoden on demand gelöst werden kann. Dies erfordert eine objektivierte Beschreibung des subjektiven Vorgangs der Farbwahrnehmung in einem Bezugssystem (Farbraum), auf dem eine geeignete, der menschlichen Wahrnehmung entsprechende, Metrik definiert ist. Ein solches System steht bspw. mit dem CIELUV-Farbraum zur Verfügung: Wenn darin der Abstand zweier Farben ein gewisses Minimum überschreitet, ist die Verschiedenheit dieser Farben für einen (normalsichtigen) Menschen klar erkennbar. Das Optimierungsproblem besteht dann darin, die minimale Distanz in einem dreidimensionalen Raum zu maximieren. Es handelt sich dabei um ein nichtlineares Problem, das durch eine Vielzahl lokal optimaler Lösungen gekennzeichnet ist. Klassische gradientenbasierte Verfahren berechnen zwar eine lokale, in der Regel aber nicht die global optimale Lösung. Probleme der globalen Optimierung sind im Allgemeinen schwer lösbar (NP-vollständig). Effiziente Verfahren wurden für diskrete Problemstellungen entwickelt, sind aber wegen der spezifischen Randbedingungen auf das vorliegende Problem nicht übertragbar.

In dieser Arbeit wird eine Methode entwickelt, die mehrere Standardverfahren und Lösungsparadigmen integriert. Ausgangspunkt ist ein randomisierter Algorithmus zur Bestimmung geeigneter Startpunkte. Dieser Algorithmus basiert auf der Beobachtung, dass sich Punkte auf dem Rand der konvexen Hülle des Farbraums in besonderer Weise als Startpunkte eignen. Die Lage dieser Punkte wird durch Anwendung eines Verfahrens der lokalen Optimierung verbessert. Ein dreidimensionales Voronoi-Diagramm wird genutzt, um eine suboptimale Nutzung des verfügbaren Farbraums zu identifizieren und Verbesserungsmöglichkeiten herzuleiten.

Das skizzierte Szenario geht zunächst von einem Standardnutzer und Standardbedingungen aus. Die Methode kann aber ohne weiteres auch auf davon abweichende Randbedingungen, z.B. Farbsehschwächen eines Nutzers, Qualität der Farbwiedergabe eines Anzeigegeräts oder Umgebungsbedingungen (abgedunkelter Raum oder helles Sonnenlicht), angepasst werden. In ihrer Auswirkung führen diese Bedingungen zu einer Einschränkung des verfügbaren Farbraums.

Abstract

Providing geospatial data over the World Wide Web has developed rapidly in recent years. Maps and aerial photographs offered by popular applications (e.g. Google Maps) can either be used directly or may be linked to data of other providers to broaden information content. Hence, a user who demands a map to answer a current question is enabled to create his own map by overlaying arbitrary data - regardless of location and time (Mapping on demand). Data is increasingly provided in connection with services - functions, which support access and appropriate use of data. Services correspond to a general trend of realising a service-oriented architecture in the internet. Web services, which implement this architecture, dissolve data from specific applications and provide them by defined interfaces.

The challenge in using services consists in the worldwide distribution of data, different intentions of data collection and the large variety of available formats. This heterogeneity is overcome by applying standardised formats and services. This thesis demonstrates the potential of technologies which are available for a mapping on demand. In finding suitable services and using technologies, a user is supported by web portals. These portals make services accessible in a structured manner and support their exploration by suitable tools. This thesis presents three examples of real implementations of web portals from the domain of recreation planning.

In the scope of mapping on demand it is important to note that data provided by services are not only available as complete maps but also as components (thematic layers or features). Integrating their graphical output shows that graphical representation of data from each source is defined independently and that graphical representations of different sources may conflict. Thus, the ad hoc combination of sources may lead to maps, which do not satisfy basic rules of graphical design or even requirements of a concise cartographic product. But only conciseness enables a user to identify features visually and to distinguish between essential and nonessential contents. This requires the choice of colours which are well distinguishable by human visual perception.

The main contribution of this thesis is the proof that the determination of well distinguishable colours can be formulated as an optimization problem, which can be solved on demand by mathematical methods. This approach requires an objectified description of the subjective process of colour vision in a reference system (colour space), which provides an appropriate metric according to human perception. Particularly the CIELUV colour space satisfies the requirement: If the distance between two colours exceeds a certain minimum, a human with full colour vision is able to distinguish these colours clearly. Thus the optimization problem is to maximise the minimal distance in a three-dimensional space. This is a nonlinear problem, which is characterized by a large number of locally optimal solutions. Classical descend methods find a locally, but usually not the globally optimal solution. In general, global optimization problems are computationally hard to solve (NP-complete). Efficient methods have been developed for solving discrete problems. Due to specific constraints these methods are not applicable to the problem at hand.

This thesis presents a method which integrates several standard methods and solution paradigms. It is based on a randomized algorithm to determine appropriate starting points. The essential observation is that points on the boundary of the convex hull of the colour space are well suited as starting points. Afterwards the location of these points is improved by a method of local optimization. A three-dimensional Voronoi-diagram is used to detect suboptimal solutions and to identify possible improvements.

Initially the scenario outlined above acts on a standard user and standard conditions. The method is adaptable to incorporate further constraints, e.g. user's colour-defective vision, device-specific colour reproduction and ambient conditions (darkened room or sunlight). These constraints result in clipping the available colour space.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

In den letzten Jahren haben das Angebot und die Nutzung raumbezogener Daten über das World Wide Web (WWW) einen rasanten Aufschwung erfahren. Auslöser für diese Entwicklung war auf Seiten der Anbieter eine wachsende Anzahl von Firmen und Institutionen, die Datenbestände allgemein zugänglich gemacht haben. Es lassen sich dabei grob zwei Arten von Angeboten differenzieren:

- *Anwendungsorientierte Angebote* offerieren einen Datenbestand im Rahmen einer bestimmten Anwendung, beispielsweise einer Routenplanung. Die Ausgabe erfolgt als graphische Darstellung in Form topographischer oder thematischer Karten (Bildschirmkarten) aus definierten Quellen.
- *Dienstorientierte Angebote* bieten unabhängig von spezifischen Anwendungen Zugriff auf raumbezogene Datenbestände über offene Schnittstellen (Web Map Service, Abschnitt 2.3.3.2). Der Umfang der Daten kann von einzelnen Objektklassen bis hin zu kompletten topographischen oder thematischen Karten reichen, die Ausgabe erfolgt graphisch und/oder in Form objektbezogener Austauschformate.

Letztgenannte Angebote fügen sich in eine übergeordnete Entwicklung. Diese beinhaltet im Kern einen Wandel der Internetarchitektur von einer reinen *Client-Server-Architektur* hin zu einer *Service-orientierten Architektur*, die durch eine prozessorientierte Sicht in Form von *Diensten* gekennzeichnet ist (vgl. Schill & Springer 2007). Dienste kapseln Funktionalitäten und Daten und machen sie plattformunabhängig über veröffentlichte Schnittstellen in einem Netzwerk verfügbar und nutzbar (Schill & Springer 2007, Dostal et al. 2005). In der konkreten Implementierung durch *Web Services* werden so Anwendungen und Daten des WWW aus festen, meist für einen menschlichen Nutzer bestimmten, Zusammenhängen gelöst und auch maschinellen Adressaten, z.B. anderen Diensten, zugänglich gemacht. In dieser Form verfügbare Daten und Informationen sind so in verschiedenen Anwendungen und neuen Kontexten nutzbar. Daten mit einem Raumbezug stehen für ein *Mapping on demand* bereit.

Das Mapping on demand bedeutet im Kontext dieser Arbeit somit also nicht die Abbildung der Realität in eine modellhafte Repräsentation, sondern die ad-hoc-Erstellung einer Karte, d.h. die graphische Repräsentation von Daten durch Auswahl graphischer Elemente und Attribute für einen spontanen Gebrauch. Wesentliches Merkmal dieser Karte ist, dass sie erst einmal nicht existent ist, sondern auf Anforderung aus vorhandenen Daten und Informationen zusammengestellt wird. Herausforderung dieser ad-hoc-Verknüpfung ist zunächst, dass die Daten weltweit verteilt sind, in unterschiedlichen Formaten vorliegen und zu verschiedensten Zwecken erhoben wurden. Diese Heterogenität in der Koexistenz verschiedener Systeme und Formate wird durch ihre Vereinheitlichung durch *standardisierte Technologien* überwunden. Insbesondere für eine Service-orientierte Architektur ist die Beschreibung von Diensten durch

offene Standards eine wesentliche Voraussetzung (Dostal et al. 2005). Die allgemein durch Standards erreichte *Interoperabilität* ist spätestens seit Verbreitung des Internets ein wichtiges Ziel vieler Anwendungsbereiche und Forschungsgegenstand unterschiedlicher Disziplinen, insbesondere auch des GIS-Bereichs. Im Kontext des WWW sind in den letzten Jahren in einer mehr oder weniger kontinuierlichen Entwicklung eine Reihe von *Standards* erarbeitet worden, die eine ad-hoc-Erstellung von Karten durch Definition geeigneter Technologien des Datenzugriffs, der physikalischen Zusammenführung von Daten und ihrer Integration zu einem kohärenten Ganzen wesentlich unterstützen.

Die Erschließung von dienstorientierten Angeboten, deren Ausgaben sich für einen unmittelbaren Gebrauch durch menschliche Nutzer eignen, erfolgt über Webportale. Diese sammeln und bündeln Informationen über Web Services und stellen einen strukturierten Zugang bereit. Die Exploration und Zusammenführung von Daten verschiedener Quellen wird durch geeignete Werkzeuge unterstützt.

Die graphischen Ausgaben dienstorientierter Angebote raumbezogener Daten versetzen so einen beliebigen Nutzer in die Lage, durch Integration von Daten und Informationen einer oder mehrerer Quellen eigene Karten zu erstellen. Durch die Auswahl bestimmter Daten besteht die Möglichkeit, diese Karten auf individuelle Zwecke und Bedürfnisse abzustimmen. Der Bezug der Daten über das WWW erlaubt einen orts- und zeitunabhängigen Zugriff – eine Nutzung on demand. Die Gefahr dieser Möglichkeiten wird allerdings in der Realität sehr schnell offensichtlich: Die graphische Repräsentation, die jedes Angebot von sich aus mitbringt, ist isoliert von anderen Angeboten festgelegt; die ad-hoc-Kombination beliebiger Quellen führt damit in der Regel zu Karten, die schon den Mindestanforderungen graphischer Gestaltungsregeln und erst recht den Erfordernissen der Prägnanz und damit einer effizienten kartographischen Kommunikation nicht genügen. Beispiele für die Ursachen einer ineffizienten Kommunikation sind schlecht gewählte Zeichengrößen (z.B. Linienbreiten) oder uneinheitliche bzw. sich widersprechende Signaturen. Eine große Bedeutung kommt der Wahl geeigneter Farben zu: Offensichtlich ist eine wesentliche Mindestforderung, dass – eine ausreichende Zeichengröße vorausgesetzt – die verwendeten Farben für das menschliche Auge überhaupt als unterschiedlich wahrgenommen werden können, oder, strenger formuliert, so deutlich verschieden sind, dass sie von einem Betrachter möglichst schnell und fehlerfrei differenziert werden können. Dieses Problem der Bestimmung geeigneter Farben wird in dieser Arbeit durch die Platzierung von Punkten im dreidimensionalen Farbraum beschrieben und als Problem der nichtlinearen Optimierung gelöst.

Erschwert wird die genannte Problematik darüber hinaus durch Einflüsse, die – unabhängig von einer Kartenerstellung on demand – die visuelle Wahrnehmung beeinflussen. Zu nennen sind hier besonders die Abhängigkeiten der Darstellung vom Wiedergabegerät, die spezifisch für bestimmte Geräte bzw. Geräteklassen und deren Anwendungskontext sind. Ein bekanntes Problem ist die Geräteabhängigkeit der Farbwiedergabe, bspw. werden Farben von Präsentationsfolien, die auf einem Bildschirm erstellt wurden, in der Regel über einen Beamer nur unzureichend wiedergegeben. Ebenso häufig macht ein Nutzer mobiler Geräte die Erfahrung, dass eine Darstellung, die auf einem Display in normaler Büroumgebung gut erkennbar ist,

auf einem mobilen Anzeigegerät im Freien, evtl. bei Sonnenschein, kaum noch sichtbar ist. Bei kleineren Geräteklassen wie Handys oder Personal Digital Assistants (PDA) kommt noch die eingeschränkte Größe des Displays hinzu. Ein weiterer limitierender Faktor der visuellen Wahrnehmung sind die Farbsehschwächen eines Betrachters. Diese betreffen ca. 8% der männlichen und 0,4% der weiblichen Bevölkerung (Schläpfer 1993) und schränken offensichtlich die Darstellungsmöglichkeiten durch Verringerung der wahrnehmbaren Farben deutlich ein. Diese beschriebenen Einflüsse lassen sich durch eine *Personalisierung* in der Kartendarstellung berücksichtigen.

1.2 Anwendungsszenarien

Der Einsatz on demand erstellter Karten bietet sich überall dort an, wo raumbezogene Informationen nicht oder nicht ausreichend in vorgefertigten Karten verfügbar sind, sondern erst durch eine Kombination vorhandener Daten in einer für einen bestimmten Zweck erstellten Karte hinreichend prägnant vermittelt werden können. Dementsprechend sind denkbare Einsatzmöglichkeiten sehr breit gefächert. Um für diese Arbeit eine konkretere Vorstellung zu geben, beschreibt dieser Abschnitt drei typische Anwendungsszenarien, von einem spontanen Einsatz in einer Notfallsituation bis hin zu einer wohl vorbereiteten Reiseplanung auf Grundlage raumbezogener Daten. Allen Szenarien ist gemeinsam, dass die erforderliche Karte einem Nutzer in einem sehr engen Rahmen und Kontext ganz bestimmte Informationen vermitteln soll. Der Verfasser benötigt diese Karten entweder für eigene Zwecke oder für einen ganz bestimmten, meist eingeschränkten, Nutzerkreis.

1.2.1 Szenario 1: Notfallsituation

Familie Müller befindet sich auf einer mehrtägigen Wanderung durch die bayerischen Alpen. Zur Orientierung nutzen sie einen PDA, auf den eine vorbereitete Bergtour einer Tourismusbehörde geladen ist. Durch eine Kartendarstellung und eine Sprachausgabe wird Familie Müller entlang mehrstündiger Etappen geführt und auf Besonderheiten längs der Tour aufmerksam gemacht; Start und Ziel jeder Etappe ist jeweils eine Berghütte. Die Navigation erfolgt über das Global Positioning System (GPS).

Trotz positiver Wettervorhersage zieht nach der Hälfte einer Tagesetappe – sehr weit entfernt von Ausgangs- und Endpunkt – überraschend ein Unwetter auf. Ein Blick in die Informationen des PDAs zeigt, dass es entlang der weiteren Route keine Zufluchtsstätten gibt; Informationen abseits der Route sind auf dem PDA nicht enthalten. Glücklicherweise kann sich Herr Müller über sein Handy mit dem Internet verbinden und auf ihm bekannte Informationsquellen über Wanderungen in den Alpen zugreifen. Er findet dort direkt eine Funktion, die nach Eingabe der aktuellen Positionskoordinaten die nächstgelegene Hütte angibt; über eine Routenplanungsfunktion wird der direkte Weg zu dieser Hütte berechnet. Herr Müller lässt sich diese Route in Kombination mit einer topographischen Karte im mittleren Maßstab anzeigen. Die Familie versucht dieser Route möglichst schnell zu folgen, wird allerdings immer wieder aufgehalten, da das Tageslicht trotz aufziehendem Unwetter noch sehr hell ist und Herr Mül-

ler auf dem lichtschwachen Display die dunkelrot dargestellte Route nur sehr schwer im Braun der Höhenlinien ausmachen kann.

Dieses Szenario gibt ein Beispiel für eine dringliche Fragestellung, für die eine ad-hoc-Karte mit wenigen, zielgerichteten Informationen (Route und Topographie entlang der Route), aber einer sehr klaren und eindeutigen Darstellung benötigt wird. Für Familie Müller wäre es hier offensichtlich wünschenswert, wenn die Graphik auf das Darstellungsmedium bzw. auf die äußeren Einflüsse – das helle Umgebungslicht – abgestimmt wäre und wesentliche Informationen „auf einen Blick" identifizierbar wären.

1.2.2 Szenario 2: Konferenzskizze

Herr Schulz bekommt den Auftrag für eine Konferenz in Bonn zwei Karten zusammenzustellen, aus denen alle für die Teilnehmer wichtigen räumlichen Zusammenhänge ersichtlich sein sollen. Die erste Karte soll alle Informationen zur Anreise mit Flugzeug und Auto enthalten, die zweite Karte soll in einem Stadtplan die Anreise mit der Bahn und alle wesentlichen Orte rund um die Konferenz darstellen: Konferenzgebäude, empfohlene Hotels, Orte der Abendveranstaltungen und wichtige Points of Interest für die Freizeitgestaltung (Restaurants, Sehenswürdigkeiten). Beide Karten sollen über die Webpräsenz der Konferenz allen Teilnehmern online zur Verfügung stehen, von dort aber auch bspw. auf mobile Geräte (PDA) heruntergeladen werden können. Die zweite Karte soll zusätzlich während der Konferenz über Beamer angezeigt werden.

Für die erste Karte nutzt Herr Schulz eine topographische Übersichtskarte als Kartengrund und ergänzt diese aus einer Datenquelle topographischer Objekte um Flughafen und Tagungsgebäude. Aus einer weiteren Datenquelle, die das gesamte Netz örtlicher und überörtlicher Straßen Nordrhein-Westfalens enthält, fügt er Autobahnen, Autobahnanschlüsse und Teile des innerstädtischen Straßennetzes hinzu. Die Straßen mit den besten Anfahrtsmöglichkeiten zum Konferenzgebäude hebt er besonders hervor.

Als Grundlage der zweiten Karte nutzt Herr Schulz einen Stadtplan. Darüber ergänzt er das Tagungsgebäude und den Bahnhof, aus einer Datenbank mit touristischen Objekten Hotels, Orte der Abendveranstaltungen, Restaurants und besondere Sehenswürdigkeiten. Zur Darstellung dieser punkthaften Objekte nutzt er Signaturen, die das entsprechende Objekt möglichst treffend repräsentieren und sich gut vom Kartenhintergrund abheben. Zusätzlich stellt Herr Schulz noch die kürzesten Wege zwischen den wichtigsten Orten dar (Bahnhof, Tagungsgebäude, Hotels); diese Wege müssen sowohl von allen anderen Karteninhalten als auch untereinander so unterscheidbar sein, dass alle wesentlichen Informationen auch in gedruckter Form darstellbar sind.

Dieses Szenario beschreibt die Erstellung einer ad-hoc-Karte für einen bestimmten Zweck und für einen überschaubaren Kreis von Adressaten. Der Fokus – insbesondere der zweiten Karte – liegt hier zum einen auf der Kombination verschiedenster Daten, zum anderen auf der Darstellung der Karte auf unterschiedlichen Anzeigegeräten unter verschiedenen Umgebungsbedingungen.

Die Herausforderung besteht zunächst darin, die unterschiedlichen Daten farblich so darzustellen, dass sie für einen Nutzer deutlich unterscheidbar und schnell erfassbar sind. Erschwert wird diese Aufgabe dadurch, dass die Farben bzw. ihre Unterscheidbarkeit nicht nur auf stationären Bildschirmen in – evtl. etwas abgedunkelten – Innenräumen gewährleistet ist, sondern auch auf mobilen Geräten im Freien und über verschiedene Beamer in Räumen unterschiedlicher Beleuchtung. Erfahrungsgemäß weicht gerade die Farbwiedergabe über (nicht kalibrierte) Beamer deutlich von der Farbwiedergabe auf Monitoren oder anderen Displays ab.

1.2.3 Szenario 3: Freizeitplanung

Familie Meier plant einige freie Tage am Niederrhein zu verbringen. Die Familie möchte die Gegend mit dem Rad entdecken, einige Sehenswürdigkeiten besichtigen, mittags gut essen und in fahrradfreundlichen Gasthäusern übernachten. Um sich vor Ort nicht lange um Details kümmern zu müssen, möchte Familie Meier das Gebiet im WWW erkunden und vorab Tagestouren, Besichtigungen und Einkehr- und Übernachtungsmöglichkeiten festlegen.

Meiers suchen dazu mehrere Karten verschiedener räumlicher Auflösungen. Diese ergänzen sie jeweils um eine Darstellung des Radwegenetzes und die Inhalte einer Datenbank mit touristischen Objekten (Sehenswürdigkeiten, Gasthäuser, Hotels, Pensionen), die sich, repräsentiert durch Signaturen, deutlich vor dem Kartenhintergrund abheben. Anhand der Kartendarstellung suchen sie sich im infrage kommenden Raumausschnitt geeignete Objekte für Besichtigungen, Einkehr und Übernachtung. Diverse Informationen über diese Objekte (z.B. Öffnungszeiten, Adressen) sind über die touristische Datenbank verfügbar. Nachdem sich Familie Meier auf Objekte für jede Tagesetappe geeinigt hat, lassen sie sich über eine Routenplanungsfunktion die kürzesten Wege zwischen diesen Objekten ausgeben und verschiedenfarbig in den Kartenbildern der unterschiedlichen Maßstäbe anzeigen.

Dieses Beispiel der Freizeitplanung beschreibt ein Szenario, dessen Fokus auf dem Entdecken liegt. Die Fragestellung ist hier weniger dringlich als beispielsweise in einer Notfallsituation. Damit ist der Zweck zwar festgelegt, die benötigten Daten sind hier allerdings sehr breit gefächert und decken unterschiedlichste Thematiken ab.

Die Herausforderung liegt besonders in der Vielfalt der anzuzeigenden Daten und damit der Vielzahl der benötigten Farben: Alle touristischen Objekte, das Radwegenetz und die einzelnen Routen müssen farblich so dargestellt werden, dass sowohl eine eindeutige Differenzierung untereinander als auch gegenüber dem Kartenhintergrund möglich ist. Im Vergleich zur Konferenzskizze kommt hier erschwerend hinzu, dass eine Integration mit Kartenhintergründen unterschiedlicher Maßstäbe und damit eine Abstimmung mit den Farben verschiedener Karten erforderlich ist.

1.3 Fragestellung und Zielsetzung

Ziel dieser Arbeit ist es, auf Basis von Technologien des WWW die Prägnanz einer sorgfältigen kartographischen Gestaltung mit der Kartenerstellung in Echtzeit in Einklang zu bringen und in ein allgemeines Verfahren zu überführen.

Für das Erreichen dieses Ziels wird von sechs Arbeitshypothesen ausgegangen, die im Verlauf dieser Arbeit verifiziert werden:

- Das WWW stellt die erforderlichen Technologien zur Verfügung.
- Die Nutzung von Diensten und Standards in einer Service-orientierten Architektur ermöglicht einen homogenen Zugriff auf räumliche Daten, die Bündelung von Diensten in Portalen eine komfortable Erschließung dieser Dienste durch einen Nutzer.
- Bei der Abbildung von Daten auf graphische Elemente und Attribute ist die Farbe in ihrer Anwendung zwar anspruchsvoll, hat aber den größten Einfluss auf eine prägnante Darstellung.
- Eine prägnante Darstellung wird durch die Nutzung visuell gut unterscheidbarer Farben erhalten. Die Bestimmung solcher Farben ist als Distanzproblem der nichtlinearen Optimierung im dreidimensionalen Farbraum formulierbar.
- Die Prägnanz lässt sich durch eine personalisierte Farbgebung, d.h. unter Berücksichtigung von Darstellungsmöglichkeiten des Anzeigegeräts, äußeren Einflüssen und individuellen Seheigenschaften des Nutzers, steigern.
- Die Bestimmung der Farben ist in ein allgemeines Verfahren überführbar, das zwar abhängig von Nutzer, Anzeigegerät und Umgebungssituation ist, aber unabhängig vom genutzten Farbraum oder einer spezifischen Anwendung.

Abbildung 1-1 stellt die wesentlichen genannten Bereiche und ihre Zusammenhänge im Überblick dar.

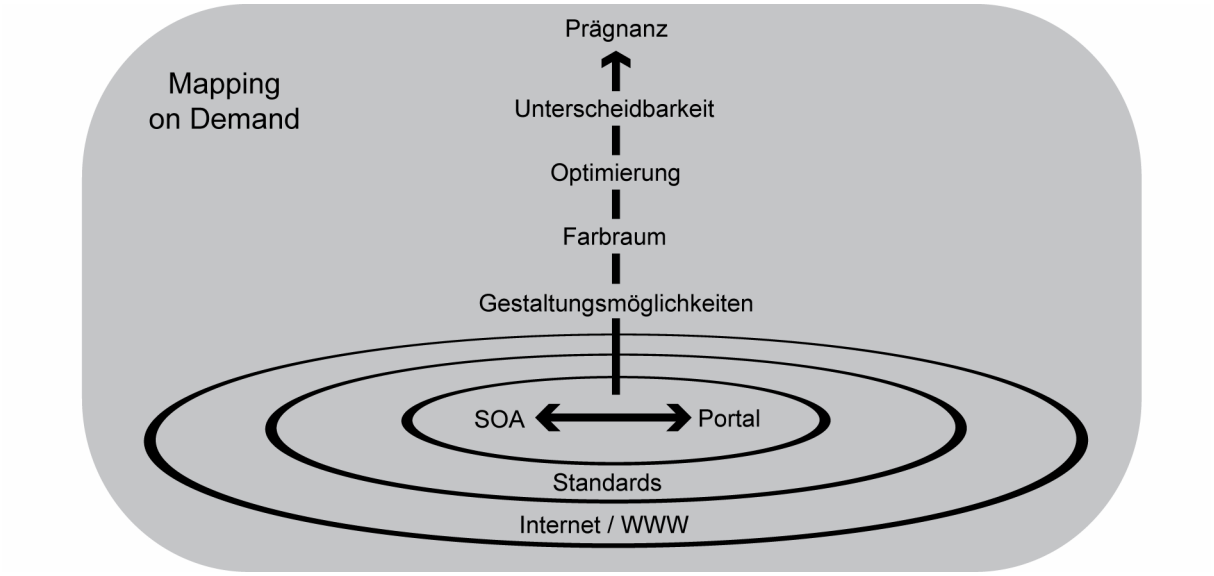


Abbildung 1-1: Übersicht über die wesentlichen Bereiche und Zusammenhänge dieser Arbeit

Die praktische Umsetzbarkeit der erzielten Ergebnisse wird durch die Beschreibung einer geeigneten Software-Architektur und eine prototypische Implementierung gezeigt.

1.4 Struktur der Arbeit

Diese Arbeit gliedert sich wie folgt: Kapitel 2 fasst wesentliche Technologien, die für die Beschaffung und Integration von Daten und Informationen aus dem WWW von Bedeutung sind, zusammen, bevor im Kapitel 3 mit dem Portal-Paradigma ein gängiges Konzept, diese Daten und Informationen einem Nutzer in strukturierter Form zugänglich zu machen, beschrieben wird. Kapitel 4 behandelt die Möglichkeiten der graphischen Darstellung raumbezogener Daten für eine effiziente kartographische Kommunikation; Kapitel 5 fokussiert das Darstellungsmittel der Farbe und beschreibt deren Wahrnehmung und Repräsentation im Hinblick auf die Bestimmung optimaler Farben. Weiterhin werden Möglichkeiten einer Personalisierung der visuellen Kommunikation durch Einbeziehung der Farbwiedergabe digitaler Geräte und Farbsehschwächen eines Nutzers aufgezeigt. Kapitel 6 beschreibt allgemein das Problem der Platzierung von Punkten maximalen Abstands im Raum und fasst relevante Lösungsansätze zusammen. Kapitel 7 fokussiert das Platzierungsproblem auf die Bestimmung optimaler Farben und beschreibt ein allgemeines Verfahren, das eine Lösung on demand ermöglicht. Im achten Kapitel werden die vorangegangenen Ausführungen in einer Architektur zur automatisierten Erzeugung personalisierter Karten gebündelt und die Umsetzbarkeit anhand eines Prototyps gezeigt. Abschließend gibt Kapitel 9 eine Zusammenfassung der erzielten Ergebnisse.

2 Grundlagen der ad-hoc-Beschaffung und Integration verteilter heterogener Informationen aus dem World Wide Web

Die Erstellung von Karten on demand integriert Daten beliebiger, weltweit verteilter Quellen. Diese Quellen sind zunächst durch ihre Heterogenität gekennzeichnet, d.h. die darin verfügbaren Daten wurden von verschiedenen Anbietern in unterschiedlichen Kontexten erhoben und veröffentlicht. Demgegenüber stehen die Vorteile gleichartiger Zugriffs- und Nutzungsmöglichkeiten, die eine physikalische Zusammenführung und den Gebrauch von Daten wesentlich erleichtern. Derartige Zugriffs- und Nutzungsmöglichkeiten werden besonders durch zwei Entwicklungen, die in jüngerer Zeit eine ständig steigende Bedeutung erlangten, unterstützt:

- *Service-orientierte Architektur*: Im Zentrum einer Service-orientierten Architektur (SOA) stehen das Suchen und Nutzen von Diensten über ein Netzwerk (Dostal et al. 2005). Diese Dienste kapseln Daten oder Funktionen und sind durch andere Dienste oder Anwendungen nutzbar, der Zugang über definierte Schnittstellen überwindet die Heterogenität der zugrunde liegenden Systeme. SOA stellen aktuell die letzte Sprosse in der Entwicklung von Programmiersprachen und Netzwerktechniken dar (Dostal et al. 2005). Sie stehen damit in der Tradition einer konsequenten Fortentwicklung in der Informatik bzw. Informationstechnologie. Zu Beginn dieser Entwicklung standen Assemblerprogrammierung und der Remote Procedure Call (entfernter Funktionsaufruf). Mit steigenden Anforderungen und Komplexität wurden Softwareprogramme zunächst in prozeduralen Sprachen, dann in objektorientierten Sprachen umgesetzt; das letzte Glied dieser Kette sind SOA (ebd.).
- *Interoperabilität durch Standards*: Die Heterogenität von Schnittstellen und Daten wird durch die Nutzung von Standards überwunden. Als Standards werden Dokumente bezeichnet, die Regeln und Leitlinien zur Vereinheitlichung von Produkten und Leistungen enthalten (Scherff 2006). Sie sind das Ergebnis einer Standardisierung, eines Konsensverfahrens, in dem sich nationale oder internationale Organisationen auf diese Regeln und Leitlinien geeinigt haben. Standardisierungsverfahren, die von staatlich anerkannten internationalen oder nationalen Institutionen durchgeführt werden, werden auch als Normung bezeichnet, die erstellten Dokumente als *Normen*. Normen können durch Gesetze oder Verordnungen Rechtsverbindlichkeit erlangen (ebd.). Allerdings wird nur im deutschsprachigen Raum zwischen den Begriffen Standard und Norm differenziert, im Englischen ist lediglich der Begriff Standard bekannt. In dieser Arbeit werden die Begriffe synonym gebraucht.

Dieses Kapitel verdeutlicht, dass auf Basis der Infrastruktur des Internets Technologien verfügbar sind, die das notwendige Potential für die ad-hoc-Beschaffung und Integration verteilter Daten besitzen.

Im Folgenden skizziert Abschnitt 2.1 zunächst einige wichtige Architekturen, die den konzeptionellen Aufbau des Internets beschreiben. Von Bedeutung sind insbesondere die Client-Server-Architektur und die Service-orientierte Architektur, die im weiteren Verlauf dieser Arbeit der Charakterisierung verteilter Datenquellen und der Beschreibung ihrer funktionalen Zusammenhänge dienen. Abschnitt 2.2 fasst wichtige standardisierte Schlüsseltechnologien des WWW, die für den Zugriff auf Daten und deren Austausch und Präsentation von Bedeutung sind, zusammen. Abschnitt 2.3 beschreibt die konkrete Implementierung einer Service-orientierten Architektur durch Web Services und zeigt Möglichkeiten des Datenzugriffs auf. Dabei liegt der Schwerpunkt auf offenen Standards, die den Zugriff auf graphische Präsentationen raumbezogener Daten über offene Schnittstellen definieren. Im Abschnitt 2.4 wir beispielhaft das Zusammenwirken verschiedener Technologien in typischen Client-Server-Konstellationen des WWW skizziert, bevor im Abschnitt 2.5 eine Zusammenfassung der Ausführungen dieses Kapitels erfolgt.

2.1 Hard- und Softwarearchitekturen

Eine Verknüpfung verteilter Daten setzt zunächst deren physikalische Zusammenführung bei einem Anwender voraus. Eine dafür geeignete Infrastruktur, die benutzerfreundliche Techniken zum Bezug und Austausch von Daten bereitstellt, ist durch das Internet und dessen wichtigsten Dienst, das WWW, gegeben. Das Internet als Zusammenschluss von Hard- und Softwarekomponenten implementiert das Konzept der *Verteilten Systeme*. Die funktionalen Zusammenhänge dieser Systeme werden durch verschiedene Architekturmodelle beschrieben: Seit jeher sind dies die Client-Server-Architekturen, in jüngerer Zeit immer häufiger die Service-orientierte Architektur.

2.1.1 Verteilte Systeme

Allgemein wird unter einem Verteilten System ein System verstanden, in dem sich Hardware- oder Software-Komponenten auf vernetzten Computern befinden, die nur über den Austausch von Nachrichten kommunizieren und ihre Aktionen koordinieren (Coulouris et al. 2002). Diese Kommunikationsinfrastruktur wird von *Verteilten Anwendungen* genutzt (Hammerschall 2006). Als solche werden Anwendungen bezeichnet, deren Logik auf mehrere, weitgehend unabhängige Komponenten verteilt ist. Diese Komponenten können auf separaten Rechnern eines Verteilten Systems abgelegt sein und erfüllen nur in ihrer Gesamtheit die Aufgaben der Anwendung. Einem Nutzer wird durch eine Verteilte Anwendung eine in sich geschlossene fachliche Funktion zur Verfügung gestellt (ebd.).

Das Internet als großer Zusammenschluss von Computern implementiert ein weltweites Verteiltes System. Einfache Verteilte Anwendungen, Dienste genannt, nutzen diese Infrastruktur (Hammerschall 2006). Populäre Dienste im Internet sind beispielsweise das WWW, Dateizugriffsdienste (z.B. File Transfer Protocol, FTP) oder E-Mail.

Als wichtige Ziele eines Verteilten Systems – die sich auch im Internet und WWW wiederfinden – nennen Tanenbaum & Van Steen (2007) unter anderem:

- *Ressourcen verfügbar machen*: Verschiedene Benutzer und Applikationen greifen auf entfernte Ressourcen zu und nutzen diese gemeinsam. Bei den Ressourcen kann es sich sowohl um Hardware als auch um Software oder Daten handeln.
- *Transparenz*: Ein Verteiltes System wird als transparent bezeichnet, falls es Benutzern und Applikationen verbirgt, dass sich Prozesse und Ressourcen physisch über mehrere Computer verteilen. Eine typische Form der Transparenz ist beispielsweise das Verbergen des Ortes einer Ressource durch einen URI im WWW (Abschnitt 2.2.1).
- *Offenheit*: Ein offenes Verteiltes System bietet Dienste an, die einer allgemein festgelegten Syntax (und Semantik) genügen. Dienste werden im Allgemeinen als Schnittstellen beschrieben, Schnittstellendefinitionen legen beispielsweise den Namen von verfügbaren Funktionen, Parameter(typen) oder Rückgabewerten fest. Schnittstellen lassen sich durch die Begriffe *Interoperabilität* und *Portabilität* charakterisieren. Die Interoperabilität beschreibt das Ausmaß, in dem verschiedene Implementierungen von Systemen oder Komponenten nebeneinander existieren und zusammenarbeiten können, Portabilität das Ausmaß, in dem eine Applikation, die für ein System entwickelt wurde, ohne Veränderungen auf einem anderen System ausführbar ist.

2.1.2 Systemarchitekturen

Ein Verteiltes System kann durch verschiedene, aufeinander aufbauende Architekturkonzepte realisiert werden (Schill & Springer 2007). Zwei dieser Konzepte, die im WWW von Bedeutung sind, werden im Folgenden näher beschrieben.

2.1.2.1 Client-Server-Modell

Das grundlegende Modell zur Strukturierung Verteilter Systeme ist das Client-Server-Modell (Schill & Springer 2007). Als Server wird ein Prozess bezeichnet, der in einem Verteilten System eine bestimmte Funktionalität oder Dienstleistung zur Verfügung stellt, als Client ein Prozess, der diese Funktionalität oder Dienstleistung anfordert, indem er dem Server eine Anfrage sendet und auf die Antwort des Servers wartet (Tanenbaum & Van Steen 2007). Ein Server kann dabei durch den Aufruf weiterer Server selbst zum Client werden.

Der Clientprozess ist durch seine Kurzlebigkeit gekennzeichnet: Nach Erfüllung einer festgelegten Aufgabe wird der Prozess beendet. Der Serverprozess läuft über einen längeren Zeitraum und steht allgemein für Anfragen eines oder mehrerer Clients zur Verfügung (Hammerschall 2006); auf eine Anfrage hin wird eine bestimmte Aufgabe erledigt. Abbildung 2-1 zeigt diesen Zusammenhang anhand eines Sequenzdiagramms in der Syntax der Modellierungssprache UML 2.1. Das Internet bzw. seine Dienste implementieren bzw. nutzen das Client-Server-Modell.

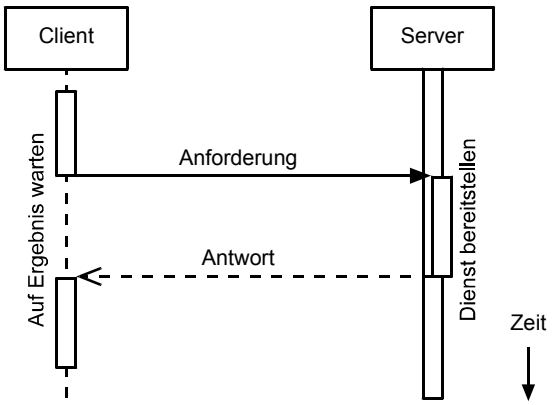


Abbildung 2-1: Sequenzdiagramm zur Zusammenarbeit zwischen Client und Server

Die Kommunikation zwischen Client und Server erfolgt durch den Austausch von Nachrichten, nach Art der Kommunikation wird dabei zwischen synchroner und asynchroner Kommunikation unterschieden (Hammerschall 2006). Bei der synchronen Kommunikation wird die Ausführung des anfragenden Senders (Client) einer Nachricht so lange blockiert, bis die Antwort des Empfängers (Server) eingetroffen ist. Bei der asynchronen Kommunikation bleibt der Sender nach Absetzten einer Nachricht weiterhin aktiv und wird parallel zum Empfänger weiter ausgeführt. Abbildung 2-2 stellt die Sequenzdiagramme dieser beiden Arten der Kommunikation gegenüber. Eine synchrone Nachricht wird dabei durch einen geschlossenen Pfeil, eine asynchrone Nachricht durch einen offenen Pfeil gekennzeichnet. Die Bedeutung der asynchronen Kommunikation wird besonders bei der Nutzung von Ajax deutlich (Kap. 2.2.3).

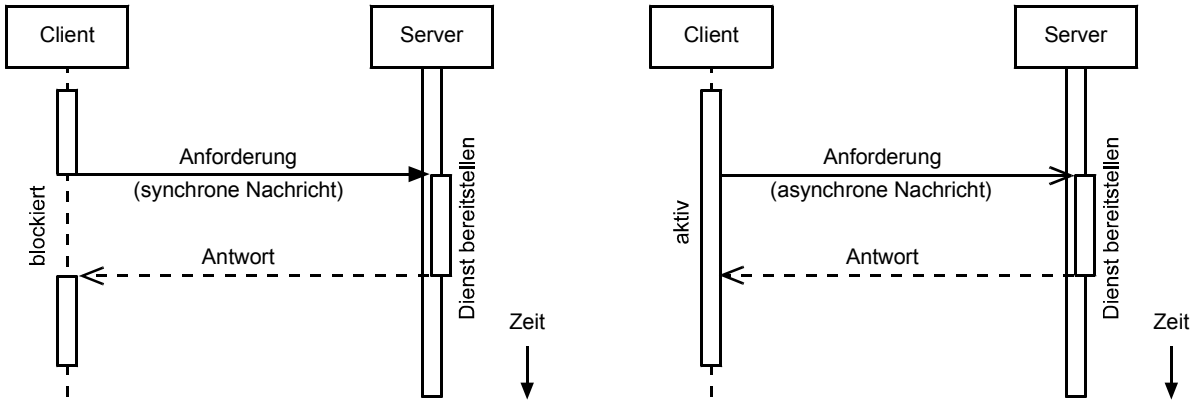


Abbildung 2-2: Sequenzdiagramme zur synchronen (links) und asynchronen (rechts) Kommunikation

Die einheitliche Kommunikation zwischen Client und Server wird durch verschiedene Protokolle geregelt, die einem Nutzer in der Regel verborgen bleiben. Für diesen treten nur die Protokolle in Erscheinung, die von Anwendungen zum Zugriff auf das Internet direkt implementiert werden. Beispiele sind das HTTP-Protokoll (Abschnitt 2.2.1) zur Kommunikation im WWW (meist mit einem Webbrowser) und das FTP-Protokoll zur Dateiübertragung.

2.1.2.2 Client-Server-Architekturen

Ein Verteiltes System lässt sich durch das Client-Server-Modell nicht ausreichend beschreiben; so ist beispielsweise häufig keine klare Unterscheidung zwischen Client und Server möglich (Tanenbaum & Van Steen 2007). Eine systematische Beschreibung von Client-Server-Architekturmodellen erfolgt deshalb vielfach anhand funktionaler Elemente typischer Applikationen. Diese Elemente, Anwendungsschichten oder Tiers (engl.: Schicht) genannt, beschreiben die Aufteilung einer Anwendung in verschiedene Ebenen. Üblicherweise nutzen Anwendungen drei Schichten (Three-tier-architecture) (Shan & Earle 1998; Tanenbaum & Van Steen 2007):

1. Die Schicht der *Benutzeroberfläche* oder *Präsentationsebene* stellt die – zumeist interaktive – Schnittstelle der Anwendung zum Benutzer dar.
2. Die *Verarbeitungsschicht* (Anwendungslogik) enthält die Kernfunktionalität einer Applikation; sie verbindet Benutzeroberfläche und Datenschicht, nimmt Eingaben des Benutzers entgegen oder verarbeitet Daten.
3. Die *Daten*- oder *Persistenzschicht* enthält Anwendungen, die die benötigten Daten verwalten und persistent, d. h. dauerhaft, speichern.

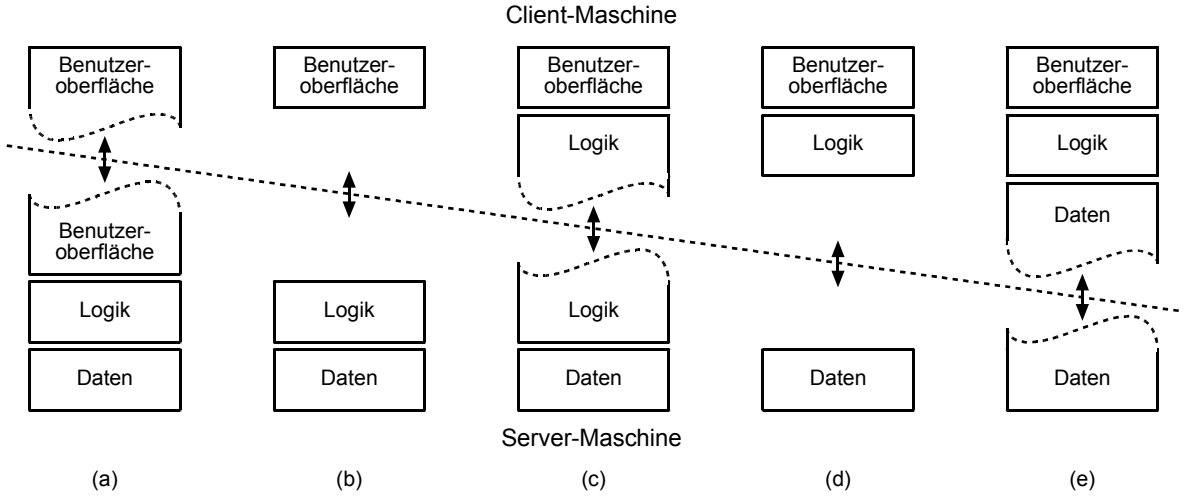


Abbildung 2-3: Möglichkeiten der Aufteilung von Anwendungsschichten auf Client und Server (nach Tanenbaum & Van Steen 2007)

Diese drei Schichten lassen sich auf verschiedene Weise auf die physische Struktur eines Client-Server-Modells verteilen. Shan & Earle (1998) geben folgende Differenzierung wieder (vgl. auch Abbildung 2-3):

a) *Verteilte Präsentation*: Der Client wird lediglich für einen Teil der Präsentation genutzt; Beispiel ist die Darstellung von HTML-Seiten in einem Webbrowser (Abschnitt 2.4.1).

b) *Remote Präsentation*: Die gesamte Präsentation erfolgt auf dem Client, Verarbeitung und Datenhaltung auf dem Server.

c) *Verteilte Funktion*: Neben der gesamten Präsentation wird auch ein Teil der Anwendungslogik auf den Client verlagert.

d) *Entfernter Datenzugriff*: Die gesamte Anwendungslogik wird auf den Client verlagert und greift von dort auf die entfernt liegenden Daten zu.

e) *Verteilte Datenbank*: Zusätzlich zu Präsentation und Anwendungslogik umfasst der Client auch einen Teil der Datenhaltung, die übrige Datenhaltung kann auf einem oder mehreren Servern abgelegt werden.

Abhängig davon, welche Schichten auf dem Client ausgeführt werden, wird auch von einem Thin oder Slim Client bzw. einem Thick oder Fat Client gesprochen (Shan & Earle 1998; Hammerschall 2006). Beim Thin Client wird lediglich die Benutzeroberfläche oder ein Teil davon auf dem Client ausgeführt, Applikation und Datenbank liegen komplett auf dem Server. Je mehr der Präsentation und der Anwendungslogik auf den Client verschoben werden, desto „schwerer“ wird dieser. Beim Thick Client schließlich enthält der Client die gesamte Anwendungslogik.

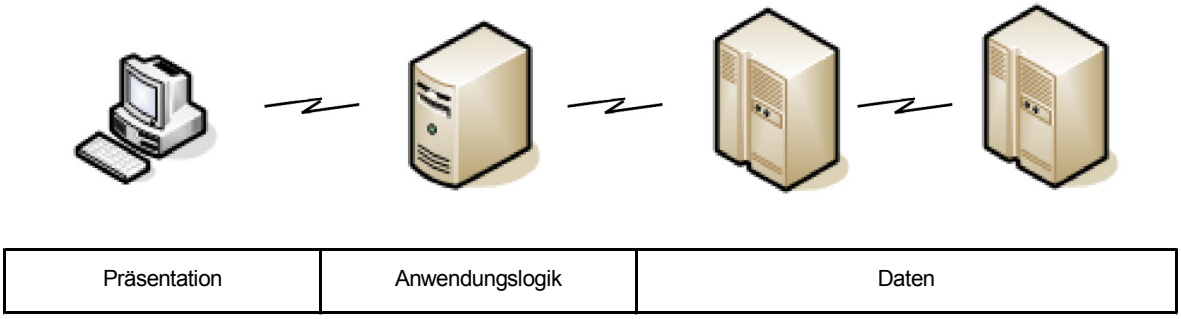


Abbildung 2-4: Logische Drei-Tier- und physikalische Vier-Tier-Architektur

Das Konzept der Tier-Architektur wird nicht nur auf die Software-Konfiguration eines Client-Server-Modells angewandt, sondern auch auf die Hardware-Konfiguration (Shan & Earle 1998). So lassen sich die oben beschriebenen Anwendungsschichten in unterschiedlicher Weise auf die Hardware-Knoten eines Verteilten Systems aufteilen. Eine Zwei-Tier-Architektur nutzt dann beispielsweise einen Client-Rechner für die Präsentationsschicht und einen Server-Rechner für Logik und Datenhaltung. Abbildung 2-4 zeigt ein Beispiel für eine logische Drei-Tier- und physikalische Vier-Tier-Architektur.

2.1.2.3 Service-orientierte Architektur

Eines der jüngsten Paradigmen der Softwaretechnologie ist die Service-orientierte Architektur (SOA). Eine mögliche Definition lautet (Dostal et al. 2005):

> *„Unter einer SOA versteht man eine Systemarchitektur, die vielfältige verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.“*

Dieses Architekturmodell stellt Dienste als Bausteine Verteilter Anwendungen in den Mittelpunkt (Schill & Springer 2007). Dabei wird unter einem Dienst oder Service eine Softwarekomponente verstanden, die von einem *Dienstanbieter* lokal oder über ein Netzwerk zur Verfügung gestellt wird und von einem *Dienstnutzer* – in der Regel ein anderer Dienst – genutzt werden kann (Dostal et al. 2005). Diese Nutzung und damit der Zugriff auf Daten und Funktionalität eines Dienstes ist allgemein oder beschränkt (nach Authentifizierung) und nur über eine genau definierte öffentliche Schnittstelle möglich; einem potenziellen Nutzer eines Dienstes muss dazu eine vollständige Schnittstellenbeschreibung in maschinenlesbarer Form vorliegen.

In einer SOA werden verschiedene Dienste zu komplexen Geschäftsprozessen und –abläufen verknüpft (Schill & Springer 2007). Die genutzten Dienste sind dabei lose gekoppelt, d.h. sie werden von Anwendungen oder anderen Diensten erst zur Laufzeit gesucht, gefunden und in den Programmablauf eingebunden (Dynamisches Binden) (Dostal et al. 2005). Um das dynamische Suchen und Einbinden eines Dienstes gut zu ermöglichen, sollte er in einem *Dienstverzeichnis* (Registry) registriert sein. Die zentralen Aktionen einer SOA lassen sich also in Kürze durch das Anbieten, Suchen und Nutzen von Diensten charakterisieren. Abbildung 2-5 zeigt die drei Rollen der SOA und ihr Zusammenwirken.

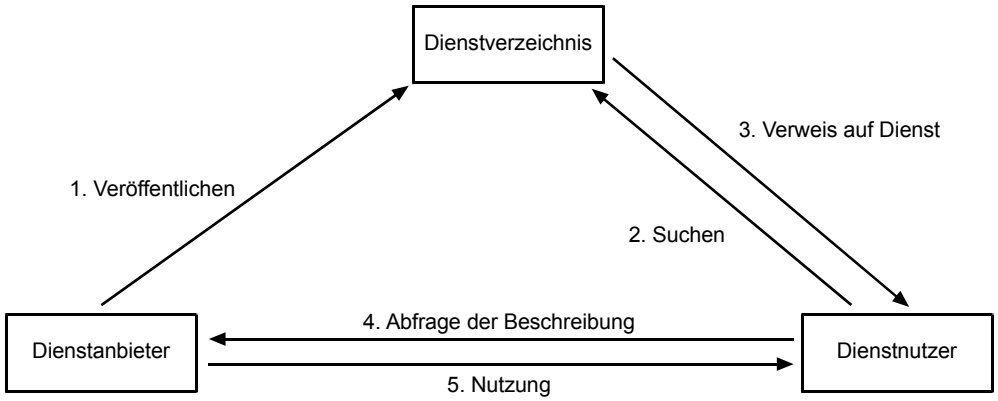


Abbildung 2-5: Magisches Dreieck einer Service-orientierten Architektur (nach Dostal et al. 2005)

Eine konkrete Implementierung des Konzepts einer SOA ist durch Web Services gegeben, die im Abschnitt 2.3 vertiefend ausgeführt werden.

2.2 Schlüsseltechnologien im World Wide Web

Für das Mapping on Demand werden Technologien zum Zugriff auf Daten, deren Austausch und ihre Präsentation im Webbrowser eines Nutzers benötigt. Für diese Zwecke sind im WWW verschiedene Schlüsseltechnologien verfügbar, die von nicht-kommerziellen Organisationen standardisiert sind und eine weit reichende Verbreitung gefunden haben:

- *Datenzugriff*: Die eindeutige Bezeichnung einer Datenquelle wird durch einen *Uniform Resource Identifier* (URI) gewährleistet, Operationen für die Abfrage dieser Quelle stellt das *Hypertext Transport Protocol* (HTTP) zur Verfügung.

- *Datenaustausch und -speicherung*: Die *Extensible Markup Language* (XML) dient als Format zum Austausch und zur Speicherung der Semantik von Anwendungsobjekten. Austausch meint dabei im Kontext des WWW den Datentransfer zwischen Client und Server bzw. zwischen den Diensten einer Service-orientierten Architektur.
- *Darstellung und Präsentation*: Die Hypertext Markup Language (HTML) dient zusammen mit den Cascading Style Sheets (CSS) als Format zur Speicherung, Darstellung und Präsentation textbasierter Daten. Diese Technologien stellen die Grundlage für die Erstellung interaktiver Anwendungen des WWW dar. Interaktion und dynamische Änderungen von HTML und CSS werden durch ergänzende Technologien erreicht, namentlich *JavaScript*, *Dynamisches HTML* (DHTML) und *Asynchronous JavaScript and XML* (Ajax).

Die genannten Technologien werden im Folgenden kurz charakterisiert. Für die Darstellung raumbezogener Daten werden darüber hinaus weitere Daten- bzw. Graphikformate benötigt. Auf deren Beschreibung wird in dieser Arbeit verzichtet. Näheres zu Datenformaten der digitalen Kartographie findet sich bspw. in Olbrich et al. (2002), zu Graphikformaten im WWW bspw. in Münz & Nefzger (1999).

2.2.1 Datenzugriff

Der Zugriff auf Web Services oder verteilte Daten erfordert deren eindeutige Bezeichnung, Informationen über ihren Ort sowie geeignete Zugriffsmöglichkeiten. Im Kontext des WWW sind hierfür besonders zwei Standards der Internet Society (ISOC[1]) von Bedeutung: Der *Uniform Resource Identifier* zur eindeutigen Identifizierung einer (Daten)quelle und das *Hypertext Transport Protocol*, das Transportprotokoll des WWW, das auf Anwendungsebene Möglichkeiten zum Datenzugriff bietet.

Uniform Resource Identifier (URI)
Ein Uniform Resource Identifier (eindeutiger Quellenbezeichner) besteht aus einer Zeichenfolge, die eine abstrakte oder physische Quelle eindeutig identifiziert; diese Quelle ist dabei keinerlei Einschränkung unterworfen, also beispielsweise nicht auf das Internet beschränkt. Die Syntax eines URI ist durch einen Standard der ISOC festgeschrieben und als RFC 3986 (Berners-Lee et al. 2005) veröffentlicht.

Ein URI kann weiter unterschieden werden in:

- *Uniform Resource Locator* (URL) identifizieren eine Quelle durch die Angabe ihres primären Zugriffsmechanismus und geben dadurch ihren „Ort“ in einem Netzwerk an. Beispiele sind Quellen, die über existierende Netwerkprotokolle (HTTP, FTP) erreichbar sind (z.B. ein Dokument: http://www.ikg.uni-bonn.de/index.html).

[1] Die ISOC ist eine internationale Organisation, die federführend in der Pflege, Administration und Weiterentwicklung des Internets ist (http://www.isoc.org, zuletzt geprüft am 31.03.2008).

- Als *Uniform Resource Names* (URN) werden Namen in einem allgemeinen, ortsunabhängigen, persistenten Namensraum bezeichnet (Berners-Lee 1994).

Die generische Syntax eines URI besteht aus einer hierarchischen Struktur von Komponenten (nach Berners-Lee et al. 2005):

```
<Scheme>://<Authority>[/<Path>][?<Query>][#<Fragment>]
          [<User>@]<Host>[:<Port>]
```

<Scheme> bezeichnet hier das verwendete Netzwerkprotokoll (bspw. HTTP), <Authority> besteht aus der optionalen Angabe eines Benutzers (diese wird bei öffentlichen URLs im WWW meist weggelassen), dem Host und einem optionalen Port. Der Host ist im Falle des WWW meist eine *Domain*, ein registrierter Name, der einen Rechner im Internet eindeutig identifiziert (z.B. uni-bonn.de). Für den Port gibt HTTP 80 als Standard an, die Angabe kann dann entfallen. <Path> gibt Pfad und Namen eines bestimmten Dokuments oder einer Datei unterhalb (im Namensraum) der spezifizierten Authority an (z.B. docs/index.html). Der <Query>-Teil einer URL wird durch ein Fragezeichen eingeleitet. Er enthält weitere nichthierarchische Daten, die eine Quelle genauer spezifizieren, hat aber keinerlei Einfluss auf die Referenzierung des in <Path> beschriebenen Dokuments. Der letzte Teil einer URL, <Fragment>, verweist auf einzelne Teile einer Quelle, beispielsweise auf einen bestimmten Bereich eines HTML-Dokuments; die Abtrennung erfolgt durch die Raute (#).

Hypertext Transfer Protocol (HTTP)

Das Hypertext Transfer Protocol ist ein Protokoll zur Datenübertragung in einem verteilten System. HTTP ist ein applikationsspezifisches Protokoll, d.h. es gehört der Anwendungsschicht an und wird direkt von Anwendungsprogrammen implementiert. Die gesamte Client-Server-Kommunikation im WWW basiert auf HTTP und wird dort direkt von Webbrowsern und -servern genutzt. HTTP ist ein standardisiertes Protokoll der ISOC und als RFC 2616 in der Version HTTP/1.1 (Fielding et al. 1999) veröffentlicht.

HTTP stellt verschiedene Methoden zur Verfügung, die es dem Client erlauben, die Ausführung bestimmter Operationen auf dem Server anzufordern. In dieser Arbeit sind die Methoden GET und POST von Bedeutung.

Durch die *GET-Methode* werden Daten von einem Server angefordert, indem der Client eine Anfrage in Form eines URIs (vgl. vorangegangenen Abschnitt) sendet. Der URI kann sowohl auf statische Dokumente verweisen als auch auf Prozesse, die dynamisch erstellte Daten zurückgeben. Diese Prozesse lassen sich durch Parameter im Query-Teil steuern; im Query-Teil können auch Daten an den Server gesendet und dort gespeichert werden.

Durch die *POST-Methode* erhält ein Server von einem Client die Anforderung, einem Dokument oder allgemein einer Ressource Daten hinzuzufügen. Die betreffende Ressource wird wiederum durch einen URI spezifiziert, die zu speichernden Daten sind üblicherweise Name-Wert-Paare, die aus den Eingaben eines Benutzers in ein HTML-Formular einer Webseite stammen; die Übertragung erfolgt nicht im Query-Teil der URI, es lassen sich so auch größere

Dokumente bzw. Datenmengen übertragen. In der Praxis lassen sich GET und POST in ähnlicher Weise zum Austausch und zur Anforderung von Daten nutzen.

2.2.2 Datenaustausch und Speicherung

Die Extensible Markup Language (XML) ist eine generische Auszeichnungssprache zur strukturierten Repräsentation von Daten. XML-Dokumente liegen im Klartextformat vor und werden sehr häufig als Austauschformat genutzt. Verschiedene Technologien im weiteren Verlauf dieses Kapitels basieren auf XML.

Die genannten Technologien stellen *XML-Anwendungen* dar. Dies sind neue Auszeichnungssprachen, die mit Hilfe der Metasprache XML definiert werden, indem XML-Elemente, deren Attribute und Beziehungen untereinander festgelegt werden. XML-Elemente sind frei benennbar und werden durch sogenannte Tags gekennzeichnet. Die komplette Struktur eines XML-Dokuments besteht aus hierarchisch verschachtelten Elementen und muss das Kriterium der Wohlgeformtheit erfüllen. Neben anderen gelten dafür folgende Bedingungen:

- Das Dokument besitzt ein Wurzelelement.
- Ein Element mit Inhalt wird durch einen öffnenden und einen schließenden Tag gekennzeichnet (`<plz>53115</plz>`); Elemente ohne Inhalt können zusammengezogen werden (`<plz/>`).
- Elemente dürfen sich nicht überlappen, d.h. ein nach einem Element A geöffnetes Element B muss vor dem Schließen von A bereits geschlossen sein (`<adresse><plz>53115</plz></adresse>`).

Alle getroffenen Festlegungen einer XML-Anwendung werden in einer Dokumenttypdefinition (DTD) (vgl. Harold & Means 2005) oder einem XML-Schema (vgl. Fallside & Walmsley 2004) festgeschrieben. Ein XML-Dokument, das einen Verweis auf eine DTD oder ein XML-Schema enthält, kann gegen die darin enthaltenen Festlegungen validiert werden. Sind alle Festlegungen erfüllt, wird das XML-Dokument als valide (gültig) bezeichnet.

Soll eine XML-Anwendung bzw. die darin festgelegten Elemente global eindeutig sein, muss ein Namensraum definiert werden. Die Deklaration eines solchen Namensraums erfolgt durch die Vergabe eines URI (Abschnitt 2.2.1).

XML ist als eine vereinfachte Teilmenge der *Standard Generalized Markup Language* (SGML) definiert, bei der es sich ebenfalls um eine Metasprache zur Definition von Auszeichnungssprachen handelt. Während SGML durch die International Organisation of Standardization (ISO[2]) als ISO-Norm 8879 standardisiert ist, ist XML eine Empfehlung (Recom-

[2] Die ISO ist die internationale Vereinigung nationaler Standardisierungsorganisationen (http://www.iso.org, zuletzt geprüft am 31.03.2008).

mendation) des World Wide Web Consortiums (W3C[3]). Aktuell liegt die Version 1.1 in vierter Auflage vor (Bray et al. 2006).

2.2.3 Darstellung und Präsentation

Wichtige Technologien zur Darstellung und Präsentation im WWW wurden einführend bereits genannt, nachfolgend wird eine kurze Beschreibung gegeben.

Hypertext Markup Language (HTML)
Die Hypertext Markup Language (HTML) ist eine Auszeichnungssprache für die Beschreibung des logischen Aufbaus textorientierter Dokumente. Elemente eines solchen Dokuments sind beispielsweise Überschriften, Textabsätze oder Listen; Graphiken und multimediale Inhalte werden durch eine Referenz auf ihre Quelle (URI) eingebunden und erst bei der Anzeige der Elemente im Webbrowser in den Text integriert. Wesentliches Charakteristikum von HTML sind dessen Hypertext-Eigenschaften: Durch die Einbindung von Hyperlinks können Verweise zwischen beliebigen Dokumenten im WWW definiert, und so ein Netz aus Inhaltselementen erstellt werden.

Die Elementstruktur eines HTML-Dokuments ist der Struktur einer XML-Anwendung sehr ähnlich. HTML wurde allerdings zeitlich vor XML entwickelt und ist keine XML-Anwendung. Stattdessen wurde HTML selbst mit Hilfe von SGML als SGML-Anwendung definiert. Aktuell (Januar 2008) liegt HTML in der Version 4.01 als Empfehlung des W3C vor (Raggett et al. 1999); die Version 5 ist als Working Draft verfügbar (Hickson & Hyatt 2008).

Mit der zunehmenden Verbreitung von XML wurde es als notwendig erachtet, HTML XML-konform zu formulieren. Aus diesem Grunde wurde HTML 4.0 als eine XML-Anwendung unter der Bezeichnung „Extensible Hypertext Markup Language“ (XHTML) in der Version 1.0 veröffentlicht (W3C 2002). Inhaltlich sind HTML und XHTML identisch. Allerdings muss ein XHTML-Dokument die für XML-Dokumente geltenden Regeln erfüllen. Einen vertiefenden Überblick über die Unterschiede gibt die W3C-Empfehlung (W3C 2002).

Aktuelle W3C-Empfehlung von XHTML ist die Version 1.1 (Altheim & McCarron 2001). Derzeit liegt eine Überarbeitung von XHTML 1.1 (2. Auflage) als Working Draft vor (McCarron & Ishikawa 2007), parallel wird auch schon die Version 2.0, ebenfalls als Working Draft, erarbeitet (Axelsson et al. 2006).

Sofern eine Unterscheidung nicht zwingend notwendig ist, wird im Folgenden immer die Bezeichnung (X)HTML genutzt.

[3] Das W3C ist ein internationales Industriekonsortium, das offene Standards für das WWW erarbeitet. Die Standards des W3C werden als Empfehlungen (Recommendation) bezeichnet (http://www.w3.org, zuletzt geprüft am 31.03.2008).

Cascading Style Sheets (CSS)
Bei (X)HTML handelt es sich um eine reine Seitenbeschreibungssprache, d.h. es werden Aufbau und Inhalt und damit eine gewisse semantische Struktur eines Dokuments definiert (z.B. Überschriften verschiedener Ordnung und Textabsätze), die genaue Darstellung im Webbrowser hängt von (X)HTML-Regeln und Voreinstellungen des Browsers ab. So wird beispielsweise der Inhalt der meisten Elemente fließend untereinander geschrieben; Schriftarten und Schriftgrößen werden vom Browser bestimmt.

Weitergehende Möglichkeiten für die Umsetzung beliebiger Layouts bieten Cascading Style Sheets (CSS). Als Ergänzungssprache für strukturierte Dokumente entwickelt (nicht nur für (X)HTML) ergänzt sie eine Seitenbeschreibungssprache und erlaubt das Festlegen von Eigenschaften beliebiger Elemente. Beispiele sind die pixelgenaue Positionierung von Elementen, die Formatierung von Schrift oder die Angabe von Hintergrundfarben und Hintergrundbildern.

Genau wie (X)HTML wird auch CSS vom W3C standardisiert. Seit 1998 ist die Version 2 (Level 2) aktuell (Bos et al. 1998), die Version 2.1 steht zur Verabschiedung an (Bos et al. 2007). Auch die Version 3 ist bereits in Vorbereitung.

JavaScript und Document Object Model
(X)HTML und CSS sind zunächst Formate für statische Darstellungsformen und bieten nur sehr rudimentäre Möglichkeiten, dynamisch auf Eingaben eines Nutzers zu reagieren. In (X)HTML sind dies das Anklicken eines Hyperlinks und das Ausfüllen und Abschicken von Formularen. Die Reaktion auf eine solche Aktion ist jeweils das Laden komplett neuer Dokumente vom Server. JavaScript bietet darüber hinausgehende Möglichkeiten, Maus- und Tastatureingaben zu erkennen und bereits im Browser dynamisch darauf zu reagieren. Damit geben Webseiten ihren statischen Dokument-Charakter auf und können wie Programme oder Anwendungen wirken.

JavaScripte werden einmalig vom Server auf den Anwender-Rechner geladen und zur Laufzeit im Webbrowser ausgeführt. Eine weitere Kommunikation zwischen Browser und Server ist im weiteren Verlauf nur notwendig, falls dies gewünscht ist (s.u.).

Bei der Nutzung von JavaScript zum Datenbezug in einer Service-orientierten Architektur ist zu beachten, dass die Möglichkeiten der Sprache auf die Domain einer Webseite, in die ein Script eingebettet ist, beschränkt sind. Nach dieser „Richtlinie gleicher Herkunft“ (Flanagan 2002) darf ein Skript nur auf Elemente und Dokumente zugreifen, die aus der gleichen Quelle (Domain) stammen. Ein Cross-Domain-Zugriff oder Cross-Domain-Scripting, d.h. der Zugriff von Skripten eines Servers A auf Elemente oder Dokumente eines Servers B stellt ein hochkritisches Sicherheitsproblem dar; in diesem Fall könnten sensible Daten aus dem Kontext des Servers A auf den Server B gelangen (Steyer 2006, Segor 2006).

Die wichtigsten Schnittstellen zwischen JavaScript und (X)HTML sind durch *Event-Handler* und das *Document Object Model (DOM)* definiert. Event-Handler können als Attribute in bestimmten (X)HTML-Tags angegeben werden; dort registrieren sie Ereignisse (Aktivitäten

des Nutzers), die dieses Element betreffen, und lösen eine Aktion aus. Beispielsweise reagiert der Handler `onmouseover` auf das Überfahren eines (X)HTML-Elements mit der Maus.

Das DOM definiert eine Plattform- und Programmiersprachen-unabhängige Schnittstelle für den dynamischen Zugriff und die Änderung von Inhalt, Struktur und Eigenschaften von XML- und (X)HTML-Dokumenten. Es repräsentiert alle im Dokument enthaltenen Elemente und deren Inhalt als Knoten in einer Baumstruktur. Im Webbrowser ist mittels JavaScript lesender und schreibender Zugriff auf den gesamten Baum und jeden Knoten möglich.

JavaScript wurde ursprünglich 1995 von der Firma Netscape als Skriptsprache für den Webbrowser Netscape Navigator entwickelt. Die Firma Microsoft konterte aus rechtlichen Gründen mit der Implementierung der eigenen Skriptsprache JScript in ihre Browser (Mintert & Kühnel 2000). Dieser Umstand fiel allerdings in der Praxis kaum ins Gewicht, da der Sprachumfang von JScript weitgehend äquivalent zu dem von JavaScript ist (Flanagan 2002). Seit 1997 wurde auf Basis dieser beiden Sprachen von der Ecma International[4] die *ECMAScript Language* durch die Spezifikation ECMA-262 standardisiert (ECMA 1999). Im Kontext browserseitiger Programmierung wird heute allerdings nach wie vor allgemein von JavaScript gesprochen; die tatsächlich in Produkten implementierten Sprachen sind streng genommen weiterhin uneinheitlich.

Das DOM liegt als Empfehlung des W3C in der 3. Erweiterung vor (Le Hors et al. 2004).

Dynamisches HTML (DHTML)
Dynamisches HTML (DHTML) stellt keine Erweiterung von HTML oder gar eine eigene Programmiersprache dar. Es handelt sich vielmehr um einen Sammelbegriff, der die oben genannten Technologien bzw. Sprachen (X)HTML, CSS, JavaScript und DOM subsumiert. Damit steht DHTML für die Anwendung von Techniken, die die Starrheit einer Webseite überwinden und dynamische Änderungen von Seitenelementen im Webbrowser ermöglichen.

Asynchronous JavaScript and XML (AJAX)
Neuere Konzepte des WWW (vgl. Web 2.0, Abschnitt 2.4.2) sind sehr eng mit dem Begriff Ajax verbunden. Ebenso wie DHTML steht *Asynchronous JavaScript and XML* (Ajax) für eine Kombination bekannter Techniken, deren Aufzählung allerdings weniger einheitlich (Spanneberg & Mintert 2007) ist, als im Falle von DHTML. Nach Garrett (2005), der den Begriff Ajax geprägt hat (Steyer 2006), werden darunter unter anderem (X)HTML, CSS, DOM, XML, XSLT (Sprache zur Transformation einer XML-Anwendung in einer andere), JavaScript und asynchrone Kommunikation subsumiert.

Besonderes Merkmal von Ajax ist ebendiese asynchrone Kommunikation. In „traditionellen" Webauftritten (Abschnitt 2.4.1) folgte auf Aktionen des Nutzers in der Oberfläche einer Webanwendung (z.B. Ausfüllen und Abschicken eines Formulars) das Nachladen und Anzeigen

[4] Die Ecma International ist ein Zusammenschluss von Industrieunternehmen zur Standardisierung im Bereich der Informations- und Kommunikationstechnologie (http://www.ecma-international.org, zuletzt geprüft am 26.08.2008).

kompletter (X)HTML-Seiten. Ein kontinuierliches Arbeiten war so kaum möglich. Dagegen verhält sich eine Ajax-Anwendung asynchron zum Verhalten der Webapplikation an der Oberfläche (vgl. Abbildung 2-6). Nach dem ersten Laden einer Webseite und der zugehörigen Ajax-Funktionalität bleibt diese Seite im Folgenden für den Nutzer stabil, die weitere Kommunikation zwischen Client und Server erfolgt im Hintergrund. Im Verlauf dieser Kommunikation können kontinuierlich Daten vom Server nachgeladen und ganz gezielt in die bereits angezeigte Webanwendung integriert werden. Es lässt sich so mit Webanwendungen die Funktionalität von Desktop-Programmen imitieren.

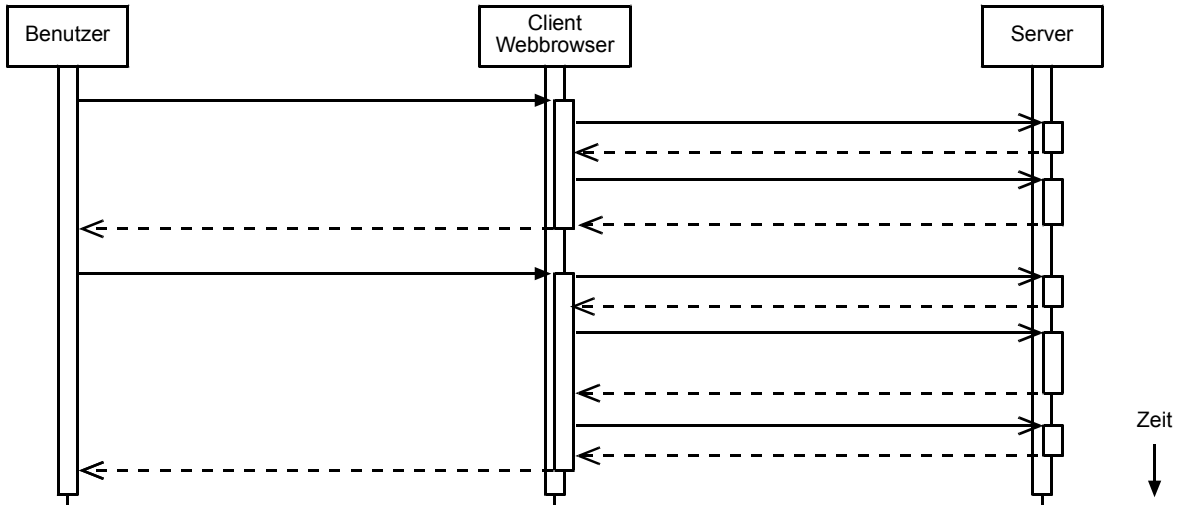


Abbildung 2-6: Sequenzdiagramm des Kommunikationsablaufs zwischen Nutzer, Webbrowser und Webserver bei der Nutzung von Ajax

Asynchrone Serveranfragen mit Ajax erfolgen clientseitig mit JavaScript. Kern der Ajax-Funktionalität ist dabei das `XMLHttpRequest`–Objekt in JavaScript. Über dieses Objekt wird durch die GET- oder POST-Methode (Abschnitt 2.2.1) eine mit JavaScript dynamisch generierte Anfrage an den Server geschickt. Der Server verarbeitet diese Anfrage analog zu denen herkömmlicher Webanwendungen und sendet das Ergebnis der Anfrage in einem textbasierten Format (Text, HTML, JavaScript – oder eben XML) zurück.

2.3 Das Dienste-Paradigma im World Wide Web

Eine Implementierung des Konzepts der Service-orientierten Architektur (Abschnitt. 2.1.2.3) ist durch Web Services gegeben. Diese bieten konkrete Möglichkeiten über das WWW auf Anwendungen oder Daten in Form von Diensten zuzugreifen.

Nach dem W3C handelt es sich bei einem Web Service um (Austin et al. 2004)

> *„…a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols."*

oder im Hinblick auf speziellere Spezifikationen (Booth et al. 2004)

> *"…a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specificially WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*

Ein Web Service wird also durch einen URI identifiziert und ist darüber erreichbar, die Beschreibung des Service erfolgt durch XML, ebenso die Kommunikation zwischen verschiedenen Services.

Dieser Abschnitt fasst zunächst die Basiskomponenten des aktuellen State of the Art in der Umsetzung von Web Services zusammen. Anschließend werden Möglichkeiten aufgezeigt, über Web Services auf Daten, die für das Mapping on demand benötigt werden, zuzugreifen. Dafür werden zunächst Daten bzw. Inhalte allgemeiner Art, d.h. unterschiedlichster Thematiken und Erscheinungsform (z.B. Texte, Bilder) betrachtet. Der Bezug raumbezogener Daten wird durch die Beschreibung konkreter Standards vertieft.

2.3.1 Basiskomponenten einer Web-Service-Architektur

Die Komponenten bzw. Aktionen einer Service-orientierten Architektur (Abschnitt 2.1.2.3) werden in der Implementierung einer Web-Service-Architektur zurzeit durch die Spezifikationen von SOAP, WSDL und UDDI beschrieben (Dostal et al. 2005): SOAP definiert das Nachrichtenformat der Kommunikation, WSDL den Web Service selbst. UDDI beschreibt einen Verzeichnisdienst. Abbildung 2-7 zeigt die Komponenten und Aktionen des *Magischen Dreiecks* einer SOA aus Abbildung 2-5 erweitert um die genannten Spezifikationen.

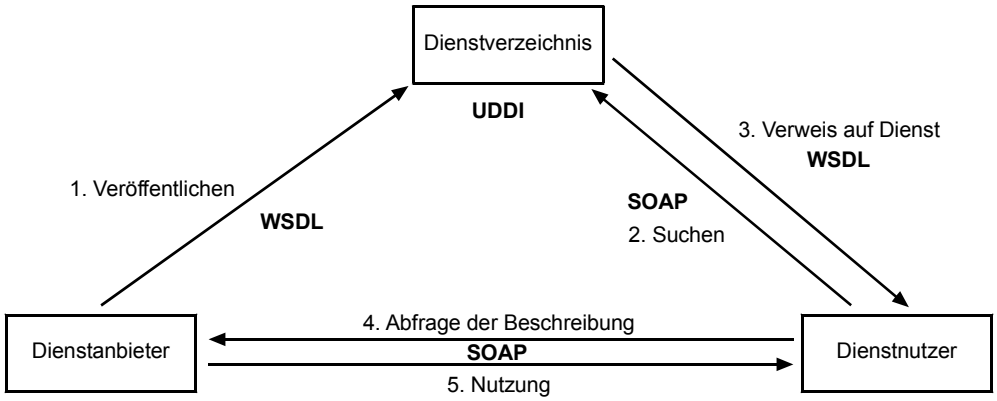


Abbildung 2-7: Magisches Dreieck einer Service-orientierten Architektur, ergänzt um konkrete Spezifikationen (nach Dostal et al. 2005)

2.3.1.1 SOAP

SOAP (früher für Simple Object Access Protocol, jetzt nicht mehr als Akronym genutzt (Gudgin et al. 2007a)) bezeichnet die Kommunikationskomponente zwischen Web Services. Es definiert die Struktur von Nachrichten, die zwischen einzelnen Diensten ausgetauscht

werden. SOAP ist eine XML-Anwendung, d.h. SOAP-Nachrichten werden als XML-Dokumente verschickt. Die eigentliche Nachrichtenübertragung erfolgt über geeignete Kommunikationsprotokolle, beispielsweise HTTP.

Eine SOAP-Nachricht besteht aus einem SOAP-Envelope, einem SOAP-Header und einem SOAP-Body. Der Envelope bildet die Klammer (den „Briefumschlag“) der SOAP-Nachricht. Der optionale SOAP-Header ist der erste Teil einer Nachricht; darin können Metainformationen verschickt werden. Der zweite Teil einer Nachricht, der SOAP-Body, enthält die eigentlich zu verschickenden Daten. Der Body ist für jede SOAP-Nachricht zwingend erforderlich.

SOAP ist als eine Empfehlung des W3C spezifiziert und besteht aus drei Teilen (Mitra & Lafon 2007, Gudgin et al. 2007a, Gudgin et al. 2007b). Aktuell (Januar 2008) ist die Version 1.2 in der zweiten Auflage verfügbar.

2.3.1.2 Web Services Description Language

Die Web Services Description Language (WSDL) stellt für die Beschreibung von Schnittstellen und Diensten in einer Service-orientierten Architektur ein Modell und ein entsprechendes XML-Format zur Verfügung. Die Beschreibung erfolgt dabei auf zwei Ebenen: Die abstrakte Ebene beschreibt die Funktionalität des Services, während die konkrete Ebene technische Details über den Ort und die Zugriffsmöglichkeit angibt.

Wichtigste Elemente von WSDL sind *types*, *interface*, *binding* und *service*. Types ermöglicht die Beschreibung von Nachrichten, die ein Dienst empfängt oder sendet. Die Angabe kann entweder durch direkte Einbindung eines XML-Schema-Dokuments (inline) oder durch Import eines externen XML-Schemas erfolgen. Im Element Interface wird die abstrakte Funktionalität eines Dienstes durch eine Menge von Operationen, über die ein Aufrufer mit einem Service interagieren kann, beschrieben. Für jede Operation wird festgelegt, welche Nachrichten ein Service darüber empfangen oder senden darf. Binding gibt die technischen Details, die für den Aufruf eines Service benötigt werden, an. Dazu gehören unter anderem das genutzte Nachrichten- und Transportprotokoll (z.B. SOAP bzw. HTTP). Die Angabe kann für jede der unter Interface angegebenen Operationen erfolgen. Das Service-Element fasst eine Menge von Endpunkten (endpoints) zusammen. Jeder Endpunkt gibt einen physikalischen Ort an, an dem ein Dienst zur Verfügung steht. Die Endpoint-Komponente in WSDL muss unter anderem die Referenz auf eine binding-Komponente und eine physikalische Adresse, meist einen URI angeben.

WSDL wird ebenfalls vom W3C standardisiert und liegt aktuell (Januar 2008) in der Version 2.0 als Recommendation vor. Genau wie SOAP ist WSDL in drei Dokumente aufgeteilt (Booth & Liu 2007, Chinnici et al. 2007b, Chinnici et al. 2007a).

2.3.1.3 Universal Description, Discovery und Integration

Dritte Basiskomponente einer Service-orientierten Architektur ist ein Verzeichnisdienst oder Registry. In einem Verzeichnisdienst werden Web Services durch die Hinterlegung von Metadaten veröffentlicht und so auffindbar gemacht. Für einen potentiellen Nutzer sind weiterhin Schnittstellenbeschreibungen der Dienste in Form von WSDL-Dokumenten verfügbar. Eine

Möglichkeit der Umsetzung eines solchen Verzeichnisdienstes ist die des Universal Description, Discovery und Integration-Verzeichnisses (UDDI-Verzeichnis)

Dostal et al. (2005) vergleichen die verschiedenen Arten der Datenbeschreibung in UDDI mit Telefonbüchern. Im Einzelnen wird dabei unterschieden:

- *White Pages* beinhalten Informationen über Akteure, die Web Services in einem UDDI-Verzeichnisdienst bereitstellen. Der Zugang für einen potentiellen Service-Nutzer erfolgt über den Namen eines Service-Anbieters.
- *Yellow Pages* gruppieren, ähnlich wie die bekannten „Gelben Seiten" für Dienstleistungen im Alltag, Dienste in einem Branchenverzeichnis. Der Zugang für einen potentiellen Nutzer erfolgt so über eine Branche.
- *Green Pages* enthalten Beschreibungen zu jedem Dienst; der Zugang für einen Nutzer erfolgt über Durchsuchen dieser Beschreibungen.
- Die *Service Type Registration* stellt die Informationen der Green Pages, die den menschlichen Nutzer adressieren, in maschinenlesbarer Form bereit.

Die Spezifikation von UDDI obliegt der Organization for the Advancement of Structured Information Standards (OASIS[5]). Aktuell (Januar 2008) ist die Version 3.0.2 des UDDI-Standards verfügbar (Clement et al. 2008).

2.3.2 Zugriff auf Inhalte

Das WWW bietet Zugriff auf eine Vielzahl von Daten in unterschiedlichsten Erscheinungs- und Ausdrucksformen, beispielsweise Texte, Bilder, Graphiken, Videos oder Animationen. Diese Inhalte genügen in ihrer Vielfältigkeit keiner allgemeinen Kategorisierung, die – zumindest zum jetzigen Zeitpunkt – einen einheitlichen Web Service für die Verfügbarmachung jeglicher Inhalte ermöglichen würde. Web Services, die nach einem weltweit gültigen Standard die Abfrage ganz bestimmter Informationen erlauben, sind bisher lediglich für bestimmte Domänen, beispielsweise im Kontext der raumbezogenen Daten (Abschnitt 2.3.3) verfügbar.

Für die Zwecke dieser Arbeit sind solche Daten von Bedeutung, die einen weltweit eindeutigen Raumbezug, beispielsweise durch die Angabe einer Koordinate in einem geeigneten Referenzsystem besitzen; weitergehend wird zwischen zwei Arten von Daten unterschieden:

- Inhalte oder Daten, die primär durch ihren informierenden Charakter geprägt sind. Sie beschreiben oder zeigen ein Objekt oder einen Sachverhalt, die Verortung im Raum ist von sekundärer Bedeutung. Diese Inhalte werden im Folgenden als Content bzw. Contents bezeichnet, im Falle einer Verortung als georeferenzierter Content.

[5] OASIS ist ein nicht-kommerzielles Industriekonsortium, das u.a. offene Standards für Web Services entwickelt (http://www.oasis-open.org, zuletzt geprüft am 01.04.2008).

- Inhalte, deren primäre Information ihre Lage im Raum ist und die zuallererst der Darstellung in Karten dienen (Abschnitt 2.3.3). Diese Daten werden allgemein als raumbezogene Daten bezeichnet.

Diese Differenzierung der Inhalte trägt zum einen der Realität der Datenbereitstellung im WWW durch eine Vielzahl von Akteuren Rechnung, zum anderen der Nutzung spezialisierter Werkzeuge:

- *Akteure:* Vergleicht man die Anbieter dieser beiden Arten von Daten, wird deutlich, dass erstere ein deutlich größere Zahl besitzen. Während jedermann durch einfachste Möglichkeiten (siehe unten) Anbieter von Contents werden kann, ist das Angebot an raumbezogenen Daten auf eine geringe Anzahl von Anbietern, meist Firmen oder öffentliche Institutionen, beschränkt.
- *Werkzeuge:* Für eine professionelle Erfassung, Verwaltung und Vorhaltung der Daten werden jeweils geeignete Werkzeuge benötigt. Für raumbezogene Daten ist dies ein Geoinformationssystem, für Contents ein Content Management System (Abschnitt 2.3.2.2).

In den nächsten Abschnitten erfolgt eine vertiefende Betrachtung bedeutsamer Schnittstellen und Werkzeuge.

2.3.2.1 Contents und ihre Schnittstellen

Wie bereits angedeutet, sind Contents in ihrer Gesamtheit durch sehr starke Heterogenität gekennzeichnet und nicht allgemeingültig durch einheitliche Standards beschrieben. Verfügbare Schnittstellen werden von einzelnen Anbietern bereitgestellt, die darüber erhältlichen Daten entsprechen dabei dem jeweiligen Geschäftsmodell und –zweck des Anbieters. Die Umsetzung der Schnittstelle fügt sich dabei entweder in die oben beschriebenen Basiskonzepte einer Service-orientierten Architektur oder setzt ein proprietäres Konzept um, das als API dokumentiert und für Interessenten veröffentlicht ist (z.B. die APIs der Firma Google[6]).

Als Beispiele für Schnittstellen, die Content zur Verfügung stellen, sind zu nennen:

- Der Online-Versandhändler Amazon[7] bietet Schnittstellen, die es beispielsweise ermöglichen, Produktangebote in die Webseite eines Dritten einzubinden. Weiterhin ermöglicht Amazon u.a. die Recherche in seiner Musik- und Buchdatenbank oder eine Schnittstelle zu einer allgemeinen Suchmaschine.
- Videoplattformen (Youtube[8], MyVideo[9]) gestatten es jedem Nutzer, Videos auf seinen Servern abzulegen. Diese Videos können dann von jedermann in die eigene Webpräsenz eingebunden werden.

[6] http://code.google.com/more (Zuletzt geprüft am 03.04.2008)

[7] http://www.amazon.de (Zuletzt geprüft am 17.11.2008)

[8] http://www.youtube.com (Zuletzt geprüft am 17.11.2008)

- Digitale Fotoalben (Flickr[10], Panoramio[11]) stellen analog zu den genannten Videoplattformen Bereiche zur Ablage von Fotos zur Verfügung. Diese Fotos sind ebenfalls über Schnittstellen erhältlich.

Für georeferenzierte Contents gilt ebenso das oben gesagte. Beispiele für solche Contents bieten ebenfalls Video- und Fotoplattformen: Die Videoplattform YouTube ermöglicht es einem Nutzer beim Hochladen eines Videos dieses in einer Karte zu verorten, nach dem gleichen Prinzip gestatten Flickr und Panoramio die Verortung von Fotos. Georeferenzierte Bilder in Panoramio, die gewissen Kriterien genügen, sind so beispielsweise automatisch über ihren Ort in GoogleEarth[12] abrufbar. Die Georeferenzierung von Contents wird im WWW auch als *Geotagging* bezeichnet (Wartala 2007).

Für bestimmte Zwecke oder Bereiche lassen sich allerdings auch eigene Content-Plattformen einrichten (vgl. Ausführungen Abschnitt 3.2 mit Content für touristische Zwecke). Die Hürden für den Aufbau solcher Plattformen sind durch eine Vielzahl leistungsfähiger Werkzeuge – Content Management Systeme – vergleichsweise gering.

2.3.2.2 Content Management Systeme

Die Vorhaltung und Nutzung eines großen Datenangebots erfordert einen professionellen Umgang mit diesen Daten, ein *Content Management*. Dies bezeichnet einen Prozess, der die Erstellung, Verwaltung und kontrollierte Veröffentlichung von Inhalten umfasst (Nix 2005). *Content Management Systeme* (CMS) sind Werkzeuge, die diesen Prozess unterstützen (ebd.). *Web Content Management Systeme* (WCMS) sind spezialisiert auf die Verwaltung und Veröffentlichung WWW-bezogener Contents im Rahmen einer Website, dem steht aber auch die Nutzung als Content-Lieferant über offene Schnittstellen nicht entgegen. Da CMS erst mit der Verbreitung des WWW und der Nutzung der Systeme für die Erstellung großer Webseiten populär wurden, werden die Begriffe „Content Management System“ und „Web Content Management System“ meist synonym gebraucht (Rockley 2003).

Als wesentliche Funktionen, die ein Content Management unterstützen, lassen sich beispielsweise nennen (vgl. Rockley 2003, Popper 2008):

- *Trennung von Inhalt und Layout*: Alle Inhalte werden unabhängig vom Design gespeichert und können so nicht nur in einem einheitlichen Layout ausgegeben werden, sondern es ist auch eine Überführung in ein anderes Layout problemlos möglich.
- *Speicherung der Inhalte in einer Datenbank*: Alle Inhalte werden zentral in einer Datenbank abgelegt und stehen nicht nur im System für verschiedene Ausgabemedien zur Verfügung, sondern lassen sich auch aus anderen Anwendungen bzw. von anderen

[9] http://www.myvideo.de (Zuletzt geprüft am 17.11.2008)

[10] http://www.flickr.com (Zuletzt geprüft am 17.11.2008)

[11] http://www.panoramio.com (Zuletzt geprüft am 17.11.2008)

[12] http://earth.google.de/ (Zuletzt geprüft am 23.09.2008)

Servern abfragen. Insbesondere muss die Ausgabe nicht zwingend in darstellbaren Formaten ((X)HTML) erfolgen, sondern ist auch durch Austauschformate über Web Services möglich.

- *Dezentrale Eingabe und Bearbeitung von Inhalten*: Gängige Systeme stellen keine hohen Softwareanforderungen auf Seiten der Lieferanten von Content. Der Zugang über einen aktuellen Browser genügt in der Regel, um alle Funktionen der Inhaltsbearbeitung nutzen zu können. In vielen Fällen genügt auch für weitreichende Administrationsaufgaben bereits ein Browser.
- *WYSIWYG-Editor*: Für die Eingabe von Inhalten steht meist ein WYSIWYG (What You See Is What You Get)-Editor – eine fortgeschrittene Online-Textverarbeitung – mit weit reichenden Eingabe- und Formatierungsmöglichkeiten zur Verfügung. Diese Möglichkeiten lassen sich auf die Anforderungen der Trennung von Inhalt und Layout anpassen.
- *Vergabe von Rollen und Zugriffsbeschränkungen*: Die Systeme erlauben die Einrichtung unterschiedlichster Rollen (z.B. Redakteure und Chefredakteure) mit sehr detaillierter Vergabe von Rechten für Funktionen und die Eingabe bzw. Bearbeitung von Inhalten.
- *Logging*: Alle Aktivitäten der Ersteller von Content werden mitgeloggt. Für jeden Zustand des Systems liegt so immer eine aktuelle Historie vor.
- *Versionierung*: Durch das Logging aller Aktivitäten der Autoren ist die Historie der eingegebenen Inhalte ständig nachvollziehbar. Gleichzeitig lassen sich durch ein Rollback ältere Versionen wieder aufrufen.
- *Mehrsprachigkeit*: Die Systeme erlauben nicht nur die Eingabe mehrsprachiger Inhalte, sondern bieten beispielsweise auch die Wahl der Sprache für jeden Autor.
- *Unterstützung von Workflows*: Inhalte können von der Eingabe bis zur Veröffentlichung festgelegte Routinen durchlaufen, beispielsweise erfolgt die Eingabe von News durch bestimmte News-Redakteure, die eigentliche Veröffentlichung obliegt einem übergeordneten Verantwortlichen, der einen Text freigeben oder zur Überarbeitung zurückgeben kann.
- *Korrektheit und Konsistenz*: Es ist jederzeit die Korrektheit und Konsistenz der Inhalte sichergestellt, beispielsweise das Vorhandensein eingebundener Bilder und die Gültigkeit von Hyperlinks innerhalb einer Webseite bzw. ins WWW.
- *Bildbearbeitung*: Autoren benötigen keine Bildbearbeitungskenntnisse, um vorhandene Bilder in ein erforderliches Graphikformat und die benötigte Größe zu ändern. Nach einem Upload erfolgt dies automatisch durch das System.
- *Daten- und Mediamanagement*: Der Umgang mit externen Formaten (Bilder, Dokumente) wird durch umfangreiche Management-Funktionalitäten unterstützt.

- *Suchmaschine*: Alle im System eingegebenen Inhalte sind direkt über eine integrierte Suchmaschine auffindbar.

Auf dem Markt ist heute eine kaum überschaubare Anzahl von CMS vertreten; diese reichen von kleineren Systemen, die auch einem Privatanwender die professionelle Erstellung seiner Webseiten erlauben, bis hin zu Systemen für den Einsatz in großen Unternehmen. Darunter befinden sich sowohl proprietäre als auch Open Source Produkte. Ein umfangreicher Vergleich von Funktionen findet sich beispielsweise unter http://www.cmsmatrix.org (Zuletzt geprüft am 07.04.2008).

2.3.3 Raumbezogene Dienste als Kern einer Geodateninfrastruktur

Das Mapping on demand erfordert die Abfrage und graphische Darstellung raumbezogener Daten. Dienste, die Funktionalität für diese Aufgaben bereitstellen, sind als Standards des Open Geospatial Consortiums (OGC[13]) verfügbar. Im Kontext der vorliegenden Arbeit sind vor allem die Standards aus dem Bereich des sogenannten Portrayals von Bedeutung.

2.3.3.1 Das Portrayal

Als Portrayal wird im Kontext des OGC die graphische Präsentation von raumbezogenen Informationen an menschliche Adressaten bezeichnet (Percivall 2003). Eine typische Darstellungsform ist die Visualisierung raumbezogener Daten in Form von Karten. Abbildung 2-8 zeigt das Modell des Portrayals, die Leserichtung ist dabei von unten nach oben.

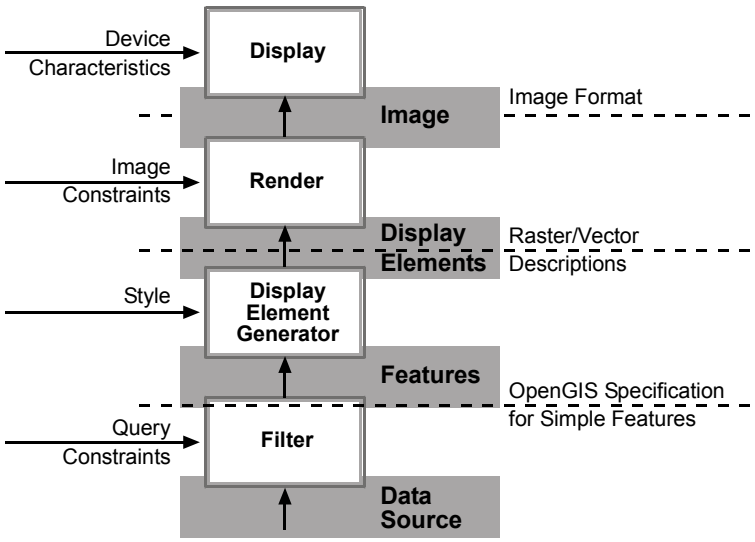


Abbildung 2-8: OGC-Modell des Portrayals (nach Percivall 2003)

[13] Das OGC ist ein internationales Industriekonsortium zur Erarbeitung offener Standards für den Austausch und die Verarbeitung raumbezogener Daten (http://www.opengeospatial.org, zuletzt geprüft am 03.04.2008).

Ausgangspunkt des Portrayals ist eine Datenquelle (Data Source), aus der durch eine Anfrage die darzustellenden Daten selektiert werden, Ergebnis sind raumbezogene Objekte (Features). In einem nächsten Schritt (Display Element Generator) werden durch eine Zuordnung von Style-Informationen graphische Repräsentationen (Display Elements) erhalten. Durch ein Rendering werden die Display Elemente als Bild gezeichnet und können dann auf einem Anzeigegerät dargestellt werden.

Im Folgenden werden vier Standards des OGC vorgestellt, die derzeit die beschriebene Abfolge des Portrayals am ehesten umsetzen:

- Der *Web Map Service* (WMS, Abschnitt 2.3.3.2) und das *Styled Layer Descriptor Profile für den WMS* (Abschnitt 2.3.3.3) als Schnittstellen zur Visualisierung raumbezogener Daten,
- das *Symbology Encoding* (SE, Abschnitt 2.3.3.4) zur Festlegung von Style-Informationen für raumbezogene Objekte,
- das *Filter Encoding* zur Selektion raumbezogener Objekte aus einer Datenquelle (Abschnitt 2.3.3.5).

Die Zusammenfassung von Diensten des OGC in einer Geodateninfrastruktur wird im Abschnitt 2.3.3.6 beschrieben.

2.3.3.2 Web Map Service

Der Web Map Service (WMS) bezeichnet den OGC-Dienst zur Visualisierung raumbezogener Daten. Ein WMS erzeugt auf Anfrage eines Clients aus raumbezogenen Daten eine georeferenzierte Karte in Form einer Bilddatei. Im überwiegenden Teil der Fälle werden diese Bilder als Rastergraphiken (z.B. *.jpg, *.gif, *.png) zurückgegeben, Vektorgraphiken (z.B. SVG) sind ebenfalls möglich, allerdings nicht sehr weit verbreitet. Die „OpenGIS Web Map Server Implementation Specification“ liegt aktuell in der Version 1.3.0 vor (de La Beaujardiere 2006).

Die Abfrage eines WMS erfolgt über das HTTP-Protokoll (Abschnitt 2.2.1). Für einen standardkonformen Service ist die Unterstützung der GET-Methode zwingend, die Unterstützung der POST-Methode optional. Der Service wird durch eine URL adressiert, bestimmte Operationen und damit die abzufragende Datenquelle, bzw. Ausschnitte daraus, durch Parameter-Wert-Paare (`key=value`) im Query-Teil der URL (Abschnitt 2.2.1). Einzelne Parameter werden dabei durch „&“ getrennt. Der WMS-Standard definiert die möglichen Parameter und damit die Regeln zur Konstruktion des Query-Teils. Die Beschreibung der konkreten Syntax erfolgt weiter unten in diesem Abschnitt.

Der WMS fügt sich damit nicht in das oben beschriebene Standardmodell einer Service-orientierten Architektur ein, insbesondere werden WSDL und SOAP nicht unterstützt. Eine Anfrage, die WMS-Spezifikation dahingehend zu ändern, liegt derzeit als *Discussion Paper* vor (Duschene & Sonnet 2005). Zudem ist es Beschluss, dass in Zukunft alle Web Services des OGC SOAP und WSDL unterstützen (OGC 2006).

Für die Nutzung eines WMS sind drei Operationen definiert. Neben `GetMap` zur Abfrage der Kartenbilder sind dies `GetCapabilities` und `GetFeatureInfo`. Während `GetCapabilities` und `GetMap` von einem standardkonformen WMS implementiert werden müssen, ist `GetFeatureInfo` optional.

GetCapabilities

Auf die Anfrage der GetCapabilities-Operation antwortet ein WMS mit einer Selbstauskunft in Form von Metadaten. Eine URL für einen Aufruf der GET-Methode hat dabei z.B. folgende Syntax:

```
http://www.example.net:8080/PfadZurKarte?
SERVICE=WMS&
REQUEST=GetCapabilities
```

Der Query-Teil der URL muss bei einem standardkonformen WMS die Parameter `SERVICE` zur Bezeichnung der Art des angefragten Dienstes und `REQUEST` zur Bezeichnung der angefragten Operation enthalten. Eine Übersicht über alle möglichen Parameter findet sich im Anhang A.1.1.

Der WMS sollte mit einer maschinen- und menschenlesbaren Beschreibung (XML) des vorliegenden Services antworten. Darin enthalten sind neben allgemeinen Informationen (z.B. Name und Titel des Services, Schlüsselwörter und Kontaktdaten des Anbieters) insbesondere Informationen über die verfügbaren raumbezogenen Daten (z.B. Kartenthemen, Raumausschnitte und Referenzsysteme). Die Metadaten müssen es einem Client ermöglichen, Kartenanfragen (GetMap) zu formulieren.

GetMap

Eine gültige[14] GetMap-Anfrage beantwortet ein WMS mit einem Kartenbild. Die Werte der im Query-Teil enthaltenen Parameter sind dabei gemäß der Metadaten einer GetCapabilities-Anfrage formuliert. Ein Beispiel für eine GetMap-Anfrage ist:

```
http://www.example.net:8080/PfadZurKarte?
VERSION=1.3.0&
REQUEST=GetMap&
CRS=CRS:84&
BBOX=-95.15,21.534,-77.987,45.712&
WIDTH=540&
HEIGHT=320&
LAYERS=Layer1&
STYLES=&
FORMAT=image/png
```

[14] Unter einer gültigen GetMap-Anfrage wird hier eine Anfrage verstanden, deren Syntax korrekt ist und die nur solche Werte für die Parameter verwendet, die den Capabilities entsprechen.

Die Anfrage enthält neben der Version des Standards (`VERSION`), nach dem die Anfrage formuliert wurde, die Art der Anfrage (`REQUEST`), das gewünschte Referenzsystem (`CRS`), die linke untere und rechte obere Ecke der angefragten Karte in Koordinaten des Referenzsystems (`BBOX`), die Breite und Höhe des Kartenbildes in Pixeln (`WIDTH, HEIGHT`), die gewünschten thematischen Ebenen (`LAYERS`), zugehörige Style-Informationen (`STYLES`) und das gewünschte Graphikformat (`FORMAT`). Eine Übersicht über alle Parameter, die eine GetMap-Anfrage enthalten muss bzw. darf, findet sich im Anhang A.1.2.

GetFeatureInfo

Hat eine GetMap-Operation ein Kartenbild zurückgeliefert, lassen sich nähere Informationen über einzelne Kartenobjekte durch eine GetFeatureInfo-Operation abfragen. Die Identifikation des betreffenden Objekts erfolgt dabei über die Auswahl von Pixelkoordinaten im System des Kartenbildes. Da sowohl ein WMS als auch das genutzte HTTP-Protokoll zustandslos sind – der WMS kann also eine GetFeatureInfo-Anfrage einer vorangegangenen GetMap-Anfrage nicht mehr zuordnen – muss eine GetFeatureInfo-Anfrage die Parameter zur eindeutigen Identifikation eines Kartenbildes (`BBOX, CRS`, `WIDTH` und `HEIGHT`) erneut mitführen. Alle möglichen Parameter einer GetFeatureInfo-Anfrage sind im Anhang A.1.3 aufgeführt.

Die Antwort kann beispielsweise in Form von GML (Geography Markup Language), einer XML-Anwendung des OGC zur Beschreibung räumlicher Objekte, zurückgegeben werden.

2.3.3.3 Styled Layer Descriptor Profile für den WMS

Ein WMS bietet nur rudimentäre Möglichkeiten, die ausgegebenen Kartenbilder in ihrer graphischen Darstellung zu beeinflussen. Wie im vorangegangenen Abschnitt deutlich wurde, können über den Parameter `STYLES` lediglich Layoutanweisungen angefordert werden, die vom WMS selbst zur Verfügung gestellt werden. Damit bleibt es dem jeweiligen Betreiber vorbehalten, die Darstellungsweise seiner Daten festzulegen. Hinzu kommt, dass sich so nur die Erscheinungsweise ganzer Datensätze bzw. Layer bestimmen lässt.

Die *Styled Layer Descriptor Profile of the Web Map Service Implementation Specification* (in dieser Arbeit kurz „SLD Profile“) beschreibt die Erweiterung der WMS-Schnittstelle um die Möglichkeit, beliebige Darstellungsanweisungen von Seiten eines Nutzers an den WMS zu übergeben. Die Darstellungsanweisungen liegen dabei als Symbology Encoding (SE, Abschnitt 2.3.3.4) vor. Ein WMS, der dem SLD Profile Standard entspricht (SLD-WMS), muss dabei nicht zwingend von Seiten des Anbieters fest mit Daten verknüpft sein, sondern kann optional als „Portrayal Engine“ für fremde, in einer Anfrage angegebene Datenquellen dienen (s.u.).

SLD Profile ist eine Spezifikation des OGC und liegt aktuell in der Version 1.1.0 vor (Lupp 2007). SLD Profile und SE sind aus einem gemeinsamen Standard, der *Styled Layer Descriptor Implementation Specification* hervorgegangen (Lalonde 2002). Letzterer ist mittlerweile als „deprecated“ eingestuft, d.h. er sollte nicht mehr verwendet werden.

Durch einen SLD-WMS werden einzelne Operationen der im vorangehenden Abschnitt 2.3.3.2 beschriebenen Spezifikation erweitert bzw. hinzugefügt. Ein SLD-WMS muss zwingend die Operationen `GetCapabilities` und `GetMap` implementieren, `DescribeLayer` und `GetLegendGraphic` sind optional.

GetCapabilities

Der Aufruf von `GetCapabilities` erfolgt in gleicher Weise wie im Abschnitt 2.3.3.2 beschrieben. Ein SLD-WMS muss seine Antwort allerdings um Informationen über seine SLD-Funktionalität erweitern.

GetMap

Die GetMap-Operation des SLD Profile erweitert die gleichnamige Operation des WMS um fünf Parameter, eine Übersicht ist im Anhang A.2.1 enthalten.

Die Einbindung von Style-Anweisungen erfolgt entweder über den Parameter `SLD`, der auf die URL einer beliebig im WWW abgelegten Datei mit Darstellungsvorschriften verweist, oder über den Parameter `SLD_BODY`, der Darstellungsvorschriften direkt als Zeichenkette in der URL einer GetMap-Anfrage erwartet. Die Parameter `REMOTE_OWS_TYPE` und `REMOTE_OWS_URL` erlauben die Spezifizierung der angesprochenen fremden Datenquelle in Form von Web Services; der Typ dieser Services muss dabei entweder raumbezogene Objekte (Features) über einen Web Feature Service (WFS, Vretanos 2005b) oder feldbasierte Daten (Coverages, ISO-Standard 2005) über einen Web Coverage Service (WCS, Whiteside & Evans 2008) zurückliefern[15]. Ein Beispiel für eine GetMap-Anfrage mit Angabe von Style-Anweisungen in einer externen Datei ist:

```
http://www.example.net:8080/PfadZurKarte?
VERSION=1.3.0&
REQUEST=GetMap&
CRS=CRS:84&
BBOX=-95.15,21.534,-77.987,45.712
WIDTH=540&
HEIGHT=320&
SLD=http://example.com/PfadZumSLDDokument/sld.xml&
FORMAT=image/png
```

Anzumerken ist, dass in einer Anfrage mit `SLD` oder `SLD_BODY` die Parameter `LAYERS` und `STYLES` nicht mehr enthalten sind, da entsprechende Angaben durch die übergebenen Style-Anweisungen definiert werden.

[15] Vertiefende Ausführungen zur objekt- bzw. feldbasierten Modellierung finden sich bspw. in Worboys & Duckham (2004).

DescribeLayer

Die Formulierung von Style-Anweisungen durch den Nutzer eines SLD-WMS erfordert Informationen über die Art der darzustellenden räumlichen Objekte. Die DescribeLayer-Operation liefert Informationen darüber zurück, welche Klassen räumlicher Objekte verfügbar sind; die notwendigen Parameter sind im Anhang A.2.2 aufgeführt. Das Rückgabeformat einer DescribeLayer-Operation ist in jedem Fall XML.

Die Operation `GetLegendGraphic` liefert für übergebene Style-Informationen eine Legende als Bild zurück. Da diese Operation im Kontext dieser Arbeit von untergeordneter Bedeutung ist, wird auf die – recht umfangreiche – Angabe der Anfrageparameter verzichtet.

2.3.3.4 Symbology Encoding

Die *Symbology Encoding Implementation Specification* (SE) beschreibt eine XML-Anwendung zur Definition von Darstellungsanweisungen von Features (raumbezogene Objekte) und Coverages (feldbasierte Daten). Wie im vorangehenden Abschnitt beschrieben, ist diese Spezifikation gemeinsam mit dem SLD Profile aus der Styled Layer Descriptor Implementation Specification hervorgegangen. SE liegt als OpenGIS Implementation Specification in der Version 1.1.0 vor (Müller 2006).

Die Angabe von Style-Anweisungen erfolgt im Falle von Features für einzelne Objektklassen, im Falle von Coverages für spezifische Coverages. Die Syntax der Anweisungen ist durch das Element Rule und dessen Kindelemente definiert. Abbildung 2-9 gibt den betreffenden Ausschnitt aus dem XML-Schema für das Symbology Encoding wieder. Für jede Objektklasse bzw. jedes Coverage können dabei beliebig viele Rule-Elemente aneinandergereiht, und so sehr ausdifferenzierte Darstellungen erreicht werden.

```
<xsd:element name="Rule" type="se:RuleType">
</xsd:element>
<xsd:complexType name="RuleType">
  <xsd:sequence>
    ...
    <xsd:choice minOccurs="0">
      <xsd:element ref="ogc:Filter"/>
      <xsd:element ref="se:ElseFilter"/>
    </xsd:choice>
    <xsd:element ref="se:MinScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:MaxScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:Symbolizer" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Abbildung 2-9: XML-Schema für das Element "Rule" des Symbology Encodings (Quelle: Müller 2006)

Die Bedeutung der Elemente von Rule ist wie folgt: Die konkrete graphische Erscheinungsform von Features in einem Kartenbild wird durch `Symbolizer` beschrieben. Sofern mög-

lich, entsprechen Terminologie und Syntax dabei den Standards von SVG[16]/CSS2. Es werden fünf Arten von Symbolizern unterschieden

- *Point Symbolizer* ordnen einem punkthaften Objekt ein Symbol (Graphic) zu. Eine Graphic kann entweder durch SE konfiguriert werden (Mark) oder als externe Graphikdatei durch einen Verweis referenziert werden. Während es sich bei ersteren um einfache geometrische Figuren (Quadrate, Kreise, …) mit spezifischen Eigenschaften (Begrenzung, Füllung) handelt, können letztere beliebig komplex sein. Für das gewählte Symbol lassen sich weiterhin Eigenschaften wie Größe und Richtung angeben.
- *Line Symbolizer* werden genutzt, um linienhafte Geometrien darzustellen. Beeinflussbar sind beispielsweise Linieneigenschaften wie Farbe, Stärke und Strichlierungen. Durch die Angabe einer Graphic (Point Symbolizer) ist eine Linie durch beliebige Signaturen (Kap 4.2.5) darstellbar.
- *Polygon Symbolizer* visualisieren flächenhafte Geometrien. Beeinflussbar sind dabei die Begrenzungslinie und die Füllung. Festlegungen für die Linien sind durch den Line Symbolizer gegeben. Füllungen können einfarbig sein oder ebenfalls durch eine Graphic (Point Symbolizer) bestimmt werden.
- *Text Symbolizer* legen die Darstellung von Beschriftungen fest. Mögliche Parameter sind beispielsweise Schriftart, Schriftgröße oder die Platzierung der Schrift (vgl. auch Abschnitt 4.2.5).
- *Raster Symbolizer* bestimmen die Darstellung von Coverages.

Die Anwendung der `Symbolizer` kann sowohl auf ganze Datensätze bzw. Layer als auch in Abhängigkeit von bestimmten Bedingungen (Filter, siehe unten) erfolgen. Die Festlegung von Werten für die Elemente `MinScaleDenominator` und/oder `MaxScaleDenominator` bindet die Darstellung an Maßstabsgrenzen oder an einen Maßstabsbereich. Die Festlegung mehrerer Rules-Elemente erlaubt unterschiedliche Darstellungen in verschiedenen Maßstäben.

Symbolizer sind durch den Einsatz von Filtern ganz gezielt auf bestimmte Features anwendbar. Die Semantik eines Filters besteht in der Formulierung von boolschen Ausdrücken, die eine Bedingung definieren. Die Angabe der Bedingung erfolgt durch das Element `Filter`, die Angabe der Alternative durch das Element `ElseFilter`. Filter sind mit den oben genannten Maßstabsbedingungen kombinierbar; durch die Nutzung beliebig vieler `Rule`-Elemente können damit bestimmte Features in unterschiedlichen Maßstäben auf verschiedene Arten dargestellt werden.

Die Formulierung von Filtern ist Teil der *Filter Encoding Implementation Specification* im nächsten Abschnitt.

[16] SVG (Scalable Vector Graphics) ist eine Empfehlung des W3C und definiert ein Vektorgraphikformat.

2.3.3.5 Filter Encoding

Im vorangehenden Abschnitt wurde im Rahmen der SE-Spezifikation die Auswahl und Anzeige einzelner Features über Filter genannt. Die *Filter Encoding Implementation Specification* des OGC (Vretanos 2005a) beschreibt eine XML-Anwendung zur Formulierung dieser Filter-Ausdrücke. Die Einbindung der Elemente `Filter` und `ElseFilter` wurde bereits beschrieben; an dieser Stelle werden deshalb lediglich die in Filtern einsetzbaren Bedingungen zusammengefasst.

Die Spezifikation definiert drei Gruppen von Filtern: Spatial Operators, Comparison Operators und Logical Operators.

Spatial Operators testen, ob die Geometrie eines räumlichen Objekts eine Beziehung zu einer bestimmten anderen Geometrie besitzt. Mögliche Operatoren sind topologische Relationen (Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects und Contains)[17], die Lage innerhalb einer Bounding Box oder die Lage inner- oder außerhalb einer bestimmten Distanz.

Comparison Operators werden für die Formulierung mathematischer Beziehungen durch die Angabe vergleichender Operatoren (=, <, >, <=, >=, <>) genutzt. Die Anwendung der Operatoren erfolgt in Form von Zeichenketten (z.B. „PropertyIsEqualTo“ statt „=“). Über diese gängigen Operatoren hinaus sind die Elemente „PropertyIsLike“, „PropertyIsBetween“ und „PropertyIsNull“ verfügbar.

```
<Filter>
  <And>
    <PropertyIsLessThan>
      <PropertyName>DEPTH</PropertyName>
      <Literal>30</Literal>
    </PropertyIsLessThan>
    <Not>
      <Disjoint>
        <PropertyName>Geometry</PropertyName>
        <gml:Envelope
              srsName="http://www.opengis.net/gml/srs/epsg.xml#63266405">
          <gml:lowerCorner>13.0983 31.5899</gml:lowerCorner>
          <gml:upperCorner>35.5472 42.8143</gml:upperCorner>
        </gml:Envelope>
      </Disjoint>
    </Not>
  </And>
</Filter>
```

Abbildung 2-10: Beispiel des Filter Encodings für die Kombination eines Comparison Operators mit einem Spatial Operator (Quelle: Vretanos 2005a)

Die *Logical Operators* erlauben mit Hilfe der boolschen Operatoren „and“, „or“ und „not“ den Aufbau boolscher Audrücke, die Spatial Operators und Comparison Operators verknüpfen. Abbildung 2-10 zeigt als Beispiel die Verknüpfung des Comparison Operators „kleiner“

[17] Mehr zu topologischen Relationen bspw. in Egenhofer & Franzosa (1991).

und des Spatial Operators „disjoint“ – bzw. dessen Negierung – durch den Operator „and“: Es werden alle Features ausgewählt, deren „DEPTH“-Wert kleiners als 30 ist und die mit der angegebenen Bounding Box in einer räumlichen Beziehung stehen.

2.3.3.6 Geodateninfrastruktur

Web Map Services sind – neben anderen Diensten, die im Kontext dieser Arbeit nicht von Bedeutung sind – häufig Teil einer sogenannten Geodateninfrastruktur (GDI). Darunter werden allgemein alle institutionellen, organisatorischen, technologischen und menschlichen Ressourcen verstanden, die eine vernetzte Vorhaltung, Verwaltung und Verteilung von raumbezogenen Daten sowie den Zugriff darauf ermöglichen (vgl. Groot & McLaughlin 2000).

Donaubauer (2004) definiert aus einer eher technischen Sicht eine dienstorientierte Geodateninfrastruktur als ein Verteiltes System aus lose miteinander gekoppelten Komponenten, u.a. Anbietern, Vermittlern und Nutzern von Geoinformations-Ressourcen (Geodaten und Dienste für deren Nutzung), das Internet als Netzwerk, Benutzerschnittstellen und Geoportale (Kap. 3.1.2) als Einstiegsknoten in die GDI. Ziel einer dienstorientierten GDI ist es, durch den Einsatz und die Vernetzung standardisierter Geo Web Services „vorhandene, verteilte, heterogene GI-Ressourcen auf einfache, kontrollierte Weise nutzbar und kombinierbar zu machen, um so einer breiteren Nutzerschicht Zugang zu Geodaten und der GIS-Technologie zu verschaffen“ (ebd.).

Eine so definierte dienstorientierte GDI lässt also die einleitend genannten institutionellen, organisatorischen und menschlichen Ressourcen weitgehend außer Acht und fokussiert Komponenten und Technologien, die für den Bezug von Geodaten über das Internet bzw. WWW von Bedeutung sind. Damit wird deren Gesamtinfrastruktur auf die Nutzung für Zwecke raumbezogener Daten spezialisiert.

Im Kontext einer GDI wird häufig von einem Wertschöpfungsparadoxon gesprochen: Die Erhebung der in einer GDI verfügbaren Daten kostet sehr viel Geld, ihr Wert in einer GDI ist aber zunächst unbestimmt, da eine GDI ein allgemeines Angebot ohne konkreten Anwendungsbezug schafft (Fornefeld et al. 2004). Als Ausweg aus diesem Dilemma empfiehlt die Studie die „Veredelung“ der Daten durch den Aufbau von Mehrwertdiensten, d.h. mit oder auf Basis der Daten wird ein ganz konkreter Nutzen geboten, durch den Zahlungsbereitschaft durch die Anwender entsteht (ebd.).

Die Bedeutung der Geodateninfrastruktur für eine ad-hoc-Erstellung von Karten ist offensichtlich: Die GDI bildet eine Basis für den Bezug von Geodaten über das WWW und ist in erster Näherung vergleichbar mit dem oben genannten Geoportal. Dieser Aspekt des Einstiegsknotens in eine GDI – und allgemein in das WWW – wird im Kapitel 3 vertiefend ausgeführt.

2.4 Architekturen im World Wide Web

In den bisherigen Ausführungen dieses Kapitels wurden wesentliche Grundlagen des Internets betrachtet, angefangen mit dem Konzept der Verteilten Systeme und möglichen Architekturen

für deren Umsetzung, bis hin zu konkreten Technologien und Schnittstellen für Anwendungen des WWW.

Abschließend soll nun die Frage beantwortet werden, wie die vorgestellten Architekturkonzepte im WWW umgesetzt sind und wie sich die genannten Technologien integrieren. Dafür werden im Folgenden zwei typische Client-Server-Architekturen gegenübergestellt. Sie verdeutlichen beispielhaft die Möglichkeiten eines Anwenders, Daten aus dem WWW zu beziehen und spannen gleichzeitig einen Bogen von den Anfängen bis zum aktuellen Status quo des WWW. Basis der Architekturen ist in beiden Fällen das Client-Server-Modell.

2.4.1 Das „frühe" World Wide Web

Die ersten Jahre nach Erfindung des WWW waren durch Konstellationen gemäß Abbildung 2-11 geprägt: Client und Server bilden eine physische Zwei-Tier-Architektur, die funktionalen Elemente eine logische Drei-Tier-Architektur. Von den in den Abschnitten 2.2.2 und 2.2.3 beschriebenen Technologien dominierte HTML, alle anderen waren entweder noch nicht entwickelt oder steckten noch in den Kinderschuhen.

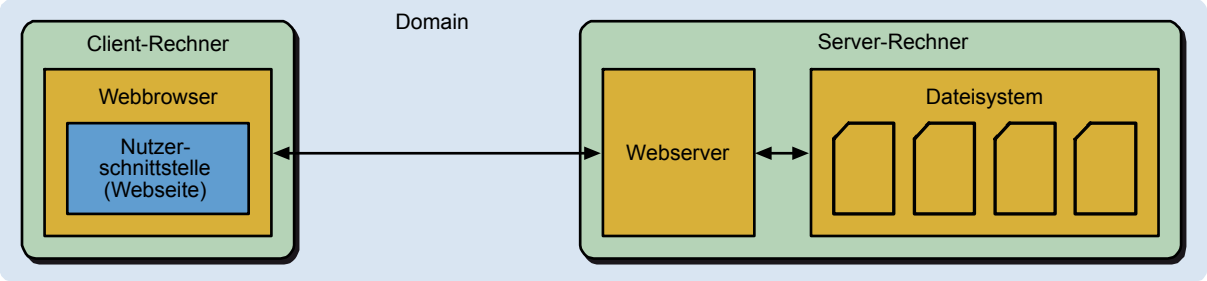


Abbildung 2-11: Modell einer Anwendung der ersten Generation von Webarchitektur

Die Client-Seite dieser Konstellation besteht aus einem Webbrowser; durch Eingabe einer bestimmten Internetadresse (URL, Abschnitt 2.2.1) werden Anfragen an den Webserver gerichtet, der unter dieser Adresse erreichbar ist. Die vom Server zurückgesendeten Daten werden im Browser als Webseite dargestellt. Im Kontext dieser Webseite kommuniziert der Client lediglich mit genau diesem Server, d.h. in der Domain des Servers.

Die auf dem Server verfügbaren Daten sind fest in HTML-Dateien eingebunden und im Dateisystem des Servers abgelegt. Die HTML-Dateien sind durch Hyperlinks miteinander vernetzt und bilden einen festen Rahmen; ein Nutzer bezieht Daten, indem er diesen Hyperlinks folgt.

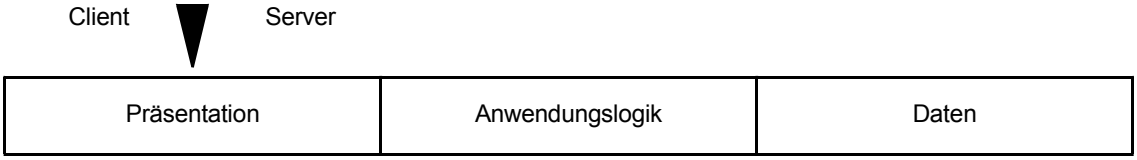


Abbildung 2-12: Aufteilung der Funktionalität zwischen Client und Server im frühen WWW (nach Shan & Earle 1998)

Die Aufteilung der Funktionalität zwischen Client und Server entspricht damit der verteilten Präsentation (Abschnitt 2.1.2.2): Datenhaltung (HTML-Dateien), Anwendungslogik (Webserver) und ein Teil der Präsentation erfolgen komplett auf dem Server-Rechner. Der Client-Rechner stellt lediglich die – in ihrem Layout schon festgelegten – HTML-Dokumente dar und ist damit als Thin-Client zu bezeichnen. Abbildung 2-12 fasst diese Aufteilung noch einmal zusammen.

Eine Verknüpfung mit Informationen aus anderen Quellen, d.h. von anderen Servern und damit aus anderen Domains, erfolgt ebenfalls durch Hyperlinks. Folgt ein Nutzer einem Link, wird eine völlig neue Webseite unter einer anderen Domain geöffnet, der Informationsbezug kann wiederum nur im Kontext dieser Domain erfolgen. Eine Verbindung zum Ausgangspunkt des Hyperlinks besteht nicht. Es wird so eine global vernetzte Struktur aus Informationen geschaffen, die es einem Nutzer ermöglicht, sich von Server zu Server zu bewegen, und die jeweiligen Inhalte abzufragen.

2.4.2 Service-orientierte Architektur im Web 2.0

Die Umsetzung einer Service-orientierten Architektur im WWW ist eng mit dem Begriff „Web 2.0“ verbunden, unter dem aktuell (Anfang 2008) der Status quo des WWW sowohl aus konzeptioneller als auch aus technischer Sicht subsumiert wird. Vor ca. drei Jahren zum ersten Mal verwendet, ist das Web 2.0 sehr schnell zu einem Trend geworden und stellt mittlerweile eines der am häufigsten gebrauchten Schlagworte im Umfeld des WWW dar.

Trotz dieser Verbreitung ist das Web 2.0 bis heute nicht eindeutig definierbar und im wissenschaftlichen Kontext nur schwer fassbar. Als Urheber des Begriffes gilt Tim O’Reilly (O'Reilly 2005), der darunter Beobachtungen im Anwendungs- und Anwenderverhalten im WWW zusammengefasst, und einen gewissen Trend erkannt hat. Beschreibungen des Web 2.0 (O'Reilly 2005; Behme 2007) lassen sich durch einige Schlagwörter kennzeichnen:

- *Teilnahme („User-generated Content“)*: Eine wachsende Zahl von Nutzern ist nicht mehr reiner Konsument von Inhalten, sondern stellt eigene Daten und Informationen auf Plattformen Dritter zur Verfügung.
- *Interaktivität und Kommunikation*: Nutzer des WWW kommunizieren über die Bereitstellung und Kommentierung bzw. Bewertung von Inhalten miteinander. Ein Beispiel ist das Blogging, das Erstellen einer persönlichen Webseite im Stil eines Tagesbuchs.
- *Konvergenz von Webanwendungen und Desktop-Programmen*: Webseiten treten nicht mehr zwingend als statisches Dokument in Erscheinung, sondern werden mehr und mehr zu interaktiv bedienbaren Anwendungen, die Erscheinungsbild und Funktionalität von Desktop-Programmen nachbilden.

Einer Service-orientierten Architektur im Web 2.0 liegt auch weiterhin eine logische Drei-Tier-Architektur zugrunde. Als physikalische Basis dient allerdings eine n-Tier-Architektur, typischerweise mit einer Verteilung auf Seiten der Datenhaltung. Dies bedeutet zunächst, dass die Daten nicht an einen Server-Rechner gebunden, sondern weltweit beliebig verteilt sein

können. Im Vergleich zu obigen Ausführungen zum „frühen“ Web sind sie damit auch nicht an einen statischen Rahmen von HTML-Dateien gebunden, sondern beispielsweise an eine bestimmte Thematik oder Anwendung (z.B. Fotoalben). Voraussetzung für den allgemeinen Zugriff auf diese Daten ist ihre Verfügbarkeit über veröffentlichte Schnittstellen in Form von Web Services.

Eine beliebige Anwendung, die Zugangsinformationen zu diesen Schnittstellen besitzt, hat die Möglichkeit, unterschiedlichste Daten verschiedener Quellen parallel abzufragen, in einem neuen Bezugsrahmen zu kombinieren, zu integrieren und einem Benutzer in geschlossener Form zu präsentieren. Für diese Art der Zusammenführung von Daten verschiedener Quellen hat sich im WWW der Begriff „Mashup“ eingebürgert (vgl. Behme 2007). Als Technologie wird in der Regel Ajax (Abschnitt 2.2.3) genutzt.

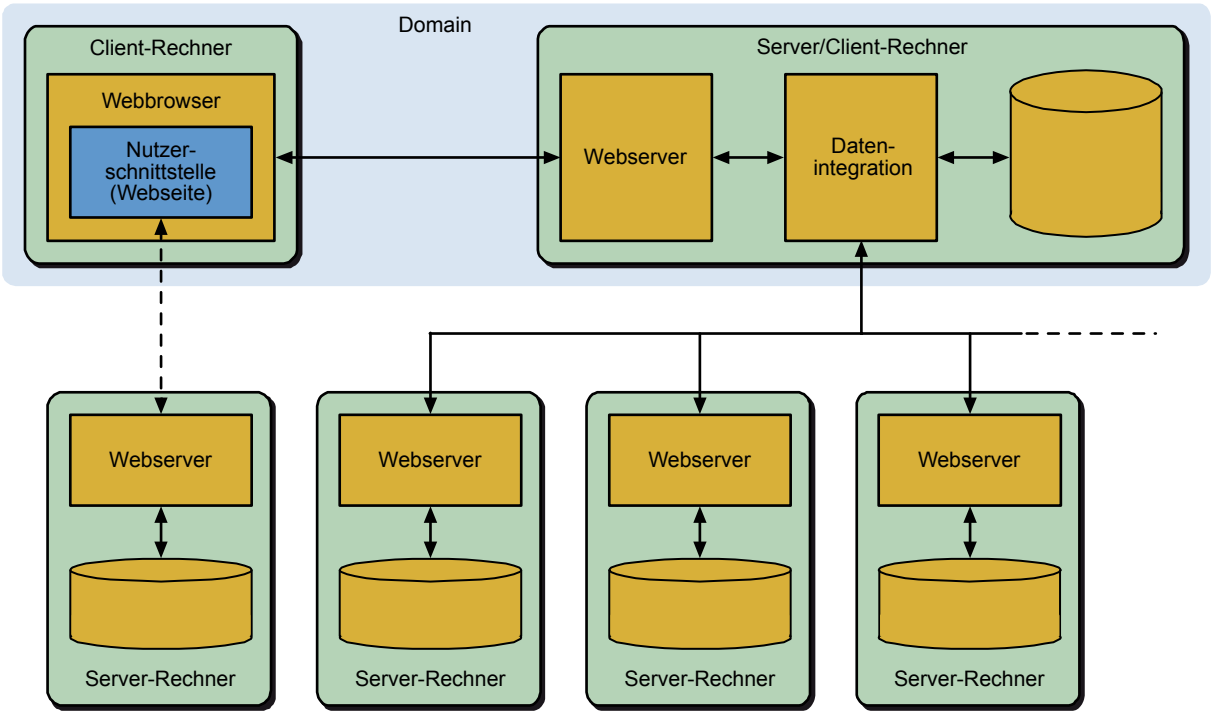


Abbildung 2-13: Modell einer Anwendung in der Service-orientierten Architektur des Web 2.0

Abbildung 2-13 zeigt die typische Architektur einer Anwendung des Web 2.0. Aus Sicht des Nutzers eines Client-Rechners ändert sich im Vergleich zur Architektur des „frühen“ WWW zunächst nichts. Der Client kommuniziert weiterhin mit genau einem Server-Rechner (innerhalb der Domain des Server-Rechners) und bezieht von dort eine mit Ajax umgesetzte Benutzeroberfläche. Der Server-Rechner wird allerdings seinerseits zum Client und bezieht Inhalte von verschiedenen anderen Servern, diese Inhalte werden durch eine erweiterte Anwendungslogik (*Datenintegration*) kombiniert und integriert. Bei einer fortgesetzten asynchronen Kommunikation kann diese Datenintegration dynamisch auf Aktionen eines Nutzers hin erfolgen. Der „Umweg“ der Datenabfragen über den Server/Client-Rechner ist aus Sicherheitsgründen in jedem Fall erforderlich, die Abfrage und Integration beliebiger Server mit Ajax direkt aus dem Client heraus wäre ein Cross-Domain-Zugriff und ist nicht möglich (vgl. Aus-

führungen zu JavaScript, Abschnitt 2.2.3). Eine Ausnahme, d.h. eine Integration von Daten, die direkt von außerhalb der Domain bezogen werden, besteht allerdings bei der Einbindung von Elementen, die sich durch Angabe ihrer Quelle referenzieren lassen (z.B. Bilder). Dies ist in Abbildung 2-13 durch die gestrichelten Pfeile angedeutet.

Die Aufteilung der Funktionalität verschiebt sich gemäß Abbildung 2-14 in einer service-orientierten Architektur in Richtung der Anwendungslogik. Die genaue Spezifizierung der Aufteilung hängt dabei immer vom Einzelfall der Anwendung ab, u.a. von den Einflussmöglichkeiten eines Nutzers auf die Auswahl der dargestellten Inhalte.

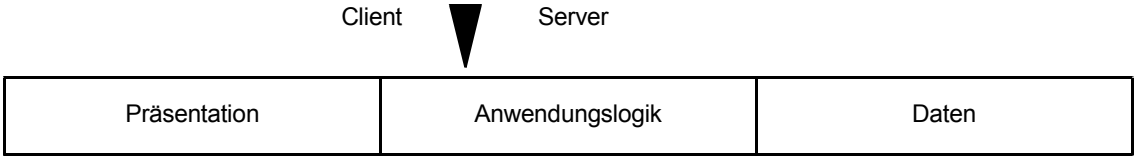


Abbildung 2-14: Aufteilung der Funktionalität zwischen Client und Server in einer Service-orientierten Architektur

2.5 Zusammenfassung

In diesem Kapitel wurde verdeutlicht, dass das Internet und seine Technologien das Potential für die Abfrage, physikalische Zusammenführung und Integration weltweit verteilter heterogener Daten besitzen.

Das Internet selbst bietet die geeignete Infrastruktur, um weltweit auf Daten zuzugreifen und auf dem Rechner eines Anwenders zusammenzuführen. Die Heterogenität von Daten und Systemen wird durch Standards auf verschiedenen Ebenen überwunden. Diese Standards, die von nicht-kommerziellen internationalen Konsortien erarbeitet werden, vereinheitlichen die Kommunikation von Systemen und die Bereitstellung von Daten durch

- eine weltweit eindeutige Bezeichnung und Lokalisierung von Datenquellen (URI),
- geeignete Transportprotokolle und Zugriffsmethoden (HTTP),
- Datenformate zur Speicherung und zum Austausch von Informationen, die sowohl die Semantik (XML) als auch die Darstellung und Präsentation ((X)HTML, CSS) von Daten unterstützen,
- Technologien zur Umsetzung von Interaktion und zur Verarbeitung der Datenformate (JavaScript, DHTML, Ajax),
- Technologien zur Umsetzung einer Web-Service-Architektur (SOAP, WSDL, UDDI),
- konkrete Definitionen von Web Services, die die Abfrage und Darstellung raumbezogener Daten ermöglichen (WMS, SLD-WMS mit den benachbarten Standards des Symbology Encodings und Filter Encodings).

Der Einsatz der genannten Technologien ermöglicht neben dem Bezug von Daten für ein Mapping on Demand auch die Entwicklung von Anwendungen, die die Nutzung der Infra-

struktur des WWW und die Erstellung von ad-hoc-Karten durch komfortable Benutzeroberflächen unterstützen. Die Standardisierung der Technologien sichert eine größtmögliche Verbreitung. Eine (ad-hoc-)Nutzbarkeit wird dadurch erreicht, dass keine besondere Hard- oder Software benötigt wird, sondern zur Nutzung bzw. Darstellung der Technologien ein gängiger Webbrowser genügt.

3 Bündelung von Diensten in Portalen

Im letzten Kapitel wurde die Verfügbarkeit von Technologien gezeigt, die den Bezug und die Integration von Daten aus dem WWW ermöglichen. Ebenfalls verfügbar sind eine Vielzahl geeigneter Datenquellen, die diese Technologien – insbesondere die WMS-Schnittstelle – anwenden und von ihren Anbietern zur allgemeinen Nutzung freigegeben sind. Allerdings garantiert die allgemeine *Verfügbarkeit* nicht die tatsächliche *Nutzbarkeit* der Daten durch einen Interessenten – diese ist offensichtlich nur dann gegeben, wenn Datenquellen auch *auffindbar* sind.

Dieses Problem, bestimmte Informationen oder Daten zu finden, gilt allgemein für die Inhalte des WWW und ist hauptsächlich der seit Erfindung des Webs ständig steigenden Menge an Daten geschuldet. Mit dieser immer größer werdenden Informationsflut erlangte auch die Erschließung von Informationsquellen – ob in Form von Webseiten oder von Diensten – für die Nutzer des WWW eine immer höhere Bedeutung. Als eine Möglichkeit hat sich die strukturierte Zusammenfassung und Integration von Quellen in *Portalen* oder *Webportalen* etabliert.

Im Abschnitt 3.1 wird zunächst das Konzept des Portals beschrieben. Dazu werden drei Arten von Portalen, das allgemeine Webportal, das Geoportal und das Webportal mit Raumbezug, unterschieden. Abschnitt 3.2 verdeutlicht das Webportal mit Raumbezug anhand dreier konkreter Umsetzungen und zeigt beispielhaft die Zusammenführung und Integration von Daten verteilter Quellen unter Nutzung der im vorangehenden Kapitel beschriebenen Technologien. Abschnitt 3.3 gibt ein Zwischenresümee.

3.1 Portale ins World Wide Web

Zur Umschreibung eines Zugangs in das WWW wird häufig die Metapher des Portals genutzt. So wie ein Portal im ursprünglichen Sinne die „Außentür eines Gebäudes, architektonisch, oft auch plastisch reich ausgestaltet“ (Lexikon der Zeit 2005) bezeichnet, werden themenbezogene Zugänge zum WWW ebenfalls als Portal benannt.

3.1.1 Webportal

Eine mögliche Definition eines Webportals lautet (Lexikon der Zeit 2005):

> *Webportale erschließen das World Wide Web, indem sie strukturierte Information über im Web abrufbare Dokumente anbieten. Damit ermöglichen Webportale einen benutzerfreundlichen Informationszugang und sind zentrale Anlaufstellen für das Suchen von Information, wobei diese kontextspezifisch zusammengestellt wird.*

Ein Webportal versucht also seinen Besuchern einen Zugang ins WWW zu öffnen, indem es die angesprochene Informationsflut ordnet und für den Nutzer so aufbereitet, dass Informationen an zentraler Stelle verfügbar sind. Vertiefende Betrachtungen erweitern die Inhalte eines

Portals um Funktionen und Dienstleistungen. So sind als Charakteristika eines Portals zu nennen (verändert nach Rösch 2000):

- *Einstieg:* Das Portal stellt einen zentralen Punkt für den Zugriff auf attraktive Informationen und Anwendungen dar.
- *Simplizität:* Adressaten eines Portals sind eine große Masse von Nutzern mit unterschiedlichstem Kenntnisstand in der Nutzung von Online-Medien und Werkzeugen. Für ein solches Massenpublikum muss ein Portal intuitiv bedienbar und ohne besondere technische Hürden (z.B. durch einen gängigen Webbrowser) zugänglich sein. Letzteres wird besonders durch die Nutzung verbreiteter und standardisierter Technologien erreicht (vgl. Abschnitt 2.2).
- *Leistungsfähige Suchwerkzeuge:* Über eine freie Stichworteingabe ist sowohl ein Durchsuchen des Portals als auch des gesamten WWW möglich.
- *Aggregation von Informationen:* Portale führen eine Vielzahl von Informationen zusammen; diese reichen von aktuellen News über Börsenkurse und Wetter bis hin zu Boulevardnachrichten.
- *Personalisierung*: Ein Nutzer registriert sich durch Angabe persönlicher Daten, eines Nutzernamens und Passworts. Der Umfang der persönlichen Daten kann dabei von Name und E-Mail-Adresse über Anschrift und Kreditkartennummer (bei Online-Versandhändlern) bis hin zu umfangreichen Profilen mit Eigenschaften und Interessen (bei Online-Communities, siehe unten) reichen. Durch diese Aufgabe der Anonymität bekommt ein Nutzer als Gegenleistung auf ihn zugeschnittene Inhalte oder ein persönliches Layout angezeigt. In personalisierten Bereichen werden weitere Dienste wie E-Mail (siehe unten) oder die Hinterlegung von Dateien auf einem Online-Speicherplatz angeboten.
- *Kommunikation:* Die Kommunikationsangebote eines Portals sind eng verbunden mit der Personalisierung. Ein weit verbreiteter Dienst von Portalen ist die Bereitstellung eines kostenlosen E-Mail-Kontos. Als weitere Kommunikationsmöglichkeiten können beispielsweise auch Chaträume und Diskussionsforen Teil eines Portals sein.
- *Communities:* Der Begriff Online-Communities beschreibt den Aufbau virtueller Netzwerke und Freundeskreise, Voraussetzung sind Personalisierung und Möglichkeiten zur Kommunikation. Weiterhin ist in diesen Portalen die Preisgabe umfangreicher persönlicher Informationen möglich.
- *Einkauf:* Ein Portal kann auch Ausgangspunkt für einen Einkauf über das WWW sein. So können entweder direkt auf den Portalseiten Produkte enthalten sein, oder das Portal bietet einen Preisvergleich über eine sogenannte Preissuchmaschine mit direkter Weiterleitung zu den eigentlichen Anbietern.

Die Verknüpfung von Web Services mit dem Konzept des Portals ist in zweierlei Hinsicht von Bedeutung:

- Ein Portal als zentrale Anlaufstelle für das Suchen von Informationen eignet sich hervorragend, strukturierte Informationen über Dienste anzubieten und damit menschlichen Nutzern als Dienstverzeichnis zu dienen.
- Falls die genannten Funktionen und Dienstleistungen in Form von Web Services verfügbar sind, muss ein Anbieter eines Portals nicht zwingend eigene Inhalte oder Funktionen bereitstellen, sondern kann diese über Dienste spezialisierter Anbieter nutzen und sie in beliebiger Zusammenstellung in eine Benutzeroberfläche integrieren. Damit bietet ein Portal beliebigen Nutzern eine komfortable Möglichkeit zum Zugang zu Diensten.

Abbildung 3-1 zeigt ein Bildschirmfoto eines typischen Webportals (Lycos[18]). Blickfang ist eine Boulevard-Schlagzeile ergänzt durch ein größeres Bild. Um diesen Aufhänger gruppieren sich mehrere Navigationen, die Zugang zu bestimmten Themen bieten, Schlagzeilen, sowie ein Bereich mit personalisierten Funktionen.

Abbildung 3-1: Bildschirmfoto eines typischen Webportals (Lycos, www.lycos.de)

Web-Portale lassen sich weiter differenzieren in horizontale und vertikale Portale (Lexikon der Zeit 2005). Erstere sammeln Informationen über unterschiedliche Themenbereiche, letztere fokussieren und vertiefen ein oder wenige Themen. Als spezielle Form eines vertikalen Portals gelten Unternehmens-Portale, die Kunden und Partnern eines Unternehmens Informationen über Produkte und Dienstleistungen bieten (ebd.). Gemäß dieser Differenzierung sind oben genannte Charakteristika in einzelnen Portalen in unterschiedlich starkem Maße vorhan-

[18] http://www.lycos.de (Zuletzt geprüft am 03.05.2008)

den. So spezialisieren sich einige beispielsweise auf das Online-Shopping bzw. auf Preisvergleich über Preissuchmaschinen, andere auf den Aufbau von Online-Communities.

3.1.2 Geoportal

Ein Webportal für den Zugriff auf raumbezogene Daten – im obigen Sinne ein vertikales Webportal – wird auch als Geoportal bezeichnet. Beispiele sind die Webseiten öffentlicher Institutionen, auf denen beispielsweise topographische Karten in digitaler Form über Web Map Services verfügbar sind. Eine genauere Definition eines Geoportals gibt Donaubauer (2004):

> *„Ein Geoportal ist eine Zusammenstellung einzelner Komponenten zu einem webbasierten Einstiegsknoten für die Suche nach sowie die Registrierung und Nutzung von verteilten, interoperablen Geoinformations-Ressourcen. Ein Geoportal stellt eine Mittlerinstanz zwischen Anbietern und Nutzern von Geoinformations-Ressourcen dar und erfüllt damit eine zentrale Funktion in einer dienstorientierten Geodateninfrastruktur.“*

Ein Geoportal stellt also im Sinne eines Webportals einen zentralen Einstieg für die Nutzung einer dienstorientierten Geodateninfrastruktur dar.

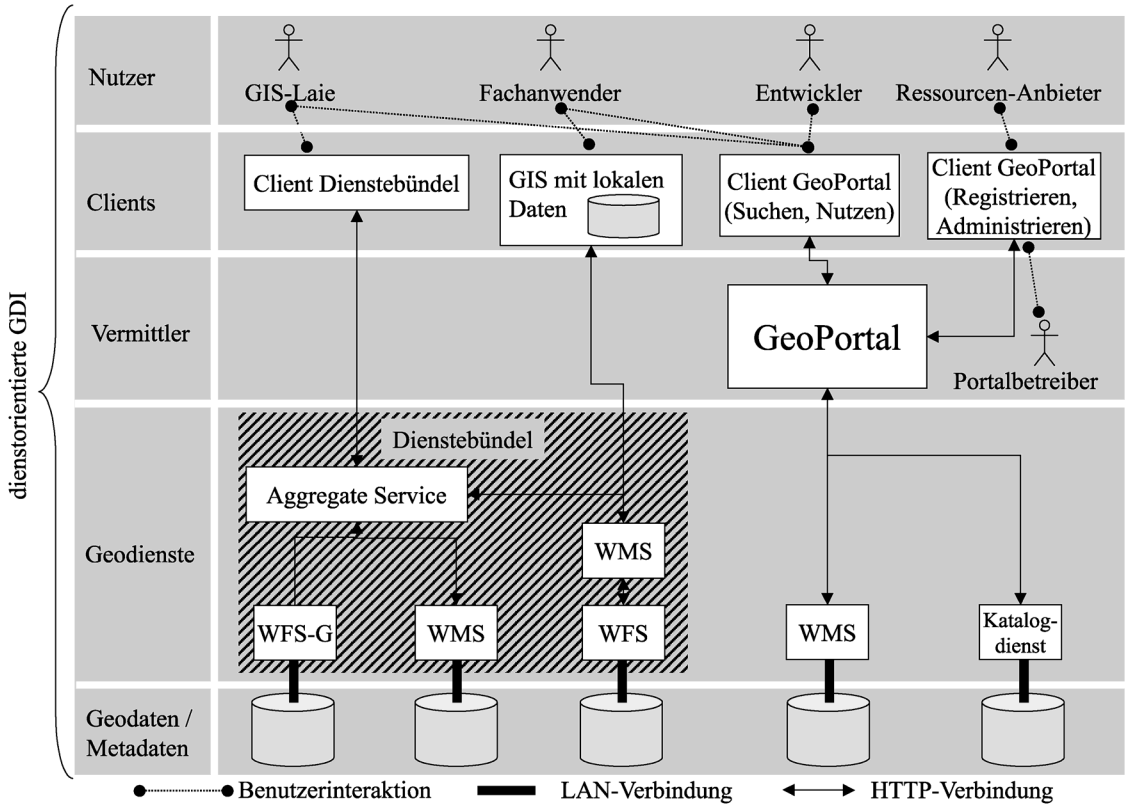


Abbildung 3-2: Geoportal als Teil einer Geodateninfrastruktur (Quelle: Donaubauer 2004)

Abbildung 3-2 zeigt die Einbettung eines Geoportals in eine Geodateninfrastruktur. Auf der untersten Ebene befinden sich die verfügbaren Daten (Geodaten und Metadaten), die über verschiedene Dienste zugänglich sind. Neben einem Web Map Service sind dies ein Dienst

zur Bereitstellung raumbezogener Objekte (Web Feature Service, WFS), ein Dienst zur Abbildung postalischer Adressen auf Koordinaten (Gazetteer, WFS-G) sowie ein Katalogdienst. Die Dienste sind über verschiedene Typen von Clients abrufbar. Ein Nutzer hat mit diesen Clients entweder direkt oder indirekt über das Geoportal Zugriff auf die Dienste. Das Geoportal steht dabei auf der Ebene eines Vermittlers.

3.1.3 Webportal mit Raumbezug

Die Erstellung von ad-hoc-Karten benötigt eine Anwendung, die im Webbrowser den Bezug und die Darstellung von Daten ermöglicht. Konzeptionell lässt sich eine solche Oberfläche als Webportal einordnen, für die Zwecke dieser Arbeit wird konkreter der Begriff *Webportal mit Raumbezug* genutzt. Als solches

- spezialisiert es ein Webportal auf Daten und Informationen, die in irgendeiner Form einen räumlichen Bezug besitzen; die Charakteristika eines Webportals bleiben weitgehend erhalten.
- verallgemeinert es ein Geoportal. Die Nutzung einer dienstorientierten GDI ist nach wie vor eine wesentliche Grundlage; darüber hinaus werden aber auch alle Arten von Daten, die sich räumlich einordnen lassen, genutzt (vgl. Abschnitt 2.3.2).

Ein Webportal mit Raumbezug bezeichnet somit ein Portal, das heterogene Informationen unterschiedlichster Art integriert und in geschlossener Form präsentiert. Zentrales Ordnungselement für die Datenintegration ist deren geo-räumliche Einordnung; zentrale Präsentationsform ist die Karte als Darstellungsform räumlicher Zusammenhänge.

Ein Webportal mit Raumbezug setzt im Kern das Konzept der *Hypermap* (Grünreich 1996, Cartwright 1999) oder *Clickable Map* (Kraak 2001, Köbben 2001) um, die im Kontext der Kartographie und der Nutzung digitaler Medien für kartographische Zwecke beschrieben werden. Eine Hypermap oder Clickable Map erlaubt die Exploration raumbezogener Datenbestände durch interaktive Werkzeuge (z.B. Zoom, Verschieben eines Kartenausschnitts). Ein Kartenbild bzw. bestimmte Objekte des Kartenbildes sind weiterhin mit sensitiven Bereichen („Hot Spots“) und weiterführenden Hyperlinks belegt. Folgt ein Nutzer einem solchen Hyperlink, gewinnt er Zugang zu vertiefenden Contents, meist in Form von Texten und Bildern. Eine Karte wird so zur Schnittstelle zu einer Vielzahl von Informationen (Kraak 2001). Allerdings wurde die Erstellung solcher Karten meist für einen in Umfang und Art feststehenden Datenbestand betrachtet; die Integration und Präsentation der Daten konnte so statisch aufeinander abgestimmt werden.

Im Kontext der Service-orientierten Architektur des Webs 2.0 bedeutet dies für das Web-Portal mit Raumbezug:

- *Mashups auf Basis einer Karte:* Die Hypermap setzt sich aus beliebigen Daten zusammen, die über öffentliche (im Fall der raumbezogenen Daten standardisierte) Schnittstellen im WWW verfügbar sind.

- *Dynamisches Mashup:* Die Zusammensetzung der Datenquellen ist nicht statisch, sondern ein Nutzer des Portals hat die Möglichkeit geeignete, seinen Bedürfnissen entsprechende Daten einzubinden.

Die besondere Herausforderung eines solchen dynamischen Mashups ist die ad-hoc-Erstellung von Karten, insbesondere

- raumbezogene Daten beliebig verteilter Quellen so zu integrieren, dass ein hochwertiges Kartenbild entsteht,
- die raumbezogenen Daten zusätzlich so mit Contents beliebiger Quellen zu kombinieren, dass eine optimale Präsentation aller enthaltenen Informationen gewährleistet ist.

Die Architektur eines Portals mit Raumbezug entspricht in erster Näherung der einer Service-orientierten Architektur im Web 2.0 (vgl. Abbildung 2-13).

Als prominentestes Beispiel, das einem Webportal mit Raumbezug sehr nahe kommt, lässt sich Google Maps[19] auffassen: Google Maps stellt in der skizzierten Weise die Karte in den Mittelpunkt und präsentiert auf dieser Basis Informationen aller Art, angefangen von punktbezogenen Daten (Texte, Bilder, Videos), bis hin zu Ergebnissen verschiedenster Suchabfragen (Adresssuche, Unternehmenssuche, Routenplanung). Kennzeichnend für Google Maps ist allerdings, dass es sich zunächst um ein geschlossenes Produkt handelt, das seine Daten – Geodaten und Contents – aus einer zentralen Quelle bezieht. Dieses proprietäre Konzept wird geöffnet, indem Google mit der Google Maps API[20] einen Client verfügbar macht, der, eingebunden in die eigene Website, den Zugriff auf die Karten der Fa. Google ermöglicht. Diese Karte kann dann durch eigene Contents (z.B. aus einem XML-File) oder durch Contents aus beliebigen Quellen zu einer einfachen Clickable Map erweitert werden. Solche Mashups auf Basis von Google Maps sind sehr weit verbreitet, eine Übersicht findet sich beispielsweise auf www.programmableweb.com (Zuletzt geprüft am 09.05.2008).

Im Vergleich zu Google Maps zeichnet sich ein Webportal mit Raumbezug vor allem dadurch aus, dass es von vorneherein, d.h. ohne dass ein Nutzer selbst einen Client anbieten muss, die Einbindung beliebiger Datenquellen ermöglicht. Dies führt weiterhin dazu, dass gerade im Bereich der raumbezogenen Daten eine sehr viel höhere Datenvielfalt verfügbar ist. In einer Ausrichtung auf eine bestimmte Thematik, beispielsweise die Freizeit- oder Fahrradtourenplanung (vgl. nachfolgenden Abschnitt), ist ein Webportal mit Raumbezug ein möglicher Weg, Mehrwertdienste für eine Geodateninfrastruktur aufzubauen.

3.2 Beispiele für Webportale mit Raumbezug

Am Institut für Kartographie und Geoinformation der Universität Bonn, ab Oktober 2006 Institut für Geodäsie und Geoinformation, Bereich Geoinformation, sind unter Mitwirkung des Verfassers dieser Arbeit in den letzten Jahren mehrere Systeme entstanden, die die Ent-

[19] http://maps.google.de (Zuletzt geprüft am 07.05.2008)

[20] http://code.google.com/apis/maps/ (Zuletzt geprüft am 07.05.2008)

wicklung von Webportalen mit Raumbezug beispielhaft verdeutlichen. Es handelt sich um drei interaktive Freizeit- und Fahrradrouten- bzw. Fahrradtourenplaner[21], die bis zum jetzigen Zeitpunkt (Mitte 2008) im WWW verfügbar sind.

Die Beispiele nutzen Karten und Contents, die nicht auf einzelne Nutzer, sondern auf bestimmte Nutzergruppen zugeschnitten sind. Diese Gruppen sollen animiert werden, ihre Freizeit in den jeweiligen Gegenden zu verbringen, indem sie sich vorab ein Bild von Landschaft und Freizeitmöglichkeiten machen und ihre Aktivitäten bereits am Rechner vorausplanen können.

Durch die Systeme wird die Integration von Daten gezeigt, die in diesem Fall nicht ad hoc gewählt, sondern aus im Voraus festgelegten Quellen bezogen werden; zwei der Systeme nutzen dafür Daten aus verteilten Quellen. Eine Umsetzung erfolgte jeweils durch Anwendung der aktuellen Möglichkeiten der in Kapitel 1 beschriebenen Technologien und ist derart generisch gehalten, dass die Systeme bereits einen großen Schritt in Richtung eines Mapping on demand darstellen.

3.2.1 „Ruhrtal à la Karte"

Ruhrtal à la Karte ist ein interaktiver Freizeit- und Fahrradtourenplaner für das Ruhrtal zwischen Schwerte im Osten und Mülheim/Ruhr im Westen. Das System ist ein Angebot des Regionalverbands Ruhrgebiet (RVR) und seit Mitte 2001 unter www.ruhrtal.de im WWW verfügbar. Ruhrtal à la Karte integriert raumbezogene Rasterkarten mit Contents (Texte, Bilder, Videos) von ca. 100 touristisch interessanten Punkten (Points of Interest). Diese Points of Interest sind über eine Koordinate georeferenziert und erweitern die Karten zu Clickable Maps. Informationen, die über die folgenden Ausführungen hinausgehen, finden sich in Steinrücken (2001).

Im Einzelnen bietet das System folgende Funktionalitäten:

- Eine Tour wird durch die Auswahl von Start-, Ziel- und Zwischenpunkten festgelegt; die Auswahl erfolgt durch interaktive Platzierung von Fähnchen auf den Knoten des Radwegenetzes.
- Die Ausgabe einer Tour erfolgt visuell im Kartenbild und durch eine Tourenbeschreibung in Textform. Beide lassen sich ausdrucken und mit „aufs Rad nehmen".
- Die Points of Interest sind durch auffällige Signaturen im Kartenbild verortet (Clickable Map). Ein Mausklick auf eine Signatur öffnet ein neues Browserfenster und zeigt die hinterlegten Inhalte an.

[21] Im Folgenden wird der Begriff Route für die rein geometrische Beschreibung eines Weges genutzt, der Begriff Tour bezeichnet eine Route mit erweiterten Informationen, beispielsweise Abbiegehinweisen in Text- oder Sprachform.

- Jeder Nutzer hat die Möglichkeit, eigene Texte oder Bilder über dem Kartenbild zu platzieren und es so zu personalisieren. Die Speicherung dieser Daten erfolgt dabei im Browser des Nutzers als Query-Teil einer URL (Kolbe et al. 2003).

Rahmenbedingungen für die Umsetzung waren Ende 2000 vor allem die Restriktionen auf Seiten der Hardware nicht-fachlicher Anwender (z.B. Bandbreite typischer Internetanschlüsse, Monitorauflösung). Weiterhin sollte als Kartenserver der Arc Internet Map Server (ArcIMS), ein Produkt der Firma ESRI zur Verfügbarmachung von Karten über das WWW, genutzt werden.

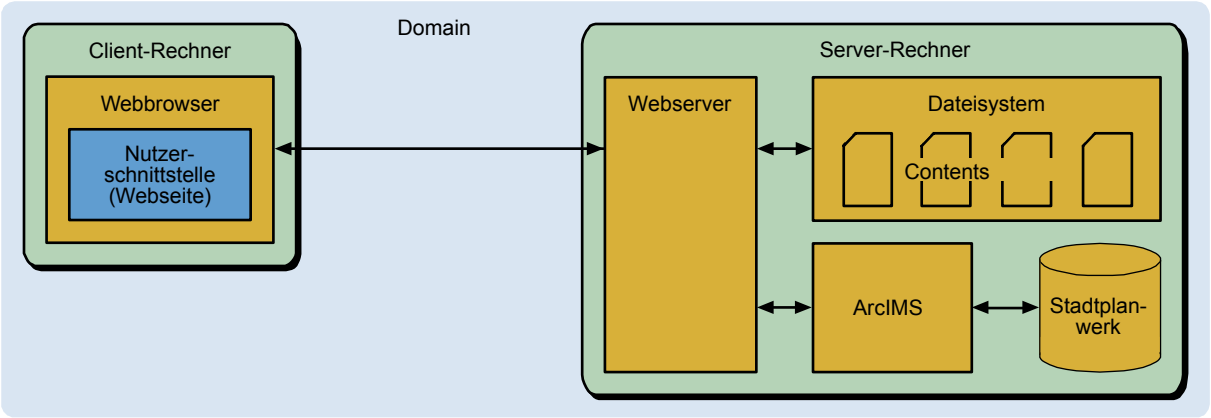


Abbildung 3-3: Architektur des Portals Ruhrtal à la Karte

Ruhrtal à la Karte nutzt eine physische Zwei- und eine logische Drei-Tier-Architektur (Abbildung 3-3). Diese ähnelt damit sehr der in Abschnitt 2.4.1 zum „frühen WWW“ beschriebenen (vgl. Abbildung 2-11). Alle Daten sind auf einem zentralen Server-Rechner integriert bzw. werden von diesem zur Verfügung gestellt, die Bedeutung der serverseitigen Komponenten wird nach einer Betrachtung der Struktur der clientseitigen Nutzerschnittstelle deutlich. Diese geht aus Abbildung 3-4 hervor. Unterhalb der Einstiegsseite sind folgende Bereiche verfügbar:

- *Charme der Region:* In mehreren Kategorien werden charakteristische Orte des Ruhrtals vorgestellt.
- *Routenvorschläge:* Besonders empfehlenswerte Touren werden vertiefend beschrieben und können direkt geladen werden.
- *Kachelübersicht:* Der durch den Tourenplaner abgedeckte Raumausschnitt ist in sieben Kacheln aufgeteilt. In der Kachelübersicht erfolgt der Einstieg in eine bestimmte Kachel und damit an einen bestimmten Abschnitt der Ruhr.
- *7 Kartenkacheln:* Diese Ebene bietet ein Kartenfenster für die Darstellung der genannten Kacheln. Diese Kacheln bestehen aus statischen Kartenbildern des kleinsten verfügbaren Maßstabs, die Erweiterung zu einer Clickable Map erfolgt durch ein Overlay von Signaturen. Alle so dargestellten Objekte sind daneben auch in einer namentlichen Aufzählung im Fenster sichtbar. Beim Überfahren einer Signatur oder eines Namens mit der Maus erfolgt synchron eine Vergrößerung der Signatur bzw. eine rote Hervor-

hebung des Namens. Ein Nutzer bekommt so sowohl eine räumliche als auch ein inhaltliche Übersicht mit einem direkten visuellen Zusammenhang. Auf dieser Ebene der Kartenkacheln ist auch die vollständige Funktionalität der Planung und Ausgabe von Touren sowie der Eingabe personalisierter Ergänzungen verfügbar. Fordert ein Nutzer über das angebotene Zoom-Werkzeug höher aufgelöste Karten an, erfolgt der nahtlose Übergang zum *Kartenclient*.

- *Kartenclient:* In diesem Kartenclient werden aus Rasterdaten zweier Maßstabsebenen (Stadtpläne) dynamisch generierte Kartenbilder angezeigt. Als Kartenserver dient der ArcIMS. Die großmaßstäbigen Karten dienen der reinen Informationsdarstellung (Kartengrundlage mit Radwegenetz, geplanten Routen und Points of Interest). Eine Planung von Touren ist hier nicht möglich.
- *Multimediaprojektoren:* In diesen sogenannten Multimediaprojektoren werden die Informationen zu Points of Interest (beschreibende Texte, Bilder, Videos, Eintrittspreise,…) präsentiert.

Alle genannten Elemente und Bereiche sind unter Verwendung von DHTML implementiert und als statische Dokumente im Dateisystem des Servers hinterlegt.

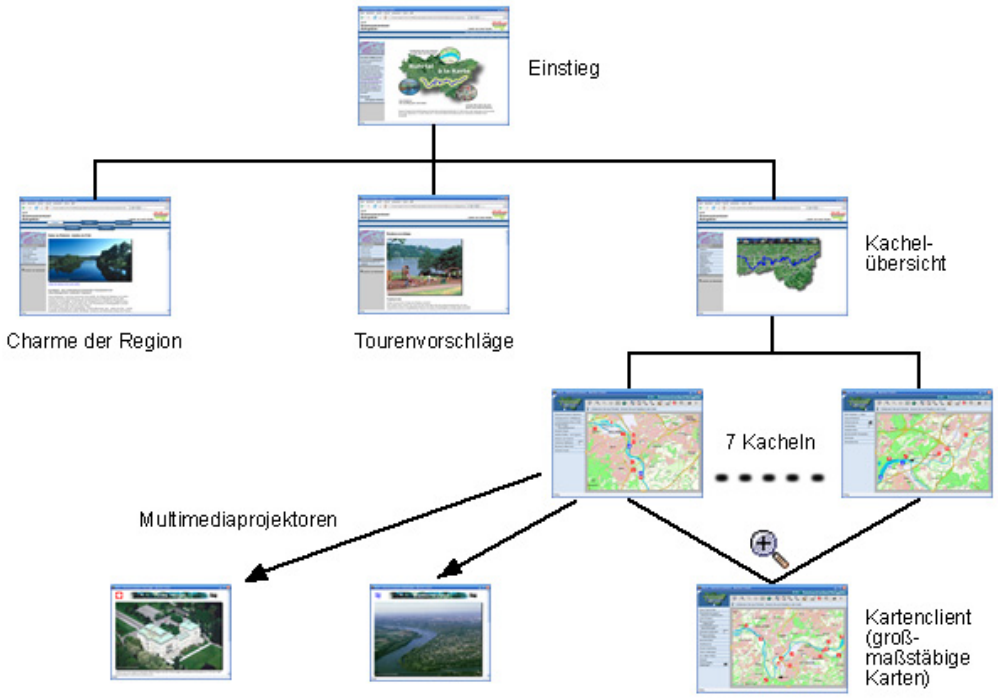


Abbildung 3-4: Struktur des Portals Ruhrtal à la Karte aus Sicht eines Anwenders

Die Unterscheidung von Kartenkacheln und Kartenclient in Abhängigkeit von der Auflösung der räumlichen Daten berücksichtigt insbesondere die Restriktion der Bandbreite typischer Internetanschlüsse. Die Beschränkung der Funktionalität der Tourenplanung auf die Kartenkacheln ermöglichte weiterhin die Implementierung eines Fat Clients und damit eines „Web-GIS ohne GIS“: Die eigentliche Routenplanung verläuft lokal im Browser des Nutzers, GIS-Funktionalität auf dem Server ist damit nicht notwendig.

Diese lokale Routenplanung wird durch Nutzung des Routingalgorithmus nach Floyd[22] umgesetzt. Durch diesen Algorithmus werden alle Routen im Voraus berechnet und als Matrix – auf Implementierungsebene als JavaScript-Array – zum Client geschickt. Das Heraussuchen einer spezifischen Route zwischen zwei oder mehr Punkten erfolgt dann auf dem Client. Da die Größe der Matrix mit dem Quadrat der Anzahl der Netzknoten zunimmt, eignet sich dieses Vorgehen allerdings nur für kleinere Netze mit höchstens einigen hundert Knoten.

Die Darstellung von Routen erfolgt ebenfalls dynamisch auf Seiten des Clients: Alle Kanten des Netzes sind als einzelne Bilder mit transparentem Hintergrund verfügbar und liegen unsichtbar im Kartenfenster. Bei Ausgabe einer Route werden mit DHTML die benötigten Fragmente sichtbar geschaltet.

3.2.2 „Grenzenlos Radfahren"

Grenzenlos Radfahren ist für das deutsch-niederländische Grenzgebiet zwischen den Flüssen Rhein und Maas verfügbar (Kreise Kleve und Viersen auf deutscher Seite, Provinz Noord- en Midden Limburg auf niederländischer Seite). Konzeptionell ähnlich zu Ruhrtal à la Karte werden Points of Interest mit Rasterkarten in Form einer Clickable Map verknüpft. Allerdings umfasst Grenzenlos Radfahren ein größeres Gebiet und eine größere Anzahl raumbezogener Daten und Contents. Die Karten bestehen in diesem Fall aus sechs Maßstabsebenen, Points of Interest sind mehr als 500 verfügbar. Weiterhin sind Teile des Radwegenetzes als überregionale Themenrouten oder vorgefertigte Tourenvorschläge ausgezeichnet.

Grenzenlos Radfahren entstand im Rahmen des Interreg III A Programms der Europäischen Union unter Federführung des Kreises Viersen und ist seit Anfang 2006 unter www.grenzenlos-radfahren.de im WWW verfügbar.

Die Funktionalität des Systems ist vergleichbar mit der von Ruhrtal à la Karte. Für die Umsetzung galten allerdings andere Rahmenbedingungen: Ein Teil der raumbezogenen Daten sollte über standardisierte Web Map Services der Geodateninfrastruktur des Landes Nordrhein-Westfalen bezogen werden, ein anderer Teil über projekteigene Dienste (ebenfalls Web Map Services). Weiterhin sollten mehrsprachige Contents der Points of Interest von verschiedenen Akteuren (Tourismusagenturen) erfasst und gepflegt werden können. Besonderheit des Systems ist das Angebot grenzüberschreitender Karten und eines grenzüberschreitenden Radwegenetzes, das eine nahtlose Planung von Touren über die deutsch-niederländische Grenze hinweg ermöglicht. Dafür muss das Wegenetz zwei Modellierungsphilosophien beiderseits der Grenze integrieren, ein Knoten-Kanten-Modell auf deutscher und eine knotenbasiertes Modell auf niederländischer Seite.

Im Bereich der raumbezogenen Daten werden folgende Web Map Services genutzt:

- Die Web Map Services der GDI in Nordrhein-Westfalen liefern Rasterkarten in fünf Auflösungsstufen, die Topographische Übersichtskarte (TÜK) 1 : 500.000[23], die To-

[22] Näheres zum Algorithmus z.B. in Güting & Dieker (2003).

[23] http://www.geoserver.nrw.de/GeoOgcWms1.3/servlet/NRW500? (Zuletzt geprüft am 12.08.2008)

pographischen Karten (TK) 1 : 100.000[24], 1 : 50.000[25], 1 : 25.000[26] und die Digitale Topographische Karte (DTK) 1 : 10.000[27].

- Ein projekteigener Web Map Service liefert Rasterdaten im ursprünglichen Maßstab 1: 80.000[28] und niederländische Rasterdaten im Maßstab 1 : 25.000[29].
- Ein projekteigener Web Map Service liefert Ausgaben des Radwegenetzes und der berechneten Routen[30].
- Ein weiterer Service berechnet auf dem Radwegenetz die angefragten Routen.

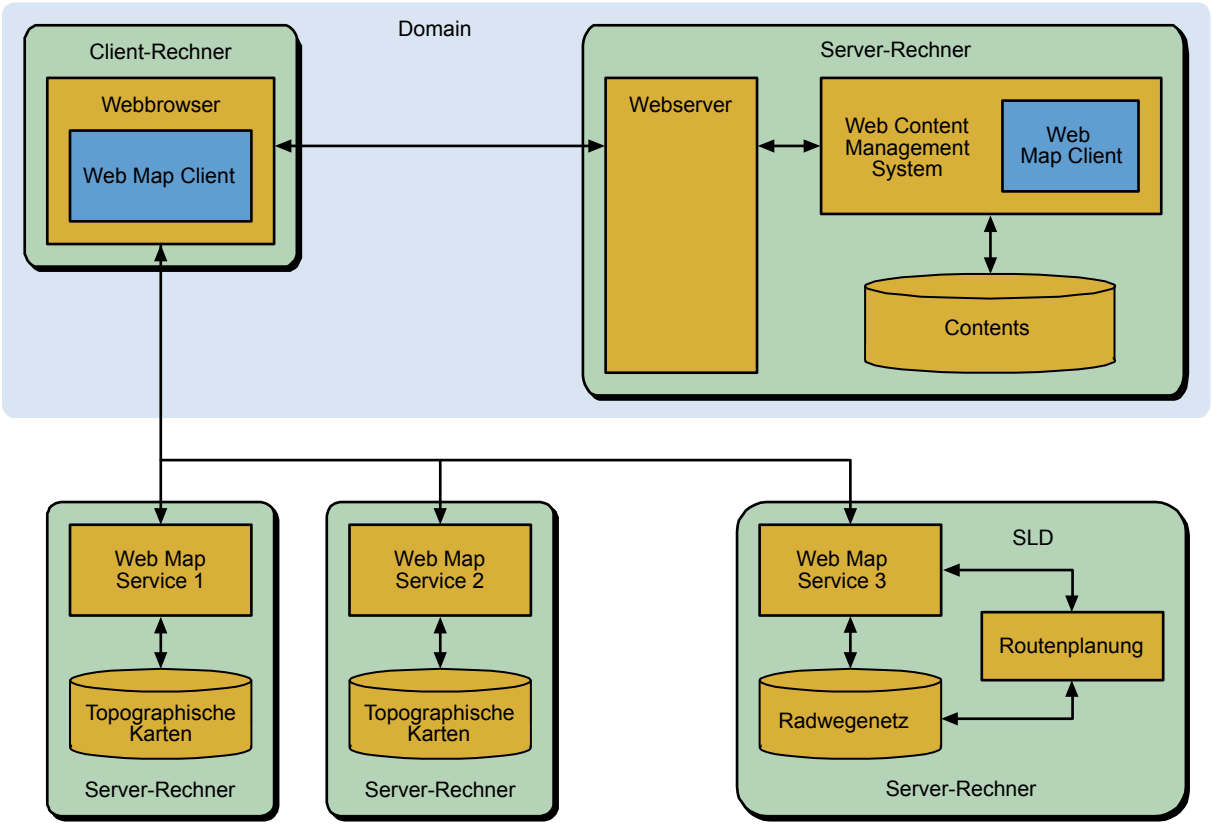


Abbildung 3-5: Architektur des Portals Grenzenlos Radfahren

Grenzenlos Radfahren nutzt eine physikalische 5-Tier-Architektur und eine logische 3-Tier-Architektur (Abbildung 3-5). Die Umsetzung erfolgte durch Kopplung der Möglichkeiten von Web Map Services mit denen von Content Management Systemen. Alle Contents der Points of Interest wurden und werden in einem Content Management System erfasst und verwaltet;

[24] http://www.geoserver.nrw.de/GeoOgcWms1.3/servlet/TK100? (Zuletzt geprüft am 12.08.2008)

[25] http://www.geoserver.nrw.de/GeoOgcWms1.3/servlet/TK50? (Zuletzt geprüft am 12.08.2008)

[26] http://www.geoserver.nrw.de/GeoOgcWms1.3/servlet/TK25? (Zuletzt geprüft am 12.08.2008)

[27] http://www.geoserver.nrw.de/GeoOgcWms1.3/servlet/DTK10? (Zuletzt geprüft am 12.08.2008)

[28] http://www.grenzeloos-fietsen.com/cgi-bin/grenzenlos? (Zuletzt geprüft am 12.08.2008)

[29] http://www.tk25nl.grenzeloos-fietsen.com/tk25nl/wms? (Zuletzt geprüft am 12.08.2008)

[30] http://grrad.grenzeloos-fietsen.com/radwege/wms? (Zuletzt geprüft am 12.08.2008)

Zugriff haben verschiedene Akteure aus dem Bereich des Tourismus direkt über ihren Webbrowser.

Die Zusammenführung der Ausgaben der Web Map Services mit Ausgaben des Content Managements Systems in Form einer Clickable Map erfolgt auf Seiten des Clients in einem *Web Map Client*; dabei wird das Ausgabeformat der Inhalte jeweils erst dynamisch zur Anfragezeit generiert. Der Web Map Client selber ist unter Verwendung von DHTML implementiert und erlaubt neben der Anzeige der Inhalte auch die Planung und Ausgabe von Touren.

In Abbildung 3-5 ist zu beachten, dass Verknüpfungen aus der Domain heraus immer die Einbindung einer Bildquelle in Form eines Web Map Services darstellt (vgl. Abschnitt 2.4.2).

Die Berechnung und Hervorhebung einer Tour im Kartenbild erfolgt über die Anfrage des WMS des Radwegenetzes. Dieser WMS bekommt über den Parameter `SLD` einen URI auf den Routenplanungsdienst übergeben. Dieser URI enthält eine Liste von Knoten, die Start-, Zwischen- und Endpunkte einer geplanten Tour repräsentieren. Durch den Aufruf dieses Verweises durch den WMS wird die Routenplanung ausgeführt und das Ergebnis als SLD-Dokument an den WMS zurückgegeben. Das SLD-Dokument enthält Style-Anweisungen und Filter, die die Darstellung nur der Kanten des Radwegenetzes bewirken, die in der berechneten Route enthalten sind. Die hier serverseitig ausgeführte Routenplanung erfolgt durch den Algorithmus von Dijkstra[31].

3.2.3 „E-RigG"

E-RigG (Referenzmodell für die Interaktion mit grenzüberschreitenden Geodaten im Internet) ist ein grenzüberschreitendes System zur Freizeit- und Radtourenplanung im westlichen Münsterland; die räumliche Abgrenzung umfasst die Stadt Bocholt auf deutscher und die Gemeinden Aalten und Winterswijk auf niederländischer Seite.

Das System tritt für einen Anwender im Wesentlichen durch zwei Komponenten in Erscheinung, einem Webportal mit Raumbezug zur interaktiven Tourenplanung und einer mobilen Komponente, die eine Tourenführung über PDA ermöglicht. E-RigG ist ein Projekt der X-Border-GDI[32]; die Umsetzung erfolgte unter Federführung der Stadt Bocholt u.a. durch das Fraunhofer-Institut für Software- und Systemtechnik (ISST) und ein lokales Konsortium aus dem westlichen Münsterland. Das Institut für Geodäsie und Geoinformation, Bereich Geoinformation, war mit der Entwicklung des interaktiven Webportals einschließlich der Tourenplanung betraut.

Das Webportal verknüpft Rasterkarten aus vier Maßstabsbereichen mit den Contents von über 500 Points of Interest; neben einer freien Tourenplanung sind auch ausgearbeitete Tourenvorschläge verfügbar. E-RigG ist seit Ende 2007 unter www.erigg.eu im WWW verfügbar. Weitere Informationen finden sich auch auf den Webseiten der X-Border-GDI (X-Border-GDI).

---

[31] Näheres zum Algorithmus z.B. in Güting & Dieker (2003).

[32] http://www.x-border-gdi.org/de/ (Zuletzt geprüft am 18.08.2008)

Die Architektur in Abbildung 3-6 zeigt eine wesentliche Änderung im Vergleich zu Grenzenlos Radfahren: Das eigentliche Portal (Server-Rechner in der Domain) stellt keine eigenen Daten mehr zur Verfügung. Stattdessen ist die Erfassung und Verwaltung der Contents der Points of Interest in eine gesonderte Komponente ausgelagert. Die Datenintegration des Portals bezieht diese Contents über eine XML-Schnittstelle.

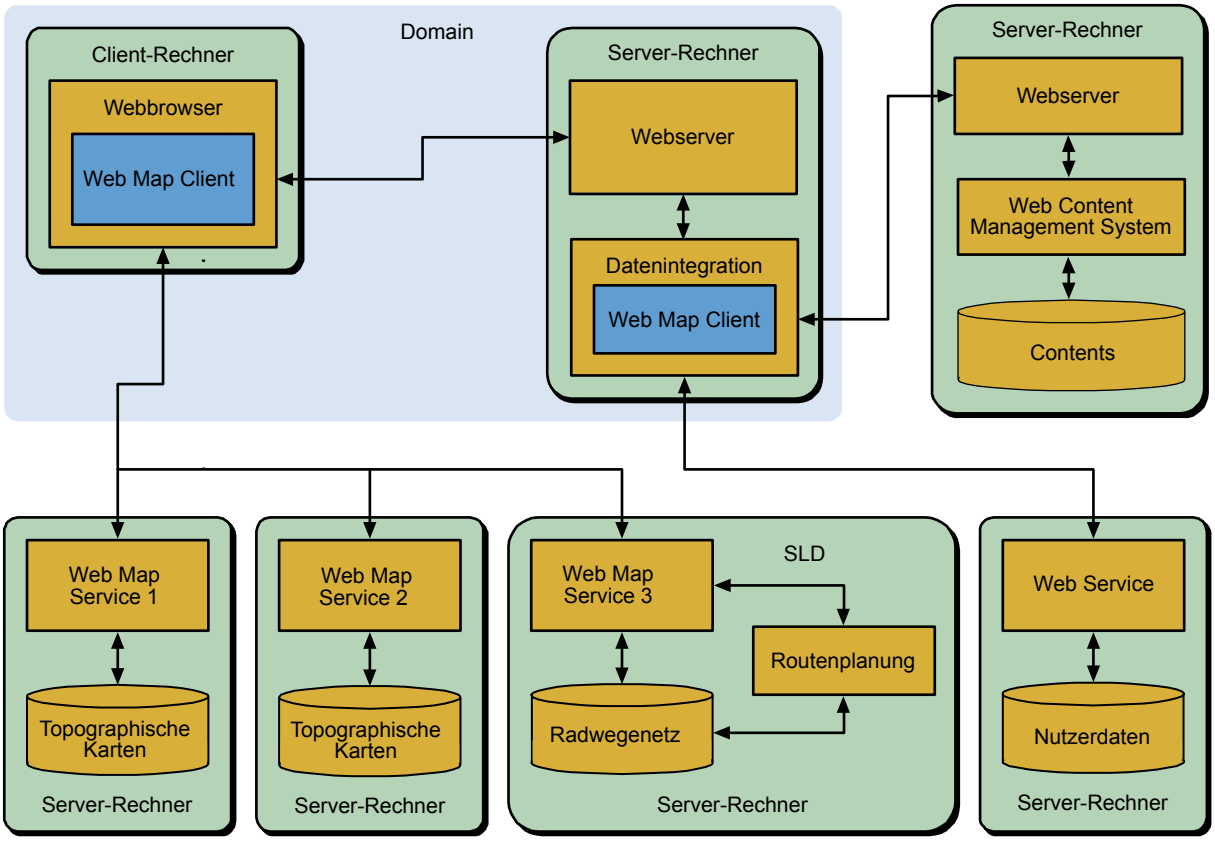


Abbildung 3-6: Architektur des Portals E-RigG

Im Vergleich zur Umsetzung von Grenzenlos Radfahren sind weiterhin folgende Änderungen hervorzuheben:

- *Ausbau der Personalisierung*: Ein Nutzer hat die Möglichkeit, sich durch die Angabe seines Namens und seiner Email-Adresse im Portal zu registrieren und so ein eigenes Nutzerkonto zu erhalten. Dieses Konto erlaubt die Speicherung und Verwaltung geplanter Touren. Diese Touren können allerdings nicht, wie bei Grenzenlos Radfahren, verschickt werden, sondern lassen sich online bestellen. Für eine so angeforderte Tour werden alle für eine mobile Tourführung benötigten Informationen zusammengestellt und auf einen PDA geladen. Dieser PDA ist dann vor Ort ausleihbar.
- *Art der Tourenplanung*: Die Planung von Touren erfolgt nicht, wie meist üblich, durch die Angabe von Start-, Ziel- und Zwischenpunkten, sondern durch einfache Aneinanderreihung von Tourpunkten, deren Reihenfolge beliebig variierbar ist. Es wird dabei zwischen zwei Arten von Punkten unterschieden: *Freie Punkte* sind willkürlich im Kartenfenster platzier- und im Nachhinein verschiebbar. *Feste Punkte* sind immer einem ganz bestimmten Point of Interest zugeordnet. Die Festlegung eines Points of In-

terest als Tourpunkt kann dabei entweder durch Auswahl einer Signatur im Kartenfenster oder durch Wahl aus einer thematisch geordneten Liste erfolgen.

- *Tourenplanung mit Präferenzen*: Die Berechnung von Routen erfolgt nicht auf einem Radwegenetz, sondern auf dem gesamten Straßennetz. Die Planung der Routen ist deshalb durch die Angabe gewisser Präferenzen (bevorzugte Straßen- und Wegearten, Wunsch nach einer landschaftlich ansprechenden Tour und die Bevorzugung komfortabler, d.h. von ihrer Beschaffenheit her für das Rad geeignete, Wege) beeinflussbar. Da es durch die Angabe von Präferenzen vorkommen kann, dass eine Tour gegenüber dem geometrisch kürzesten Weg deutlich länger wird, hat ein Nutzer die Möglichkeit, anzugeben, um wie viel Prozent länger die Tour höchstens sein darf.

3.3 Zwischenresümee

Im vorliegenden Kapitel wurde das Konzept des Webportals beschrieben und mit dem Geoportal und dem Geoportal mit Raumbezug auf die Erfordernisse dieser Arbeit spezialisiert. Das Webportal mit Raumbezug wurde durch drei Beispiele illustriert, die die Entwicklung einer weitgehend geschlossenen Anwendung des „frühen World Wide Webs“ (vgl. Abschnitt 2.4.1) hin zu einer Anwendung in der Service-orientierten Architektur des Web 2.0 (vgl. Abschnitt 2.4.2) demonstrieren. Als letztere sind Grenzenlos Radfahren und E-RigG einzuordnen. Diese Beispiele zeigen die Nutzung von Daten verteilter Quellen und standardisierter Technologien des WWW. Weiterhin wird die Kombination und Integration von Daten verschiedener Dienste (WMS) mit Inhalten, die durch Content Management Systeme bereitgestellt werden, demonstriert. E-RigG geht dabei so weit, keine eigenen Daten mehr zur Verfügung zu stellen, sondern tritt nur als Mittler für Daten auf, die bei Anforderung durch einen Nutzer von Servern Dritter bezogen werden.

Damit setzen die Systeme bereits wesentliche Aspekte eines Mapping on demand um. Eine Vereinfachung ist dadurch gegeben, dass die genutzten Datenquellen nicht on demand eingebunden werden, sondern – ebenso wie die Verknüpfung der Daten – im Vorhinein festgelegt wurden. Die Darstellungsvorschriften wurden ebenfalls im Voraus berechnet. Dieses Vorgehen ist für den Zweck der Portale ausreichend und sichert die Einbindung stabiler und verlässlicher Datenquellen. Die Konzeption und Umsetzung der Portale würde allerdings auch ad hoc die Hinzunahme weiterer Quellen gestatten.

Von dieser Hinzunahme beliebiger Datenquellen durch einen Nutzer wird in den Ausführungen der folgenden Kapitel ausgegangen. Ziel ist es dann, trotz einer Vielfalt graphischer Ausdrucksmöglichkeiten, eine hohe Prägnanz in der kartographischen Darstellung, zugeschnitten auf einen Nutzer und sein Anzeigemedium, zu erhalten. Erreicht wird dies dadurch, dass die Farben für die darzustellenden Objekte so ausgewählt werden, dass sie sich für das menschliche Auge möglichst gut unterscheiden. Die Bestimmung der Farben wird als Problem der Mathematischen Optimierung formuliert. Die wesentliche Herausforderung liegt dann darin, dieses Problem on demand zu lösen.

4 Prägnanz in der visuellen Kommunikation

In den Beispielen des letzten Kapitels wurde die Integration von Daten verschiedener Quellen beschrieben. Ergebnis waren jeweils Karten, die bestimmte Nutzergruppen bei ihrer Freizeitgestaltung und –planung unterstützen.

Allgemein stellen Karten ein Mittel dar, raumbezogenen Daten und Informationen Ausdruck zu verleihen, einem menschlichen Adressaten über die visuelle Wahrnehmung zu kommunizieren und ihm so eine Vorstellung eines raumbezogenen Sachverhalts zu vermitteln. Für die Darstellung der Daten stehen verschiedene graphische Mittel zur Verfügung, die eine Vielzahl von Ausdrucksmöglichkeiten ergeben. Ziel ist es, diese Mittel so einzusetzen, dass sich eine möglichst prägnante Darstellung ergibt, die einen Nutzer in die Lage versetzt, möglichst effektiv das für ihn Wesentliche aus einer Karte zu entnehmen.

Gegenstand dieses Kapitels sind die für die visuelle Kommunikation benötigten methodischen Grundlagen der Kartographie und die zentrale Rolle der Farbe für die Erstellung prägnanter Darstellungen.

Abschnitt 4.1 betrachtet grundlegende Begriffe und Zusammenhänge der visuellen Kommunikation, Abschnitt 4.2 fasst, ausgehend von der Graphischen Semiologie nach Bertin, das kartographische Zeichensystem und die Gestaltungsmittel für Karten zusammen. Abschnitt 4.3 gibt einen kurzen Abriss der Randbedingungen, die auf eine graphische Darstellung, insbesondere eine Karte, wirken. Im Abschnitt 4.4 wird ein gestalterischer Rahmen für die ad-hoc-Erstellung von Karten in dieser Arbeit beschrieben, bevor im Abschnitt 4.5 eine kurze Zusammenfassung dieses Kapitels erfolgt.

4.1 Visuelle Kommunikation durch Karten

Daten und Informationen werden durch die Überführung in strukturierte graphische Repräsentationen der visuellen Wahrnehmung zugänglich und lassen sich so einem menschlichen Adressaten kommunizieren. Diese Art der Informationsvermittlung wird in den verschiedensten Bereichen für eine Vielzahl von Daten angewandt; in dieser Arbeit sind die Grundlagen der Kartographie von besonderer Bedeutung.

4.1.1 Kartographische Visualisierung

Der Begriff der Visualisierung bzw. des Visualisierens bezeichnet im englischsprachigen Raum ursprünglich die Konstruktion interner mentaler Repräsentationen durch den Menschen (Ware 2000); im Kontext von Wissenschaft und Technik wird darunter seit einigen Jahren die graphische Repräsentation und Präsentation von Daten oder Informationen verstanden (Schumann & Müller 2000).

Nach Art der darzustellenden Objekte oder Daten werden zwei Arten der Visualisierung unterschieden, die *wissenschaftlich-technische Visualisierung* (Scientific Visualization) und die

Informationsvisualisierung (Information Visualization) (Zhang 2008). Gegenstand der wissenschaftlich-technischen Visualisierung sind physische Objekte (vgl. Spence 2001), die bereits einen räumlichen Bezug oder eine Geometrie besitzen und durch eine graphische Darstellung vereinfacht oder, beispielsweise aufgrund ihrer geringen realen Größe, erst zugänglich werden; Gegenstand der Informationsvisualisierung sind abstrakte Informationen wie Programmabläufe oder Datenmodellierungen (Zhang 2008). Allerdings ist anzumerken, dass die Unterscheidung der Visualisierungsarten nicht gleichzeitig ihre Eigenschaften bestimmt: Wissenschaftlich-(technische) Visualisierung ist ebenso informativ wie Informationsvisualisierung wissenschaftlich ist (ebd.).

Als Ziel der wissenschaftlich-technischen Visualisierung nennen Schumann & Müller (2000) die Erleichterung von Analyse, Verständnis und Kommunikation von Modellen, Konzepten und Daten in der Wissenschaft und im Ingenieurbereich. Diese Ziele gelten offensichtlich ebenso für die Informationsvisualisierung. Soll, beispielsweise beim Wissenserwerb, ein Sachverhalt verstanden werden, wird Visualisierungen in Form von Abbildern (Bilder, die zeigen, wie etwas aussieht (Weidenmann 2002)) die Unterstützung des Lernenden bei der kognitiven Konstruktion eines mentalen Modells des Sachverhalts zugeschrieben (ebd.).

Der Prozess der Erzeugung graphischer Darstellungen aus Daten ist durch Referenzmodelle beschreibbar. Abbildung 4-1 zeigt ein Beispiel für ein solches Modell; eine Spezialisierung, die den Visualisierungsprozess (Portrayal) des OGC wiedergibt, wurde bereits im Abschnitt 2.3.3.1 kurz vorgestellt.

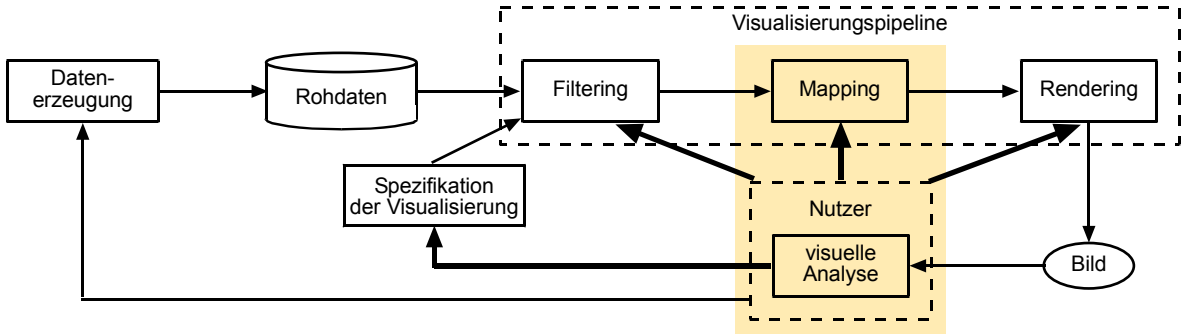


Abbildung 4-1: Integriertes Referenzmodell für die Visualisierung (geändert nach Schumann & Müller 2000)

Wesentlicher Teil des Visualisierungsprozesses in Abbildung 4-1 ist die sogenannte Visualisierungspipeline mit den Schritten *Filtering*, *Mapping* und *Rendering*: Aus Rohdaten werden durch einen Filter nach bestimmten Kriterien Daten extrahiert. Durch das Mapping wird die graphische Repräsentation dieser Daten festgelegt, indem Geometrie und Attribute bestimmt werden (vgl. Abschnitt 4.2). Durch den abschließenden Schritt des Renderings werden die Geometriedaten in Bilddaten überführt und ausgegeben. Ein Nutzer hat dabei verschiedene Möglichkeiten auf den Visualisierungsvorgang Einfluss zu nehmen (dicke Pfeile).

Der farbig hinterlegte Bereich der Abbildung 4-1 verdeutlicht den Schwerpunkt dieser Arbeit, eingeordnet in das Referenzmodell: Das Mapping von raumbezogenen Daten und Contents sowie dessen Abhängigkeit von einem Nutzer. Allerdings sollen die Abhängigkeiten in dieser

Arbeit weniger in aktiven Einflussmöglichkeiten des Nutzers, als vielmehr in dessen visuellen Möglichkeiten und Eigenschaften bestehen.

Die *kartographische Visualisierung* wird als Spezialisierung der wissenschaftlich-technischen Visualisierung auf raumbezogene Daten aufgefasst (vgl. Buziek 2001). Der Gebrauch des Begriffes in der Kartographie ist dabei eng verbunden mit der Adaption von Techniken und Ausdrucksmöglichkeiten aktueller Informations- und Kommunikationstechnologien: Das primäre Darstellungsmedium ist statt des Papiers immer häufiger ein Bildschirm oder Display (Peterson 1999); die Karte wird um multimodale Darstellungselemente erweitert (vgl. Ausführungen zur Hypermap, Abschnitt 3.1.3) und ist Teil einer interaktiven graphischen Benutzeroberfläche (Miller 1999), die einem Kartennutzer die selbstständige dynamische Exploration räumlicher Daten erlaubt. Nach MacEachren (1994) ist gerade dieser letzte Aspekt charakteristisch für die kartographische Visualisierung.

Unabhängig von sich ändernden Technologien besteht die zentrale Aufgabe der Kartographie nach wie vor in der Herstellung „*...kartographischer Ausdrucksformen als Mittel der visuellen Kommunikation über die geo-räumliche Verteilung von Daten...*“ (Grünreich 1996). Wichtigste Ausdrucksform der Kartographie ist die Karte (Hake et al. 2002), die Hake (1988, zitiert nach Hake et al. 2002) wie folgt definiert:

> *„Die Karte ist ein maßgebundenes und strukturiertes Modell räumlicher Bezüge. Sie ist im weiteren Sinne ein digitales, graphikbezogenes Modell, im engeren Sinne ein graphisches (analoges) Modell.“*

Danach ist eine Karte zunächst das Ergebnis einer räumlichen Modellierung; erst das graphikbezogene oder graphische Modell ist eine Darstellung im Sinne der Visualisierung. In dieser Arbeit ist vor allem letztere Sicht von Bedeutung.

4.1.2 Kommunikation

Vorgang und Ziel der kartographischen Visualisierung wird in der kartographischen Theorie über den Aspekt der Kommunikation in ein übergeordnetes Modell überführt. In dieser Modellvorstellung sind insbesondere die Anforderungen an eine Visualisierung bzw. eine Karte (Abschnitt 4.1.4) darstellbar.

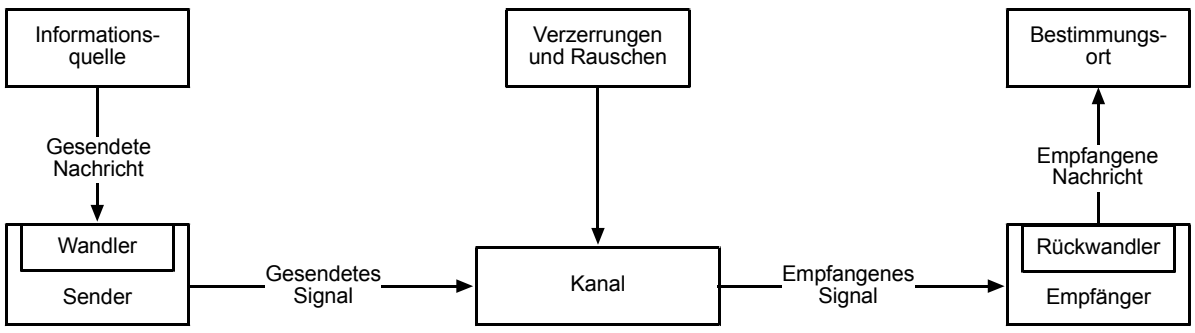


Abbildung 4-2: Nachrichtenübertragungssystem (nach Young 1975, mit eigenen Ergänzungen)

Modelle der kartographischen Kommunikation beschreiben den Prozess der Kartengestaltung durch einen Fachmann als Vermittlung und/oder Austausch geo-räumlicher Informationen an einen Nutzer. Ziel der Kommunikation ist die Gewinnung einer Umweltvorstellung und damit eine Beeinflussung des Denkens und Handelns des Nutzers. In Kürze beschreibbar ist dieser Vorgang durch den Satz „Wer sagt was zu wem mit welcher Wirkung?“ (Hake et al. 2002).

Die Modellvorstellung der kartographischen Kommunikation basiert auf dem Nachrichtenübertragungssystem der Informationstheorie (Abbildung 4-2). Dieses System beschreibt die einseitige Übertragung von Informationen oder Nachrichten von einer Informationsquelle zu einem Bestimmungsort. Eine Nachricht wird zunächst beim Sender in ein Signal umgewandelt (codiert), das sich für eine Übertragung eignet. Der Sender schickt dieses Signal über einen Kanal zum Empfänger. Dort wird es in die ursprüngliche Nachricht zurückgewandelt (decodiert) und zur Auswertung zum eigentlichen Bestimmungsort weitergeleitet. Das übertragene Signal kann auf dem Weg durch den Kanal von Verzerrungen und Rauschen beeinflusst, und die eigentliche Nachricht verfälscht werden.

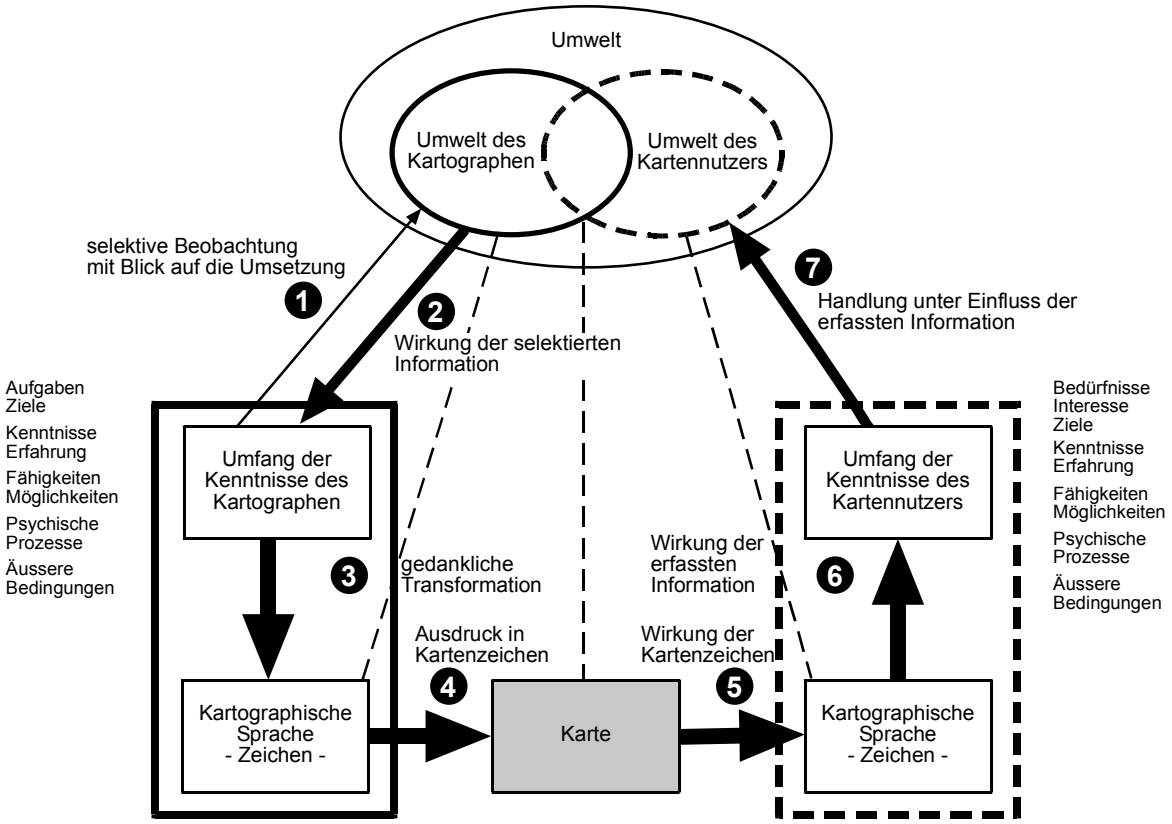


Abbildung 4-3: Modell der Kommunikation raumbezogener Informationen (nach Kolácný 1977, graphisch und inhaltlich geringfügig geändert)

Abbildung 4-3 zeigt die Adaption des informationstheoretischen Modells der Nachrichtenübertragung auf die Kartographie durch Kolácný (1977). Dieses Modell enthält keine Verzerrungen und Rauschen, die auf den Kanal „Karte“ einwirken. Die auftretenden Störungen betreffen in dieser Darstellung ausschließlich die Person des Kartographen und des Kartennutzers (Abbildung 4-3, ganz links und ganz rechts).

Die in diesem Ablauf in Form verschiedener Modelle vorliegende kartographische Information wird auch durch die Modelltheorie beschrieben (Hake et al. 2002). Informationen, die durch einen Fachmann aus einer direkten Umweltbeobachtung gewonnen werden, stellen das *Primärmodell (PM)* dar. Aus diesem Primärmodell fertigt der Kartograph ein *Sekundärmodell (SM)* in Form einer Karte oder digitaler Daten. Ein Karten- bzw. Datennutzer gewinnt daraus ein *Tertiärmodell (TM)* der Umwelt. Dieses Tertiärmodell ist dadurch gekennzeichnet, dass es sich nicht um ein reales, sondern lediglich um ein abstraktes Vorstellungsmodell handelt.

Im Kontext der kartographischen Visualisierung und deren Möglichkeiten der interaktiven Datenexploration über digitale Medien bildet Buziek (2001) den Prozess der Kommunikation mit Hilfe der Kybernetik[33] auf ein dynamisches System nach dem Regelkreisprinzip ab (Abbildung 4-4). Ein solches *Kartographisches Informationsverarbeitungssystem (KIVS)* dient der Transformation von Geodaten in individuelle Geoinformation.

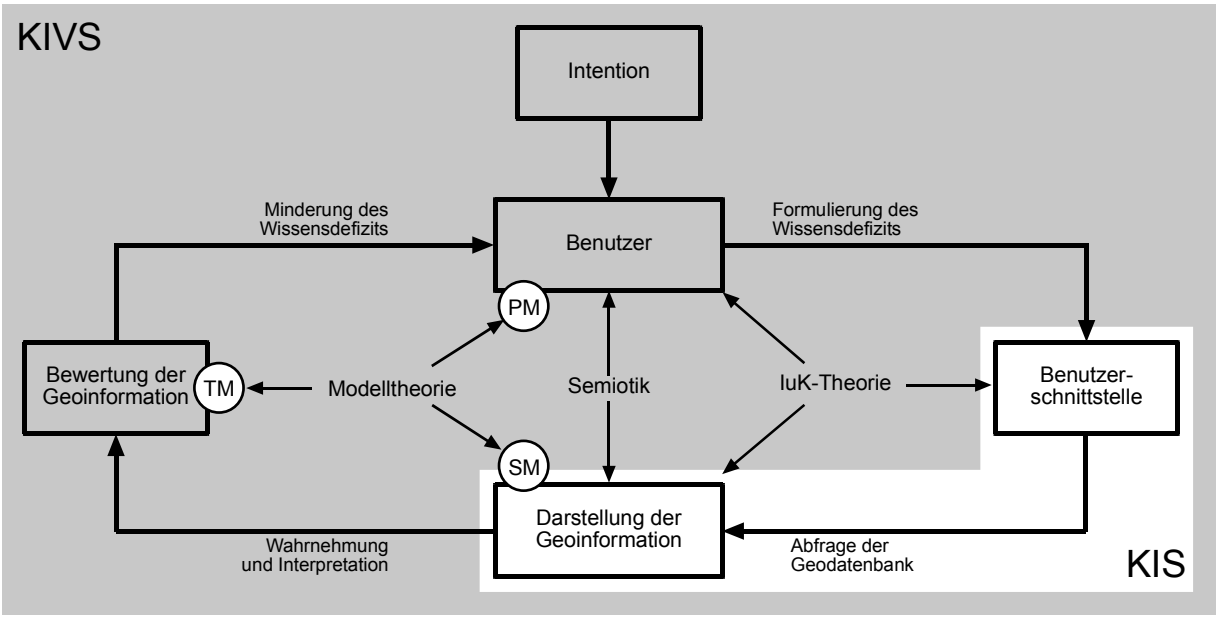


Abbildung 4-4: Kommunikation zwischen Benutzer und kartographischem Informationssystem (KIS) nach dem Regelkreisprinzip (nach Buziek 2001)

Ausgangspunkt des Regelkreises ist eine Intention bzw. ein Wissensdefizit eines Kartennutzers. Zur Behebung dieses Wissensdefizits fragt der Nutzer durch Interaktionen ein *Kartographisches Informationssystem (KIS)* an und erhält (Geo-)Informationen, die durch ihn wahrgenommen, interpretiert und bewertet werden können. Falls das Wissensdefizit behoben wurde, ist das Ziel erreicht, andernfalls werden iterative Durchläufe des Regelkreises notwendig.

Das KIVS integriert die Modelltheorie der Kartographie (PM, SM, TM, siehe oben) sowie die Informations- und Kommunikationstheorie (IuK-Theorie). Entsprechend der IuK-Theorie können im Kommunikationsablauf des Regelkreises sowohl nutzerseitig als auch auf Seiten

[33] Wissenschaft, die sich mit der mathematischen und formallogischen Beschreibung von Systemen befasst (Bollmann & Koch 2002).

des kartographischen Informationssystems Störungen auftreten und iterative Systemdurchläufe nötig machen.

Wesentliches Charakteristikum des KIVS ist also die Aktivität des Nutzers und seine selbstständige Erschließung von Informationen zur Minderung seines Wissensdefizits. Aus Sicht der Lerntheorie fördert eine solche Selbststeuerung eines Lernprozesses die individuellen mentalen Konstruktionsprozesse (Strzebkowski & Kleeberg 2002).

4.1.3 Semiotik

Wie im Rahmen der Informationstheorie bereits angedeutet wurde, ist Kommunikation immer durch den Austausch von Zeichen gekennzeichnet. Voraussetzung für eine „sinnvolle" Kommunikation ist ein gemeinsames Zeichenrepertoire und eine einheitliche Zeichenbedeutung für beteiligte Kommunikatoren (Hake et al. 2002).

Morris (1972) bezeichnet den Prozess, in dem etwas als Zeichen fungiert, als Zeichenprozess oder Semiose, die Zeichentheorie oder Semiotik als die Wissenschaft von den Zeichenprozessen.

Der Prozess der Semiose wirkt zwischen vier Faktoren oder Korrelaten, dem *Designaten*, dem *Zeichenträger*, dem *Interpretanten* und dem *Interpreten* (Morris 1972). Ein Designat ist ganz allgemein etwas (z.B. Ereignis, Objekt), von dem über den Zeichenträger (mittelbar) Notiz genommen wird. Der Vorgang der Notiznahme wird als Interpretant bezeichnet, das Subjekt des Vorgangs als Interpret (ebd.).

Auf dieser Grundlage werden verschiedene Relationen oder Dimensionen zwischen den Korrelaten unterschieden (Morris 1972):

- Die *syntaktische Dimension* des Zeichenprozesses (Syntaktik) bezeichnet die Relation der Zeichenträger untereinander.
- Die *semantische Dimension* des Zeichenprozesses (Semantik) bezeichnet die Relation zwischen Zeichenträger und Designat, d.h. die Beziehung zwischen den Zeichen und den Objekten, die sie bezeichnen.
- Die *pragmatische Dimension* (Pragmatik) beschreibt die Relation zwischen Zeichenträger und Interpreten und damit besonders, inwiefern die Wahrnehmung eines Zeichens das Verhalten des Interpreten beeinflusst.

Wie bereits aus der Definition der Syntaktik hervorgeht, betrachtet die Semiotik nicht nur einzelne Zeichen, sondern auch Zeichensysteme. Ein Zeichensystem besteht aus einer Menge funktionsbezogener Zeichen (Bollmann & Koch 2002), die, systematisch kombiniert, eine Vielzahl von Ausdrucksmöglichkeiten ergeben.

Eine Beschreibung von Zeichensystemen graphischer Darstellungen wird im Abschnitt 4.2 gegeben.

4.1.4 Qualität und Eignung einer Visualisierung

Für jede Art von Information ergeben sich mit den Mitteln des graphischen Systems (Abschnitt 4.2) eine Vielzahl von Möglichkeiten, graphische Darstellungen umzusetzen. Daraus folgt offensichtlich die Frage, welche Art einer graphischen Darstellung sich für einen bestimmten Zweck möglichst gut eignet. Die Eignung lässt sich an einigen Anforderungen an Visualisierungen festmachen; diese Anforderungen erlauben gleichzeitig die Bewertung einer Visualisierung. Basis ist der oben geschilderte Kommunikationsprozess bzw. die Informationsentnahme aus einer graphischen Darstellung durch einen Betrachter.

Bertin (1974) empfiehlt Darstellungen, die die Betrachtungszeit bzw. den „geistigen Aufwand" zur Wahrnehmung der Darstellung minimieren. Solche Abbildungen charakterisiert er durch den Begriff der „Prägnanz":

> *„Wenn eine Konstruktion zur Beantwortung einer gestellten Frage unter sonst gleichen Voraussetzungen eine kürzere Betrachtungszeit erfordert als eine andere Konstruktion, so bezeichne man diese als prägnanter in Bezug auf die gestellte Frage."*

Die prägnantesten Konstruktionen sind nach Bertin diejenigen, die die Beantwortung einer Frage während eines einzigen Augenblicks der Wahrnehmung mittels einer einzigen Graphik ermöglichen (ebd.).

Nach dieser Definition der Prägnanz hängt also die Eignung einer graphischen Darstellung ausschließlich von dieser selbst ab, der Betrachter und seine Eigenschaften gehen darin nicht ein. Weiterhin geht Bertin davon aus, dass die Information einer graphischen Darstellung durch einen Betrachter immer vollständig erfasst wird. Im Sinne des Regelkreises nach Buziek (vgl. Abschnitt 4.1.2) wären demnach keine iterativen Durchläufe aufgrund mangelhafter Interpretation der graphischen Darstellung möglich bzw. nötig.

Jüngere Arbeiten differenzieren dagegen stärker. Schumann & Müller (2000) definieren die Qualität einer Visualisierung wie folgt:

> *„Die Qualität einer Visualisierung definiert sich durch den Grad, in dem die bildliche Darstellung das kommunikative Ziel der Präsentation erreicht. Sie lässt sich als das Verhältnis von der vom Betrachter in einem Zeitraum wahrgenommenen Information zu der im gleichen Zeitraum zu vermittelnden Information beschreiben. Die Qualität einer Visualisierung ist somit in starkem Maße abhängig von den Charakteristika der zugrunde liegenden Daten und ihren Eigenschaften, dem Bearbeitungsziel, den Eigenschaften des Darstellungsmediums sowie den Wahrnehmungskapazitäten und den Erfahrungen des Betrachters."*

Offensichtlich fügt sich diese Definition gut in die Abläufe und Begrifflichkeiten der kartographischen Kommunikation (Abschnitt 4.1.2, Abbildung 4-3): Die Qualität einer graphischen Darstellung wird durch das Maß bestimmt, in dem ein Nutzer eine Vorstellung der realen Umwelt gewinnt. Dabei wirken auf Seiten des Kartographen und des Nutzers verschiedene Störungen auf diesen Prozess.

Als konkrete Anforderungen an eine Visualisierung nennen Schumann & Müller (2000) weiter:

- *Expressivität*: Eine Visualisierung soll die abzubildende Datenmenge möglichst unverfälscht darstellen, insbesondere sollen nur die Informationen zum Ausdruck gebracht werden, die auch in den darzustellenden Daten enthalten sind. Grundvoraussetzung für eine expressive Visualisierung ist eine adäquate Wahl der graphischen Mittel, beispielsweise bei der Darstellung von Daten in Diagrammen die Wahl eines geeigneten Diagrammtyps.
- *Effektivität*: Eine Visualisierung ist effektiv, wenn sie unter Berücksichtigung von Zielsetzung und Kontext der Anwendung die Fähigkeiten eines Betrachters und die Eigenschaften des Ausgabemediums optimal ausnutzt.
- *Angemessenheit*: Auch wenn eine Visualisierung expressiv und effektiv sein soll, sollte der Aspekt des Herstellungsaufwands beachtet werden, d.h. der Aufwand für die Umsetzung einer Visualisierung sollte ihrem Nutzen angemessen sein.

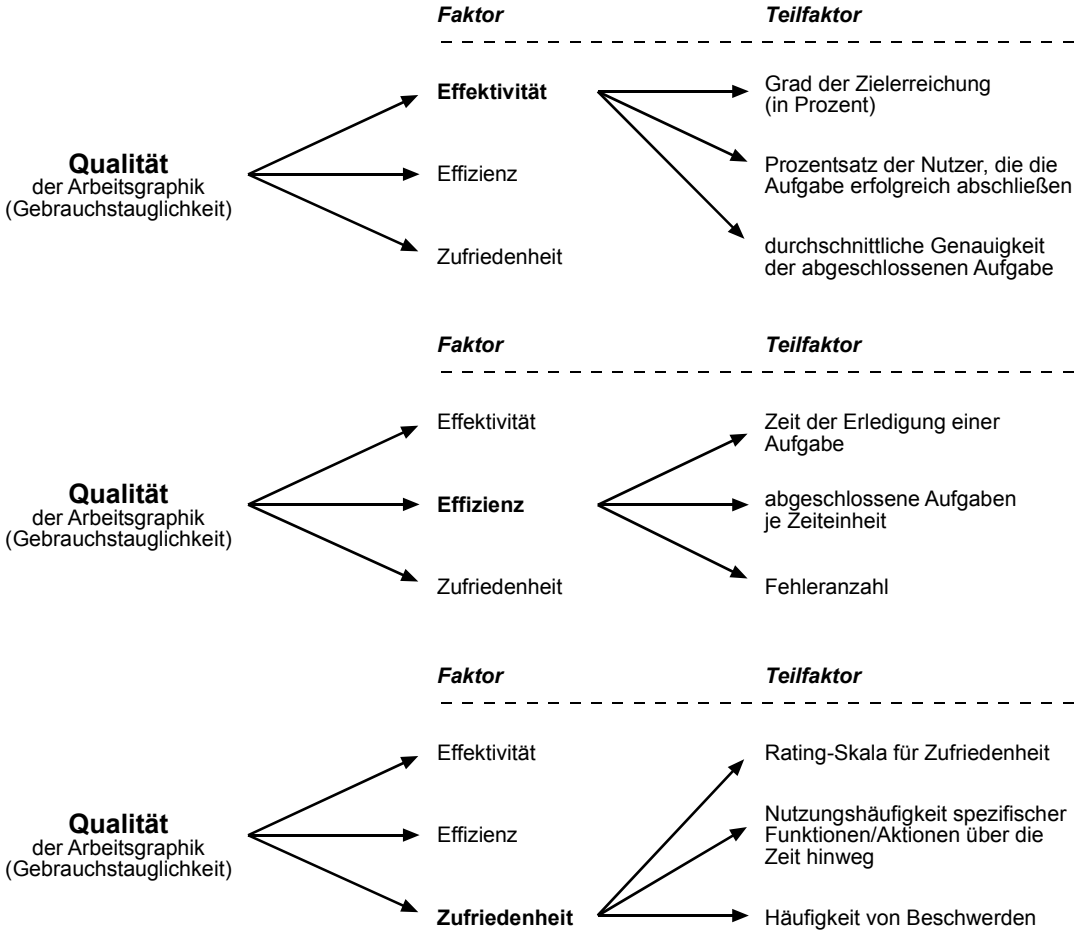


Abbildung 4-5: Maße der Gebrauchstauglichkeit von Arbeitsgraphik (nach Heidmann 1999)

Ein Vergleich dieser Anforderungen mit der Prägnanz nach Bertin lässt vor dem Hintergrund des Gesamtwerks der Graphischen Semiologie und dessen Beschreibung einer Vielzahl präg-

nanter und weniger prägnanter Darstellungen den Schluss zu, dass der Begriff der Prägnanz die Anforderungen der Expressivität und Effektivität subsumiert.

Zahlenmäßig erfassbare Faktoren beschreibt Heidmann (1999) bei der empirischen Bestimmung der Qualität oder Gebrauchstauglichkeit von Arbeitsgraphiken[34]; er charakterisiert diese Qualität durch die Faktoren Effektivität, Effizienz und Zufriedenheit eines Nutzers. Die Bestimmung der Faktoren erfolgt über jeweils drei Teilfaktoren (Abbildung 4-5), die bei der Bearbeitung von Aufgabenstellungen durch Probanden objektiv (im Falle der Effektivität und Effizienz) und subjektiv (im Falle der Zufriedenheit) messbar sind.

4.2 Graphische Zeichensysteme und Gestaltungsmittel

Die Abbildung von Daten oder Informationen auf eine graphische Darstellung – das Mapping im Sinne der Visualisierungspipeline – erfolgt nach einer Methodik, die den Daten bestimmte graphische Elemente und Attribute zuordnet. Dabei handelt es sich um Zeichen, die in ihrer Gesamtheit ein Zeichensystem im Sinne der Semiotik bilden.

Dieser Abschnitt fasst wesentliche Aspekte graphischer Zeichensysteme zusammen. Obwohl der überwiegende Teil der Ausführungen für alle Arten von Visualisierungen Gültigkeit besitzt, liegt der Fokus auf der Darstellung raumbezogener Daten in Karten.

4.2.1 Graphische Semiologie nach Bertin

Eine systematische Beschreibung der graphischen Gestaltungsmöglichkeiten wurde von Jacques Bertin mit seiner „Sémiologie graphique“ 1967 (Bertin 1967) veröffentlicht; die deutsche Übersetzung „Graphische Semiologie“ folgte 1974 (Bertin 1974). Bertin entwirft in seinem Werk eine grundlegende Methodik graphischer Darstellungsmittel und beschreibt deren Anwendung auf drei wesentliche Arten graphischer Darstellungen in der Ebene: Diagramme, Netze und Karten. Aufgrund der fundamentalen Bedeutung der Graphischen Semiologie für alle Disziplinen der Visualisierung (vgl. bspw. Schumann & Müller 2000, MacEachren 1995), werden in diesem Abschnitt einige Grundzüge und wesentliches Vokabular des Werkes zusammengefasst.

Ursprüngliches Darstellungsobjekt der Graphischen Semiologie ist der Gedanke, dessen Ausdruck in einem beliebigen System von Zeichen möglich ist. Der transkribierbare Inhalt eines Gedankens stellt die Information dar; bei einer graphischen Darstellung handelt es sich dann um die Transkription eines Gedankens (einer durch irgendein Zeichensystem bekannten Information) in das graphische Zeichensystem.

Jede Information besteht aus einer Invarianten und einer oder mehrerer Komponenten: Die Invariante stellt den unveränderlichen Teil einer Information dar, die Komponenten den vari-

[34] Das Konzept der Arbeitsgraphik fasst Modellansätze der aufgaben- und nutzerorientierten Modellierung kartographischer Medien zusammen (Bollmann & Koch 2002). Eine Arbeitsgraphik unterstützt beispielsweise die Nutzung von dynamisch-interaktiven Bildschirmkarten durch graphische Aktionen, z.B. die Bewegung oder Veränderung von Kartenobjekten (Heidmann 1999).

ablen Teil. Damit setzt sich eine Information aus Beziehungen zwischen Komponenten zusammen. Eine graphische Darstellung wird wesentlich durch Eigenschaften dieser Komponenten charakterisiert:

- *Anzahl der Komponenten*: Mit maximal drei Komponenten ist eine Information in Form eines graphischen Bildes wahrnehmbar. Liegen mehr als drei Komponenten vor, ist für das Verständnis der Information die sukzessive Wahrnehmung mehrerer Bilder erforderlich.
- *Länge einer Komponente*: Die Länge einer Komponente bezeichnet die Anzahl nominaler Elemente, Werte oder Kategorien, die für eine Komponente unterschieden werden können. Die Länge bestimmt die Komplexität einer Abbildung.
- *Gliederungsstufen der Komponenten*: Die Gliederungsstufen bezeichnen die Beziehungen zwischen Komponenten bzw. den Elementen oder Werten innerhalb einer Komponente:
    - Die *Qualitative* (kombinatorische) *Stufe* unterscheidet einfache Begriffe innerhalb einer Komponente. Es gibt keine allgemeingültige, eindeutige Reihenfolge.
    - Die *Ordnungsstufe* umfasst Komponenten, deren Elemente sich allgemeingültig und eindeutig in eine Reihenfolge bringen lassen. Elemente dieser Reihenfolge lassen sich durch Aussagen wie „mehr als“ oder „weniger als“ vergleichen.
    - Die *Quantitative* (metrische) *Stufe* ist eine strenge Form der Ordnungsstufe; eine Reihenfolge bzw. die Unterschiede der Ordnung lassen sich darin zahlenmäßig genau angeben.

In ihrer graphischen Umsetzung werden Komponenten als *visuelle Variablen* bezeichnet; zwei weitere Variablen sind durch die beiden Dimensionen der Ebene gegeben.

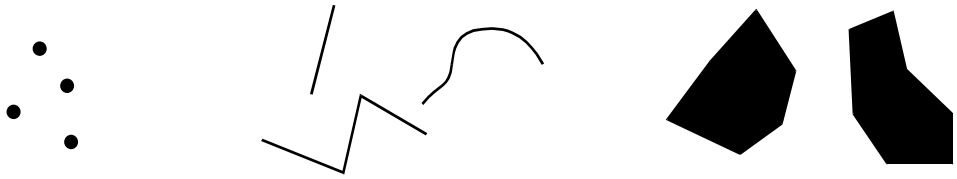


Abbildung 4-6: "Sichtbare Flecken" in den Implantationen Punkt, Linie und Fläche

Letztere geben die Lage des grundlegenden Elements der graphischen Darstellung, des „sichtbaren Flecks“, an. Innerhalb der Dimensionen kann ein Fleck in drei Bedeutungen (*Implantationen*) auftreten, als Punkt (nulldimensional), Linie (eindimensional) oder Fläche (zweidimensional) (Abbildung 4-6). Ein sichtbarer Fleck ist außer in seiner Lage noch durch die *Farb-Muster-Variablen* oder *Variablen der 3. Dimension* in seiner Darstellung veränderbar. Bertin unterscheidet dafür *Größe*, *Helligkeitswert*, *Muster*, *Farbe*, *Richtung* und *Form*. Insgesamt sind damit acht visuelle Variablen verfügbar.

Für die Beschreibung der Eignung der Variablen für bestimmte Darstellungen ist eine Charakterisierung in Bezug auf die Wahrnehmbarkeit vonnöten. Analog zu den Gliederungsstufen der Komponenten werden dafür Gliederungsstufen der Variablen unterschieden:

- Eine Variable ist *selektiv* ($\neq$), wenn sich spontan alle Beziehungen isolieren lassen, die ein- und dieselbe Kategorie dieser Variablen darstellen (Frage nach dem „Wo“).
- Eine Variable ist *assoziativ* ($\equiv$), wenn sich spontan alle Beziehungen zusammenfassen lassen, die durch diese Variable differenziert werden.
- Eine Variable ist *geordnet* (O), wenn sich die Stufen spontan in eine allgemeingültige visuelle Reihenfolge bringen lassen.
- Eine Variable ist *quantitativ* (Q), wenn sich der Abstand zwischen den Stufen einer Ordnung spontan durch ein Zahlenverhältnis ausdrücken lässt.

Anhand dieser Gliederungsstufen lassen sich die Ausdrucksmöglichkeiten jeder Variablen beschreiben (siehe auch Abbildung 4-7):

Dimensionen der Ebene: Ein sichtbarer Fleck ist als Punkt, Linie oder Fläche in Bezug auf seine Lage in der Ebene variier- bzw. festlegbar. Die Ebene verfügt als einzige Variable über alle Gliederungsstufen. Gleichartige Zeichen, beispielsweise zwei Kreise gleicher Größe, die sich an verschiedenen Orten der Ebene befinden, werden als verschieden angesehen (Selektivität), andererseits aber trotz ihrer Lage als gleichartig erkannt (Assoziativität). Weiterhin können Zeichen entlang einer Geraden geordnet werden, Quantitäten sind über Schätzung oder Messung von Strecken, Winkeln und Flächen wahrnehmbar.

Größe: Punkthafte Objekte können in ihrer Größe, linienhafte in ihrer Breite variiert werden. Bei flächenhaften Objekten sind Bestandteile in Form von Punkten oder Linien veränderbar. Die Variation der Größe ist dissoziativ, unterstützt aber die Darstellung von Selektivität, Ordnung und Quantität. Die größtmögliche Länge einer Komponente ist abhängig von der Gliederungsstufe: Die selektive Wahrnehmung erlaubt die Unterscheidung von maximal fünf Stufen, die geordnete und quantitative erlauben grundsätzlich unbegrenzt viele Stufen, für das Auge unterscheidbar sind unter bestimmten Rahmenbedingungen lediglich bis zu zwanzig Stufen.

Helligkeitswert: Die Skala der Grautöne von Weiß nach Schwarz entspricht einer Variation des Helligkeitswerts. Durch die Helligkeit lassen sich Informationen in selektiver und geordneter Form darstellen. Die Länge der Variation sollte für die Selektivität höchstens sieben Stufen (einschließlich Schwarz und Weiß) umfassen, die Darstellung der Ordnung ist unabhängig von der Stufenanzahl.

Muster: Ein Muster ist ein „Feld von Flecken“, dessen Variation sich durch eine Folge von Verkleinerungen bei konstantem Helligkeitswert ergibt. Unterschiedliche Muster werden durch Variation der Form und des Helligkeitswerts erhalten, die Abfolge der Flecken weist dabei nach Art eines Rasters immer eine gewisse Regelmäßigkeit auf. Die Wahrnehmung des Musters ist assoziativ, selektiv und geordnet. Die Länge der Variablen ist sehr stark von der Implantation abhängig. In punkthafter Implantation – für die ein Gebrauch von Mustern ausreichend große Zeichen voraussetzt – können lediglich zwei oder drei selektive Stufen unter-

schieden werden, bei linienhafter Implantation mit einer Linienbreite von 1 mm drei oder vier selektive Stufen. Muster in flächenhafter Implantation erlauben in Bezug auf die Ordnung zahlreiche Stufen, in Bezug auf die Selektivität vier oder fünf Stufen.

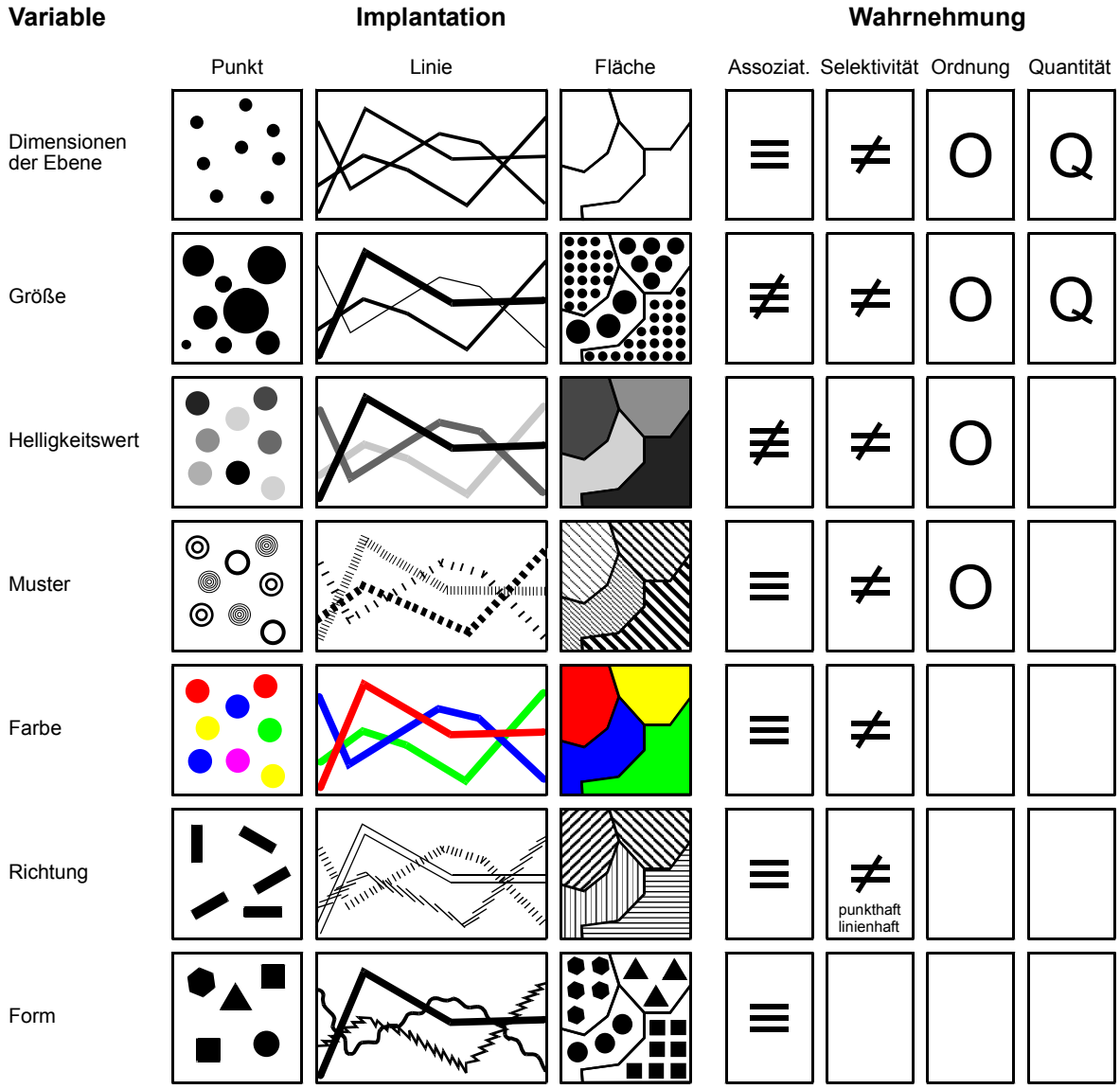


Abbildung 4-7: Übersicht über die acht visuellen Variablen nach Bertin; die Bedeutung der Zeichen im rechten Bereich geht aus der Beschreibung der Gliederungsstufen der Variablen hervor (nach Bertin 1974, Ellsiepen 2005)

Farbe: Für den Gebrauch von Farben ist zu beachten, dass diese durch drei unabhängige Dimensionen bestimmt werden (Kapitel 5.1). Eine Variation wird erreicht durch Änderung von Farbton, Sättigung und Helligkeit. Dies führt dazu, dass die Skala der Farben maximaler Sättigung unterschiedliche Helligkeitswerte besitzt, andererseits die Farben gleicher Helligkeit unterschiedliche Sättigung.

Als Variation der Farbe definiert Bertin deshalb die Änderung des Farbtons bei konstantem Helligkeitswert. Diese Variation ist selektiv und assoziativ, wobei die Selektivität für gesättigte Farben am höchsten ist.

Nach Bertin ist eine Ordnung nur durch die Variation des Helligkeitswertes erreichbar; die Verwendung von Farben ist dann verzichtbar.

Richtung: Zeichen können durch die Änderung ihrer Richtung variiert werden. Voraussetzung ist, dass die Form des Zeichens die Wahrnehmung der Richtungsänderung ermöglicht, d.h. von länglicher Form ist. Die Wahrnehmung der Richtung ist assoziativ und in punkt- und linienhafter Implantation selektiv. Im Falle der Selektivität sollte die Anzahl der Stufen auf vier beschränkt werden, die Winkel sollten eher 30° und 60° statt 45° betragen.

Form: Die Anzahl der Variationen der Form eines Zeichens – und damit die Länge der Komponenten – ist unbegrenzt. Allerdings eignet sich die Form lediglich zur Unterstützung der assoziativen Wahrnehmung.

4.2.2 Kartographisches Zeichensystem

Das kartographische Zeichensystem, auch als Kartengraphik bezeichnet (Hake et al. 2002), geht von den gleichen Grundelementen und Variablen aus, wie das Bertinsche System; in einem dreistufigen Aufbau werden allerdings weitere Darstellungselemente unterschieden (ebd.):

- Die unterste Stufe des kartographischen Zeichensystems wird durch die *Graphischen Elemente* gebildet: Nach ihrer geometrischen Ausprägung werden Punkte, Linien und Flächen als Bausteine jeder Graphik unterschieden. Diese Stufe entspricht den Bertinschen Implantationen des sichtbaren Flecks in der Ebene.
- Zusammenfügungen graphischer Elemente zu komplexeren Gebilden werden als *Zusammengesetzte Zeichen* bezeichnet. Neben dem Bekanntesten, der Signatur, werden Diagramm, Halbton und Schrift unterschieden (Abbildung 4-8).

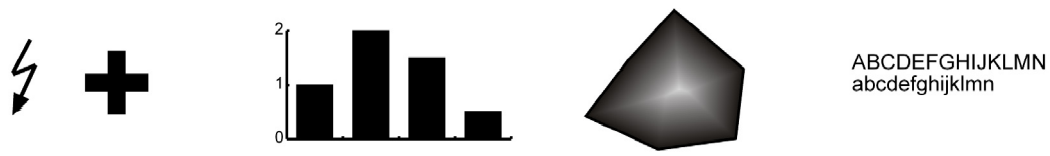


Abbildung 4-8: Beispiele zusammengesetzter Zeichen

- Die graphischen Strukturen, die sich durch das Zusammenspiel von Graphischen Elementen und Zusammengesetzten Zeichen ergeben, werden als *Graphische Gefüge* bezeichnet und bestimmen den Gesamteindruck einer Karte (Abbildung 4-9).

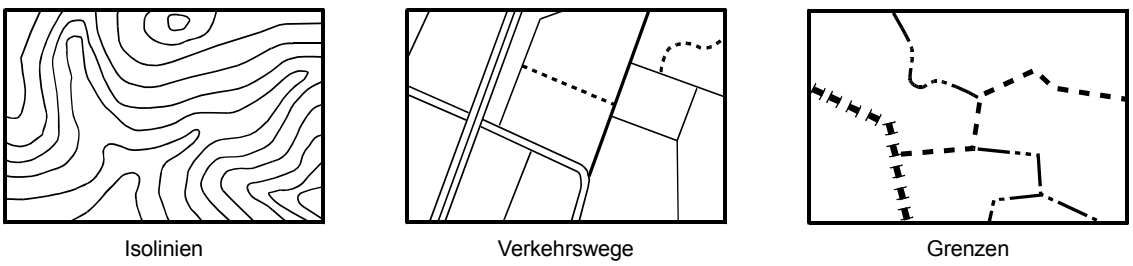


Abbildung 4-9: Beispiele linearer graphischer Gefüge (nach Hake et al. 2002)

Durch die genannten drei Stufen ist eine Kartengraphik nicht vollständig beschrieben. Weitere Ausdrucksformen ergeben sich in analog zu den Regeln nach Bertin durch den Einsatz visueller Variablen.

4.2.3 Erweiterungen der Visuellen Variablen

Wie bereits an anderer Stelle erwähnt, ist die Graphische Semiologie nach Bertin die weit verbreitete Grundlage für die Abbildung von Daten und Informationen auf graphische Darstellungen. Allerdings ist die Arbeit Bertins nicht unumstritten und wurde in den letzten Jahrzehnten in der Kartographie nicht nur aufgrund ihrer dogmatischen Natur kritisiert, sondern auch als unvollständig angesehen (MacEachren 1995).

Diese Kritik mag u.a. auch in den veränderten Rahmenbedingungen seit der Arbeit von Bertin begründet sein. So schränkt Bertin die Gültigkeit seines graphischen Systems explizit auf alles ein, was auf einem weißen Papierbogen dargestellt oder gedruckt werden kann, in einem mittleren Format mit einem Blick erfassbar und unter normalen Lichtverhältnissen in Leseentfernung zum Auge ist (Bertin 1974). Zur Berücksichtigung eventueller Farbfehlsichtigkeit eines Betrachters und aus Wirtschaftlichkeitsgründen sollte nur Farbe verwandt werden, falls es unumgänglich erscheint. Weiterhin werden dreidimensionale Abbildungen und bewegte Darstellungen ausgeschlossen (ebd.).

Offensichtlich sind diese Rahmenbedingungen durch den Einzug des Computers in die Kartenerstellung und –verbreitung nicht mehr zwingend gültig: Karten werden sehr häufig auf einem Bildschirm betrachtet und der Einsatz von Farbe ist ebenso schnell und kostengünstig möglich wie die Erstellung von bewegten Bildern.

Aufbauend auf dem Zeichensystem von Bertin wurden die visuellen Variablen in verschiedenen Arbeiten erweitert bzw. an die genannten neuen technischen Möglichkeiten und Ausdrucksformen angepasst. Im Folgenden wird in chronologischer Reihenfolge ein kurzer Überblick über einige Arbeiten im Kontext der Kartographie gegeben; eine ausführlichere Diskussion verschiedener Ansätze geben beispielsweise MacEachren (1995) oder Ellsiepen (2005). Weiterhin beschränkt sich diese Arbeit auf visuelle Variablen in ebener Darstellung, auf die Schilderung von Raumwahrnehmung oder auditiver Veränderlicher wird verzichtet.

Morrison (1974, zitiert nach MacEachren 1995) führt die Variable „Sättigung“ als Parameter der Farbe ein, zusammen mit „Farbton“ und „Helligkeit“ ist die Farbe dann in einem dreidimensionalen Farbraum beschreibbar (Ausführungen zur Farbbeschreibung vgl. Kapitel 1). Ein Raster beschreibt Morrison explizit durch drei Variablen: Die eigens eingeführte „Anordnung“ von Elementen, sowie „Muster“ (texture) und „Richtung“ nach Bertin. Die Dimensionen der Ebene sind zudem nicht Teil des Systems.

DiBiase et al. (1992) betrachten die Bewegung zur Darstellung des Raumes über die Zeit in animierten oder dynamischen Karten. Sie definieren drei dynamische Variablen: „Veränderungsdauer“, „-rate“ und „-reihenfolge“. MacEachren (1995) fügt diesen drei weitere hinzu: „Zeitpunkt des Beginns der Veränderung“, „Veränderungsfrequenz“ und die „Synchronisation“ zeitgleich ablaufender Änderungen.

Zur Darstellung von Datenunsicherheiten wird von MacEachren (1995) eine komplexe Variable „Klarheit“ eingeführt. Diese wird – analog zur Farbe – durch die drei unabhängigen Variablen „Schärfe“, „Auflösung“ und „Transparenz“ parametrisiert. „Schärfe“ deutet eine scharfe bzw. unscharfe Begrenzung eines Objekts an, die Darstellung ist vergleichbar mit verwischten Farbübergängen in Bildbearbeitungsprogrammen. Die „Auflösung“ soll die räumliche Genauigkeit widerspiegeln, der Gebrauch bzw. die Darstellung in digitalen Karten entspricht dem Auflösungsbegriff digitaler Rasterbilder. „Transparenz“ visualisiert unsichere Datenbereiche, indem diese Bereiche durch einen „Nebel“ wahrgenommen werden.

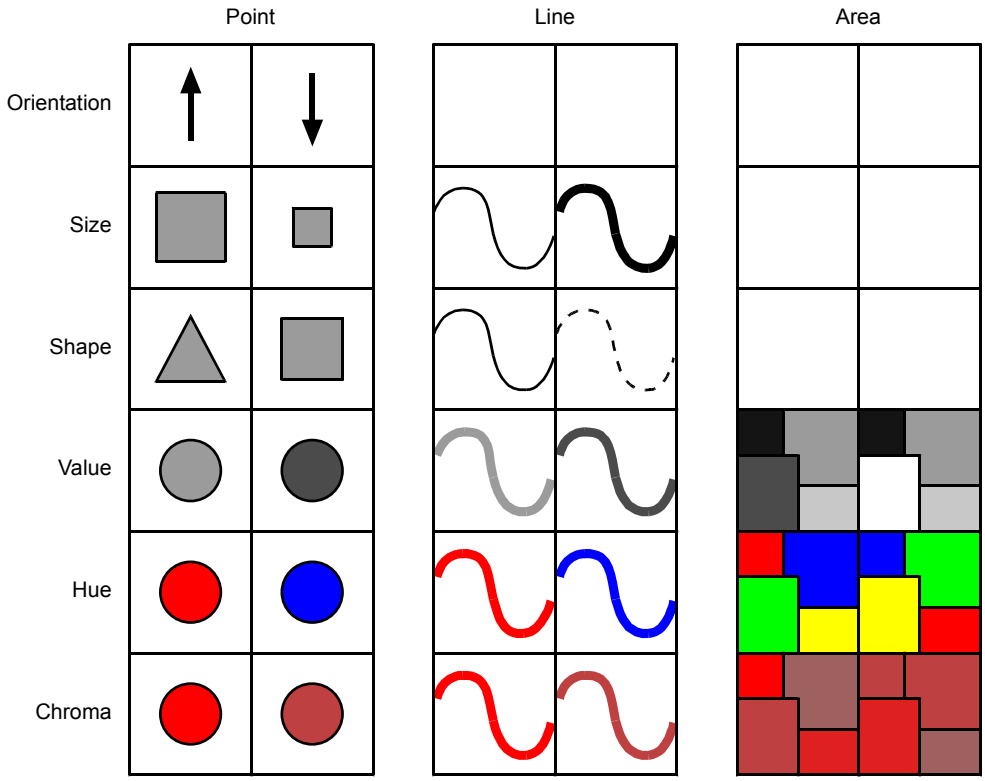


Abbildung 4-10: Primäre Variablen nach Robinson et al. (1995)

Robinson et al. (1995) beschreiben ein System, das die Variablen nach Morrison in primär und sekundär differenziert. Primäre Variablen sind „Richtung“, „Größe“, „Form“ und die drei „Dimensionen der Farbdarstellung“ (Abbildung 4-10). Wiederholungen von Graphischen Elementen und die Anwendung der primären Variablen auf diese Elemente ergeben ein Muster (pattern). Dieses Muster ist in den sekundären Variablen „Textur“, „Richtung“ und „Anordnung“ veränderbar (Abbildung 4-11).

Buziek (2001) verwirft die Variablen der „Klarheit“ nach MacEachren und führt sie auf Merkmale der Variablen nach Bertin zurück, beispielsweise setzt sich die „Auflösung“ aus „Fläche“ und „Füllung“ zusammen. Die „Veränderung“ betrachtet Buziek nicht nur unter dem Aspekt der Darstellung von Bewegung oder Prozessen, sondern auch als Mittel, die Aufmerksamkeit eines Adressaten gezielt zu steuern. Ergebnis ist ein graphisch-temporales Variablensystem in dem die „Veränderung“ keine graphische Variable, sondern ein Merkmal der graphischen Variablen nach Bertin darstellt. Die dynamischen Variablen nach DiBiase et al. und

MacEachren (siehe oben) fügen sich in dieses Schema als Eigenschaften des Merkmals „Veränderung“ ein.

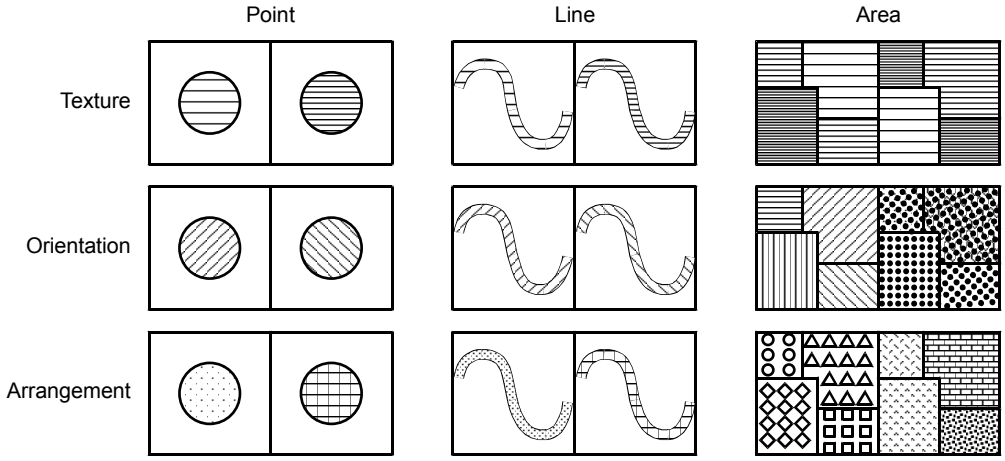


Abbildung 4-11: Sekundäre Variablen nach Robinson et al. (1995)

Ellsiepen (2005) greift die Differenzierung in primäre und sekundäre Variablen nach Robinson auf. Im Vergleich zu Robinson werden die primären Variablen um die Bertinsche Variable „Position“ ergänzt; die sekundären Variablen setzen sich aus „Transparenz“, „Veränderung“, „Textur“ und „Anordnung“ zusammen. „Transparenz“ wird durch Modifikation der (primären) Farbparameter erreicht und als Sonderfall der „Farbe“ aufgefasst. Die Veränderung wird in der von Buziek beschriebenen Weise betrachtet (siehe oben). „Anordnung“ und „Textur“ sind analog zu Robinson Teil eines „Musters“, im Gegensatz zu Robinson wird ein „Muster“ aber nicht mehr durch die „Richtung“ beschrieben.

4.2.4 Anwendbarkeit der visuellen Variablen

Im Rahmen der Beschreibung der Bertinschen Semiologie wurde bereits dargelegt, welche Gliederungsstufen der Wahrnehmung (Assoziativität, Selektivität, Ordnung und Quantität) jede der visuellen Variablen unterstützt. In diesem Abschnitt werden diese Beschreibungen noch einmal zusammenfassend dargestellt und um weitere Betrachtungen, die zum Teil aus den oben beschriebenen Erweiterungen der Variablen hervorgehen, ergänzt. Dabei ist zu beachten, dass in den genannten Arbeiten nicht die Unterstützung der Gliederungsstufen der Wahrnehmung betrachtet wird, sondern direkt nach der Skala der Daten (nach Bertin der Gliederungsstufen der Komponenten) eine Differenzierung in quantitative (numerische), ordinale und nominale Größen vorgenommen wird (vgl. Abschnitt 4.2.1).

Die Anwendbarkeit der Variablen nach Bertin, die sich mit dieser Unterscheidung ergibt, ist noch einmal in Abbildung 4-12 dargestellt (vgl. auch Abbildung 4-7 der Übersicht über die visuellen Variablen).

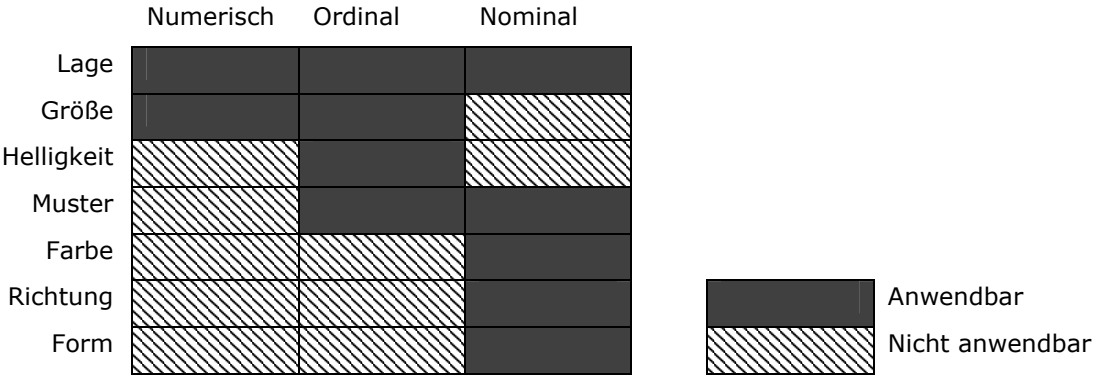


Abbildung 4-12: Eignung der graphischen Variablen nach Bertin zur Darstellung numerischer, ordinaler und nominaler Daten (Darstellung nach MacEachren 1995)

Die Variablen nach Morrison (1974, zitiert nach MacEachren 1995) wurden bereits im vorangegangenen Abschnitt beschrieben. Ihre Eignung für die Darstellung von Daten gibt Abbildung 4-13 wieder. Dabei ist zu beachten, dass Morrison ordinale und quantitative Größen zusammenfasst und damit lediglich zwischen ordinal und nominal differenziert. Für die Anwendbarkeit unterscheidet er eine dreistufige Skala.

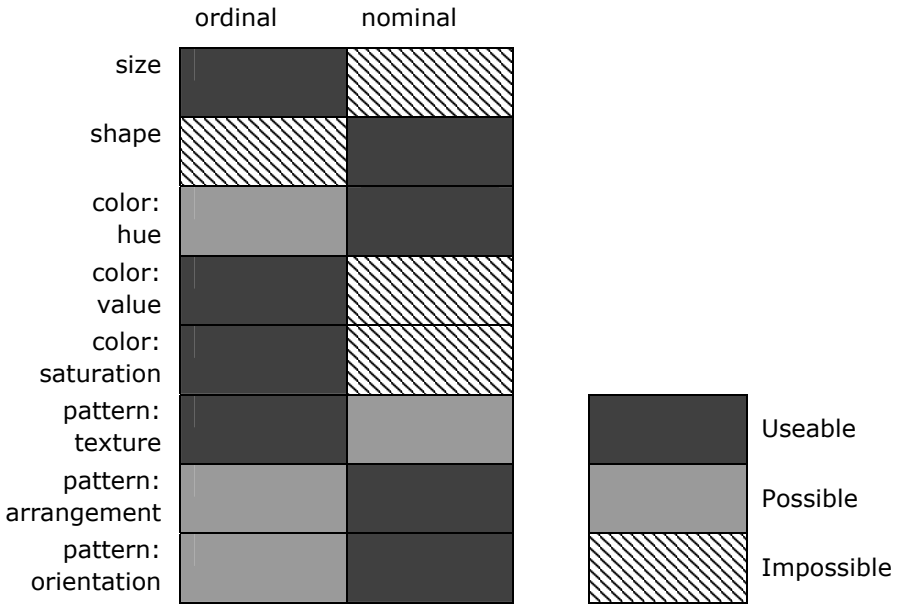


Abbildung 4-13: Eignung der acht Variablen nach Morrison (1974, zitiert nach MacEachren 1995) zur Darstellung ordinaler und nominaler Daten (Darstellung nach MacEachren 1995)

Ein Vergleich der in den Systemen von Morrison und Bertin identischen Variablen zeigt, dass die Einschätzung für „Größe“ und „Form“ übereinstimmt, ebenso entsprechen sich die Eignung von „Farbe“ bzw. „Farbton“ (color: hue) und „Richtung“ bzw. „Richtung eines Rasters“ (pattern: orientation) für nominale Größen. Die Anwendbarkeit auf ordinale Größen ist bei der „Helligkeit“ bzw. der „Helligkeit der Farbe“ (color: value) sowie dem „Muster“ bzw. dem „Muster eines Rasters“ (pattern: texture) gegeben.

MacEachren (1995) gibt ebenfalls ein umfassendes System von Variablen samt deren Anwendbarkeit an (Abbildung 4-14). Dieses umfasst neben Variablen nach Bertin und Morrison

auch die von MacEachren selbst eingeführte komplexe Variable der „Klarheit“ (vgl. vorangegangenen Abschnitt).

Die Buchstaben a) bis d) der Abbildung 4-14 spezifizieren einige der Variablen genauer (MacEachren 1995):

a) Mit den drei Variablen der „Klarheit“ können nicht mehr als zwei oder drei Kategorien unterschieden werden, ihre Anwendung ist zudem nicht empirisch getestet.

b) Die Darstellung einer Hierarchie durch Variation des „Farbtons“ erfordert die Auswahl von Farben, die eine Ordnung erkennbar machen (z.B. von gelb über orange zu rot).

c) Das „Muster eines Rasters“ eignet sich für die Unterscheidung von zwei oder drei Kategorien.

d) Die „Anordnung eines Rasters“ eignet sich, um einen Unterschied zwischen Kategorien redundant zu visualisieren.

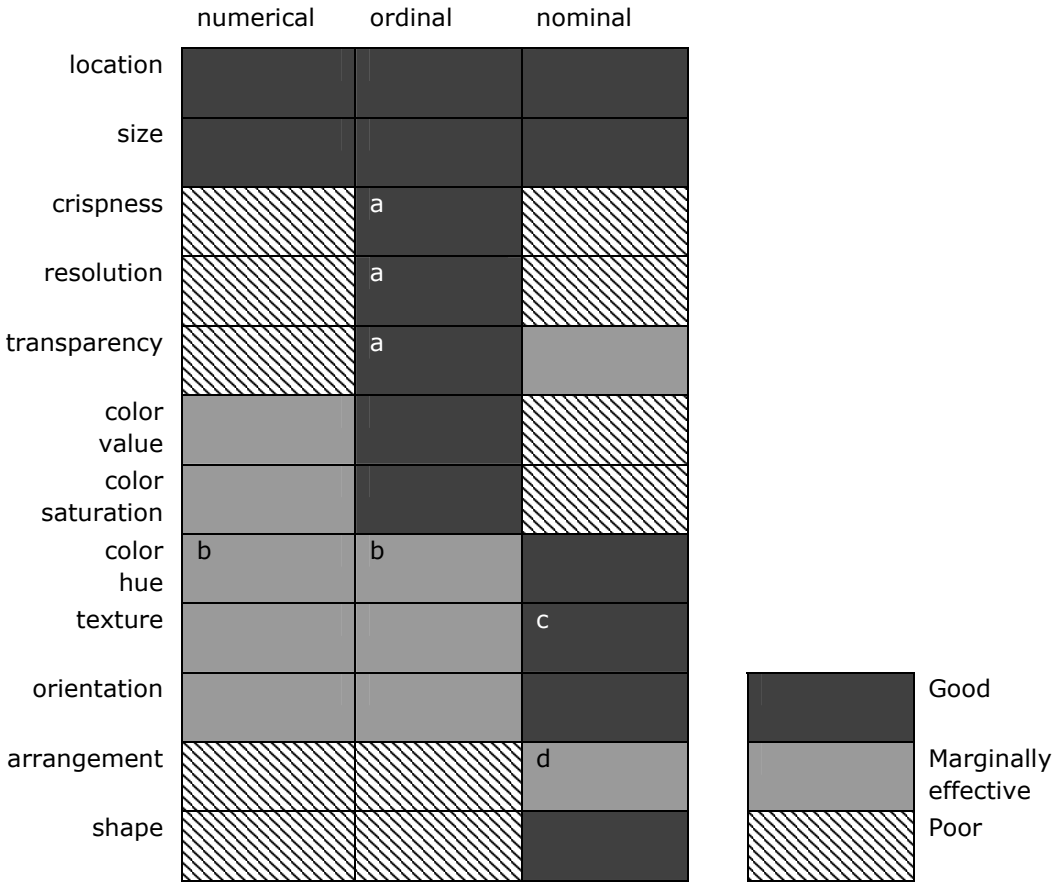


Abbildung 4-14: Eignung der Variablen nach MacEachren (1995, geändert) zur Darstellung numerischer, ordinaler und nominaler Daten; die Buchstaben a) bis d) geben weitere Details an, die Beschreibung findet sich im Text

Soweit vergleichbar, stimmt die Einschätzung nach MacEachren mit der von Morrison bzw. Bertin nur zum Teil überein. Auffällige Unterschiede liegen bei der Einschätzung von „Größe“, „Muster“ (texture) und „Anordnung“ für die Anwendung auf nominale Daten vor.

Außerhalb kartographischer Anwendungen betrachten Cleveland & McGill (1984) die Eignung visueller Variablen für die Entnahme quantitativer Informationen aus Diagrammen. Gemäß dieser Eignung werden die Variablen in eine Reihenfolge gebracht, die durch Tests bestätigt wird (Position entlang einer Skala, Länge, Richtung, Winkel, Fläche, Volumen, Krümmung, Helligkeit und Farbsättigung).

Mackinlay (1986) beschreibt, aufbauend auf den Arbeiten von Cleveland & McGill, die Effektivität einer Anzahl visueller Variablen ebenfalls durch Rankings. Er erweitert allerdings seine Betrachtungen über die Visualisierung quantitativer Informationen hinaus auf ordinale und nominale Daten (Abbildung 4-15). Weiterhin fügt er den von Cleveland & McGill betrachteten Variablen einige hinzu, verifiziert seine Erweiterungen aber nicht durch empirische Untersuchungen (Mackinlay 1986).

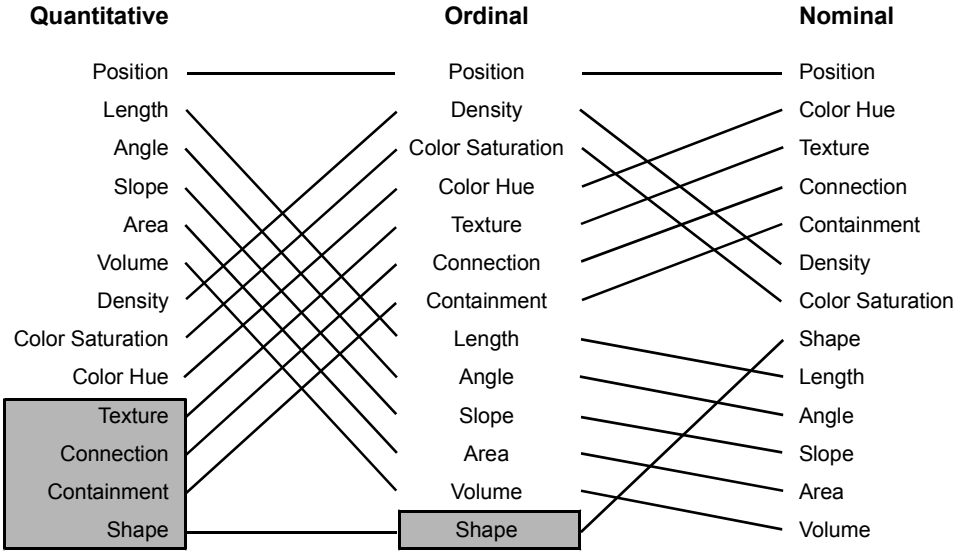


Abbildung 4-15: Rankings visueller Variablen für die Darstellung quantitativer, ordinaler und nominaler Daten (nach Mackinlay 1986); die grau hinterlegten Variablen sind für die jeweilige Art der Daten nicht relevant

Ein Teil der von Mackinlay angegebenen Variablen ist im kartographischen Kontext nicht betrachtet worden (z.B. Winkel, Länge, Umfassung), lässt sich aber offensichtlich auf die dort sehr viel weiter gefasste Variable „Position" zurückführen. Die Bedeutung bzw. Eignung der „Position" geht aus allen vorgestellten Bewertungen hervor.

Die übrigen Variablen nach Mackinlay lassen sich zum Teil mit den vorher beschriebenen Ergebnissen vergleichen (in Klammern jeweils die Übersetzung in der englischen Originalsprache, falls diese bei den verschiedenen Ansätzen abweicht):

- *Quantitativ*: Alle Ansätze kommen auf eine ähnliche Einschätzung der guten Eignung der „Größe" (size bzw. area) und einer weniger guten für „Helligkeit" (color value bzw. density), „Farbsättigung" und „Farbton". Widersprüche bestehen in der Anwendbarkeit der „Orientierung" (orientation bzw. slope) und des „Musters".
- *Ordinal*: Alle Ansätze stimmen in ihren Einschätzungen von „Helligkeit" (color value bzw. density), „Farbsättigung", „Farbton" und „Muster" überein, ein krasser Wider-

spruch liegt bei der „Größe“ (size bzw. area) vor, die Bertin, Morrison und MacEachren als gut geeignet einschätzen, Mackinlay dagegen als eine der ineffektivsten Variablen. Die „Form“ ist nach keinem der Anätze geeignet.

- *Nominal*: Alle Systeme kommen zur Einschätzung einer guten Eignung des „Farbtons“, MacEachren und Mackinlay stimmen weiterhin in Bezug auf die gute Eignung des „Musters“ überein. Widersprüche sind bei der „Helligkeit“ (color value bzw. density), der „Farbsättigung“ und der „Form“ sichtbar: „Helligkeit“ und „Farbsättigung“ eignen sich nach Morrison und MachEachren nicht, bei Mackinlay liegen sie im Mittelfeld, noch vor der „Form“, die sich bei erstgenannten sehr gut eignet. Noch deutlicher ist der Widerspruch bei der Bewertung der „Richtung“ (orientation bzw. slope).

Nowell (1997) bestimmt die Effektivität der Farbe, Form und Größe von Icons bei der Darstellung quantitativer und nominaler Daten auf Displays durch empirische Tests. Die Variable „Farbe“ wird dabei analog zu Bertin genutzt, d.h. es wird nicht zwischen „Farbton“, „Helligkeit“ und „Sättigung“ unterschieden. Die Variation der für die Tests genutzten Farben erfolgt primär durch den „Farbton“. Die empirischen Tests erfordern die Identifikation und Zählung bestimmter Icons; Ergebnis sind verschiedene Rankings in Bezug auf folgende Größen (vgl. auch Ausführungen zur Qualität und Eignung einer graphischen Darstellung im Abschnitt 4.1.4) (Nowell 1997):

- „*Time*“: mittlere Zeit zur Erledigung einer Aufgabe,
- „*Errors*“: Häufigkeit der Fehler,
- „*Ease*“: subjektive Einschätzung der Schwierigkeit, eine bestimmte Visualisierung zu nutzen,
- „*Likelihood*“: subjektive Einschätzung, ob ein Proband eine bestimmte Visualisierung zur Darstellung genutzt hätte,
- „*Discriminability*“: Unterscheidbarkeit der genutzten graphischen Elemente, gemessen wurde die mittlere Zeit zur Erledigung einer Aufgabe.

Die Rankings sind in Tabelle 4-1 für die Darstellung nominaler Daten und in Tabelle 4-2 für die Darstellung quantitativer Daten zusammengefasst.

	Time	Errors	Ease	Likelihood	Discriminability
Color	1	1	1	1	1
Shape	3	2	2	2	2
Size	2	3	3	3	3

Tabelle 4-1: Ranking der Variablen Farbe, Form und Größe bei der Darstellung nominaler Daten (vereinfacht nach Nowell 1997)

	Time	Errors	Ease	Likelihood	Discriminability
Color	1	1	1	1	1
Shape	2	2	2	2	2
Size	3	3	3	3	3

Tabelle 4-2: Ranking der Variablen Farbe, Form und Größe bei der Darstellung quantitativer Daten (vereinfacht nach Nowell 1997)

Aus den Ergebnissen schließt Nowell (1997), dass für die Wahl graphischer Elemente weniger die Art der Daten ausschlaggebend ist, als vielmehr

- die Aufgaben, die ein Nutzer damit zu lösen vermag: Das Identifizieren und Zählen von Objekten oder die Entnahme quantitativer Informationen wie bei Cleveland & McGill (1984).
- die Größe, nach der die Effektivität festgelegt wird (nach Zeit, Fehler usw.).

Eine abschließende Bewertung der Ausführungen dieses Abschnitts erfolgt im Abschnitt 4.4 im Rahmen der Ableitung derjenigen Variablen, die für die on demand erstellten Karten in dieser Arbeit benötigt werden.

4.2.5 Signaturen und Schrift

Als Teil des dreistufigen Aufbaus des Kartographischen Zeichensystems wurden als Zusammengesetzte Zeichen Signatur und Schrift genannt, die beide wesentlicher Bestandteil von Karten sind.

Signaturen stellen eine kartenspezifische Kurzschrift dar (Hake et al. 2002); nach ihrer Gestalt lassen sich unterscheiden (ebd.):

- *Bildhafte* Signaturen stellen Objekte als Grundriss-, Aufriss- oder Schrägbilder schematisch bis individuell dar.
- *Symbolhafte* Darstellungen kennzeichnen Objekte als abstrahierte Sinnbilder.
- *Geometrische* (abstrakte) Signaturen können in Form einfacher Figuren (Kreis, Dreieck, …), Linienunterbrechungen oder Schraffuren auftreten.
- *Buchstaben* können als Abkürzungen dienen, *Ziffern* und *Zahlen* geben einen Wert (Index, Schlüssel oder Verhältnis) an, *Unterstreichungen* differenzieren Qualitäten (z.B. Hauptstädte).

Die Repräsentation einer Signatur kann punkt- (lokaler), linien- (linearer) oder flächenhafter Natur sein; durch Anwendung der graphischen Variablen werden Qualitäten und Quantitäten ausgedrückt (Hake et al. 2002). Da Signaturen nicht allgemeingültig in ihrer Semantik festgelegt sind, ist immer eine Aufzählung der Zeichenbedeutungen in einer Legende erforderlich.

Die Schrift wirkt in Karten durch die Angabe von Namen, Abkürzungen und Zahlen in der Hauptsache als erläuterndes Element, die geometrischen Aussagemöglichkeiten beschränken sich weitgehend auf eine Raumtreue (Hake et al. 2002). Eine Schrift wird durch verschiedene Eigenschaften wie Schriftart oder Schriftgröße beschrieben, eine Zusammenstellung geben Hake et al. (2002) wieder; vertiefende Ausführungen finden sich im Schrifttum zur Typographie (bspw. Ambrose & Harris 2007).

Schriften differenzieren Objekte nach Qualitäten durch die Angabe von Eigen- oder Kategorienamen und die Variation nach Form oder Farben; die Differenzierung nach Quantitäten erfolgt durch die Angabe von Zahlenwerten oder die Variation der Schriftgröße (Hake et al. 2002).

Bei der Bezeichnung von Objekten ist weiterhin die Platzierung der Schrift von Bedeutung. Neben einer klaren Zuordnung zum jeweiligen Objekt lassen sich als Regeln angeben (ebd.):

- *Lokale Objekte*: Rechts und etwas höher als das Objekt in waagerechter Ausrichtung
- *Lineare Objekte*: parallel zur Linienführung
- *Flächenhafte Objekte*: Waagerecht oder in Richtung der größten Ausdehnung.

Die genannten Regeln sind offensichtlich nur dann unmittelbar anwendbar, wenn ausreichend freie Kartenfläche zur Verfügung steht. Für die andernfalls erforderlichen, weitergehenden Regeln, wird auf vertiefende Arbeiten verwiesen (Ellsiepen 2002, Petzold 2003).

4.3 Bedingungen der Kartengestaltung

Für die Abbildung von Daten oder Informationen auf graphische Elemente und deren Eigenschaften sind neben einem Zeichensystem weitere Bedingungen von Bedeutung. Im Sinne der Kommunikationstheorie stellen diese Bedingungen bestimmte Anforderungen an die Codierung. Von Bedeutung ist zum einen die Beachtung einer korrekten Syntax, zum anderen die Reduktion der Störungen, die auf den Kanal „Karte“ wirken, d.h. Ziel ist ein möglichst gutes Signal-Rausch-Verhältnis. In ihrer Auswirkung schränken diese Bedingungen den Gebrauch der graphischen Variablen und damit den Umfang des Zeichensystems ein. Im Einzelnen nennen Hake et al. (2002):

- Maßstäblichkeit und Grundrissdarstellung,
- Wahl der Gestaltungsmittel entsprechend der Semantik,
- Lesbarkeit des einzelnen Kartenzeichens,
- Lesbarkeit der Kartenzeichen in Bezug auf ihre gegenseitigen Beziehungen.

Maßstäblichkeit und Grundrissdarstellung folgen offensichtlich unmittelbar aus der Definition der Karte. Alle Zeichen sollten in ihrer geometrischen Ausprägung und Anordnung der Lage und Größe in der Realität entsprechen, nur so kommuniziert die Karte räumliche Gegebenheiten und Zusammenhänge.

Die *Wahl der Gestaltungsmittel* bzw. die oben beschriebene graphische Variation der Zeichen erfordert bei der Abbildung von Daten auf graphische Elemente die Beachtung der Semantik, insbesondere gelten folgende Grundsätze (Hake et al. 2002):

- Gleiches gleich, Ungleiches ungleich darstellen,
- Wichtiges erhalten, Unwichtiges fortlassen,
- Typisches Betonen, Untypisches abschwächen.

Die *Lesbarkeit des einzelnen Kartenzeichens* wird besonders durch Minimaldimension und kartographische Mindestgrößen bestimmt. Erstere gibt die Größe von Zeichen und Abständen an, die dem menschlichen Auge noch die Lesbarkeit erlaubt, letztere die Größe, die graphische Elemente oder Abstände gemäß kartographischer Regeln bzw. Zeichenvorschriften mindestens besitzen müssen (vgl. Malic 1998, Hake et al. 2002). Die Minimaldimension für die

Darstellung in Bildschirmkarten ist abhängig von der Auflösung des Ausgabemediums und damit von der Größe eines Pixels des jeweiligen Bildschirms (Hake et al. 2002). Die durchschnittlichen Bildpunktgrößen liegen z.B. zwischen 0,19 und 0,39 mm für Röhrenbildschirme mit 17'' Bildschirmdiagonale und zwischen 0,26 und 0,3 mm für LCD-Desktop-Monitore mit einer Diagonalen von 15'' bis 18'' (Brunner 2001). Tabelle 4-3 gibt beispielhaft die Minimaldimension für einen Röhrenbildschirm mit einer Auflösung von 1024x768 Pixel bei 17'' Bildschirmdiagonale wieder (Bildpunktgröße: 0,3 mm).

Element	Minimaldimension
Strichstärke	0,3 mm
Linienabstand	0,5 mm
Flächenabstand	0,6 mm
Formerkennbarkeit: Dreieck	1,0 mm
Formerkennbarkeit: Quadrat	1,2 mm
Formerkennbarkeit: Kreis	1,5 mm
Schrifthöhe (Helvetica)	8 pt (2,0 mm)

Tabelle 4-3: Minimaldimensionen in Bildschirmkarten für einen 17'' Röhrenbildschirm mit einer Auflösung von 1024x768 Pixeln (nach Malic 1998)

Die Angabe von Minimaldimensionen erfolgt meist für Schwarz-Weiß-Darstellungen. Höhere Werte müssen beispielsweise berücksichtigt werden, falls der Kontrast des Objekts vor einem Hintergrund geringer ist oder falls feine Farbabstufungen erkennbar sein sollen (vgl. Hake et al. 2002).

Weitere Ausführungen zu Besonderheiten der graphischen Darstellung in Bildschirmkarten finden sich beispielsweise in Malic (1998), Brunner (2001) und Neudeck (2001).

Für die *Lesbarkeit der Kartenzeichen in Bezug auf ihre gegenseitigen Beziehungen* nennen Hake et al. (2002) verschiedene Regeln:

- Für eine ausreichende *graphische Differenzierung* sollten die graphischen Variablen in möglichst vielfältiger Weise genutzt werden.
- Aus Platzgründen erlaubt eine Graphik lediglich die Darstellung einer gewissen Anzahl von Objekten je Flächeneinheit. Diese *graphische Dichte* sollte nicht zu groß sein.
- Durch ausreichende *Kontrast- und Objekttrennung* sollten sich Objekte deutlich vom Hintergrund abheben. Dies wird beispielsweise durch die Verwendung heller Hintergründe, kräftiger Objektfarben oder Freistellung erreicht.

Besonders der letzte Aspekt der Kontrast- und Objekttrennung liegt im engeren Fokus dieser Arbeit.

4.4 Anwendung der Gestaltungsregeln in dieser Arbeit

In den letzten beiden Abschnitten wurden zum einen die Möglichkeiten der graphischen Darstellung von Objekten, zum anderen verschiedene Einschränkungen, die auf die Anwendung

der graphischen Mittel wirken, vorgestellt. Diese theoretischen Ansätze werden nun in einem Schema, nach dem in dieser Arbeit die graphische Gestaltung on demand erstellter Karten erfolgt, zusammengefasst. Ziel ist allerdings kein allgemeingültiges Variablensystem oder Verfahren zur Gestaltung von Karten jeglicher Art, sondern die Identifizierung der Möglichkeiten, die die Ziele dieser Arbeit möglichst gut unterstützen.

4.4.1 Vorbemerkungen

In den bisherigen Ausführungen (Anwendungsszenarien in der Einleitung, Art der Daten im Abschnitt 2.3.2) wurde bereits die Art bzw. der Inhalt der hier betrachteten Karten eingegrenzt: Es sollen

- räumliche Objekte mit punkt-, linien- und flächenhafter Geometrie zu einer (nicht zwingend vollständigen) topographischen Karte integriert werden,
- vorhandene topographische Karten[35] um räumliche Objekte mit punkt-, linien- und flächenhafter Geometrie angereichert werden.

Dazu ist jeweils die graphische Ausprägung der darzustellenden Objekte durch Anwendung der visuellen Variablen festzulegen. Bevor deren Einsatz in dieser Arbeit systematisch zusammengefasst wird, sind noch einige präzisierende Erläuterungen von Bedeutung.

Die Literatur zur kartographischen Gestaltung betrachtet primär die Erstellung thematischer Karten. Die Unterscheidung von Daten nach ihrer Skala (nominal, ordinal, quantitativ) bezieht sich dann meist auf die Differenzierung oder Anordnung von Daten nach Eigenschaften innerhalb *einer Klasse*. In dieser Arbeit ist dagegen auch die Differenzierung von Objekten *verschiedener Klassen* von Bedeutung. Gemäß dieser Unterscheidung wird für einen klareren sprachlichen Ausdruck im Folgenden zwischen Visuellen *Variablen* und *Konstanten* unterschieden:

- Variablen differenzieren oder ordnen (nominal, ordinal, quantitativ) Objekte innerhalb einer Klasse nach einer Eigenschaft, differenzieren diese Klasse bzw. deren Objekte aber auch von den Objekten anderer Klassen.
- Konstanten sorgen dafür, dass Objekte einer Karte überhaupt eine graphische Ausprägung besitzen und stellen alle Objekte innerhalb einer Klasse gleich da. Konstanten differenzieren allerdings die Objekte einer Klasse von denen anderer Klassen.

Im Folgenden ist die Unterscheidung von Klassen ein primäres, die Unterscheidung innerhalb einer Klasse sekundäres Ziel.

Aus den Ausführungen des Abschnitts 4.2.4 ging bereits die Bedeutung der Farbe bzw. ihrer Dimensionen für die graphische Darstellung hervor. Die genannten raumbezogenen Objekte

[35] Unter einer topographischen Karte versteht die Internationale Kartographische Vereinigung (1973) eine „Karte, in der Situation, Gewässer, Geländeformen, Bodenbewachsung und eine Reihe sonstiger zur allgemeinen Orientierung notwendiger oder ausgezeichneter Erscheinungen den Hauptgegenstand bilden und durch Kartenbeschriftung eingehend erläutert sind“ (zitiert nach Hake et al. 2002).

verschiedener Thematiken können offensichtlich am besten durch den Einsatz von Farbe visuell differenzierbar gemacht werden. Für diese Arbeit heißt dies, dass die primäre Darstellung durch Farben erfolgen muss, die für das menschliche Auge möglichst gut unterscheidbar sind bzw. einen möglichst großen Kontrast besitzen (Vertiefende Ausführungen zur Unterscheidbarkeit von Farben finden sich im weiteren Verlauf dieser Arbeit).

4.4.2 Anwendung der graphischen Variablen

Es wird nun die konkrete Anwendung der im Abschnitt 4.2 vorgestellten graphischen Variablen in dieser Arbeit aufgezeigt. Dafür werden diejenigen Variablen betrachtet, die eine Schnittmenge der oben aufgeführten verschiedenen Systeme bilden, einige davon können allerdings direkt ausgeschlossen werden:

Position: Da ausschließlich Objekte der Realwelt betrachtet werden, deren Lage im Raum festgelegt ist, steht diese Variable nicht zur Verfügung.

Größe: Die Größe dient als Variable dem Ausdruck einer Ordnung oder Quantität. Als Variable und Konstante ist sie nach unten durch die Mindestgröße festgelegt, nach oben sollte sie so gewählt sein, dass sich Objekte möglichst wenig verdecken.

Form: Die Form als Variable wird direkt nur auf punkthafte Objekte angewandt. Für Linien und Flächen ist sie Teil Zusammengesetzter Zeichen.

Farbe (Farbton, Sättigung, Helligkeit): Die Bedeutung der Farbe für jegliche Art der Darstellung wurde bereits betont.

Richtung: Die Variation der Richtung wird direkt nur für punkthafte Objekte angenommen. Bei Flächen kann die Richtung auf ein Muster wirken.

Anordnung und Textur (Muster): Diese Variablen des Musters werden lediglich für die Füllung einer Fläche angenommen.

Klarheit: Es werden Objekte mit realer räumlicher Ausdehnung betrachtet, die Darstellung von Unsicherheit (vgl. Abschnitt 4.2.3) ist nicht erforderlich.

Bewegung: Ziel dieser Arbeit sind statische Darstellungen, die Nutzung von Bewegung wird nicht betrachtet.

Die Abbildungen 4-16 bis 4-18 stellen die Anwendung der Variablen auf die graphischen Grundelemente systematisch zusammen. Dabei wird gemäß der Differenzierung im vorangegangenen Abschnitt zwischen variabler und konstanter Darstellung unterschieden. Im einfachsten Fall wird immer das Element selber dargestellt, alle Darstellungen, die über das Grundelement hinausgehen, werden als Zusammengesetzte Zeichen interpretiert, die wiederum bestimmte Darstellungseigenschaften besitzen.

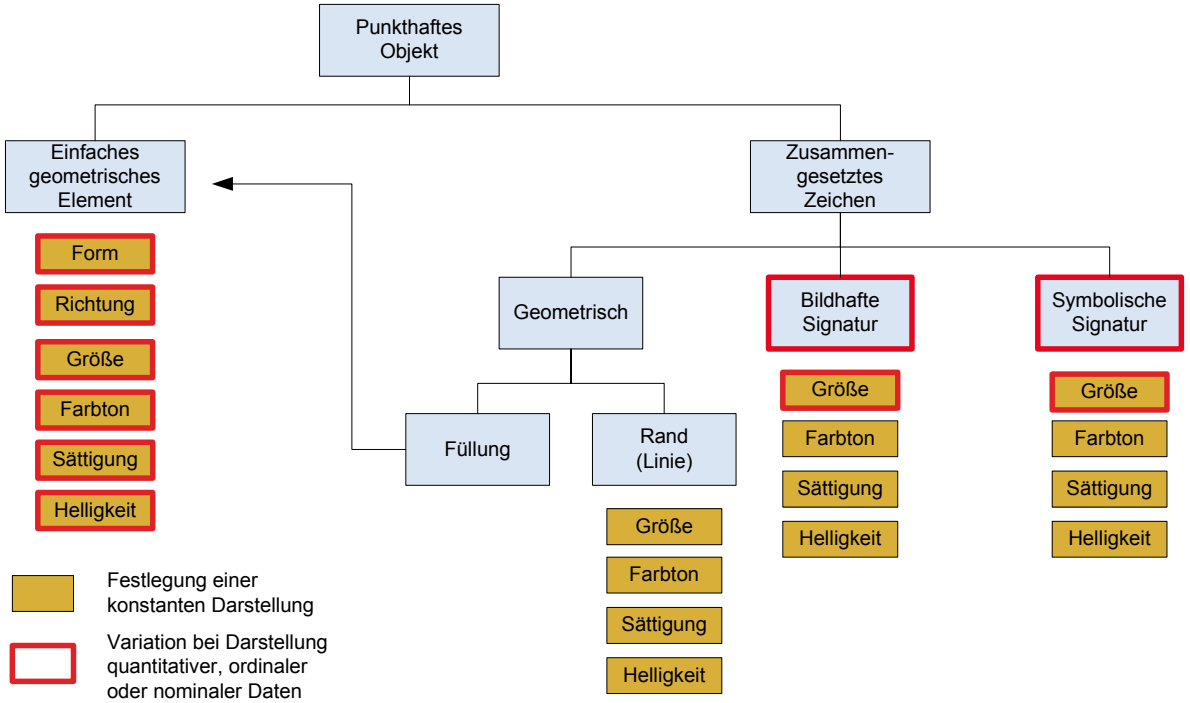


Abbildung 4-16: Darstellungsmöglichkeiten für punkthafte Objekte in dieser Arbeit

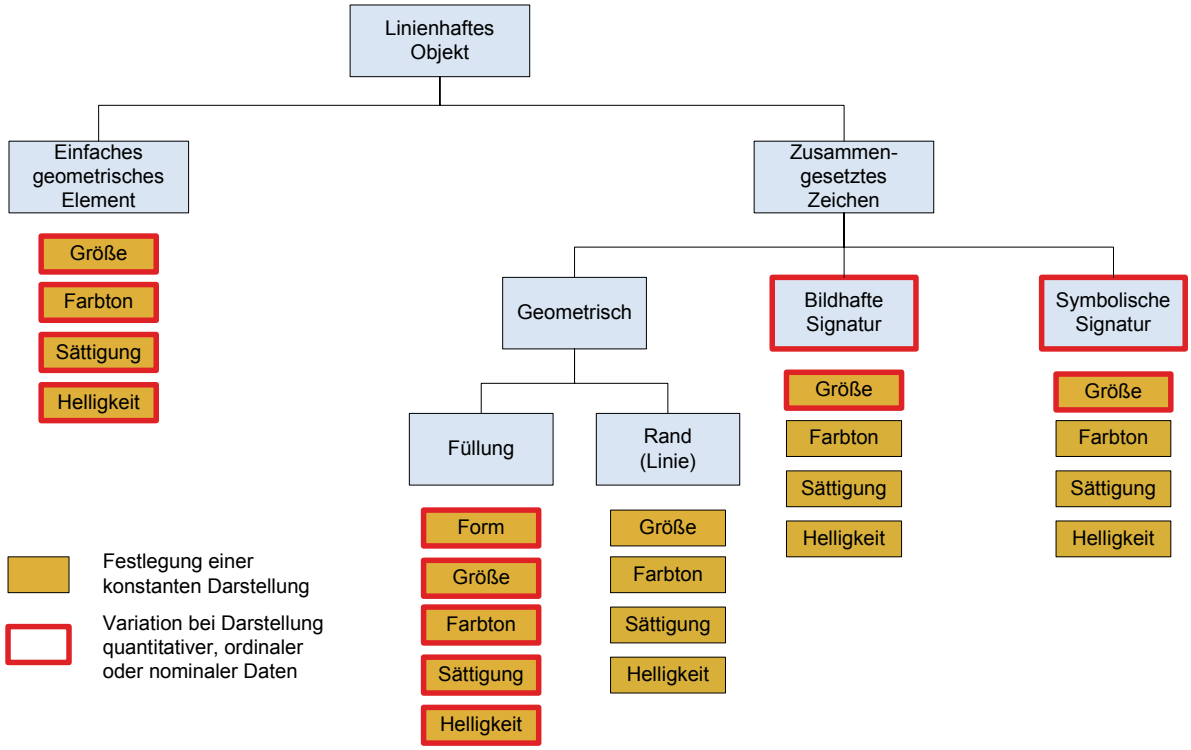


Abbildung 4-17: Darstellungsmöglichkeiten für linienhafte Objekte in dieser Arbeit

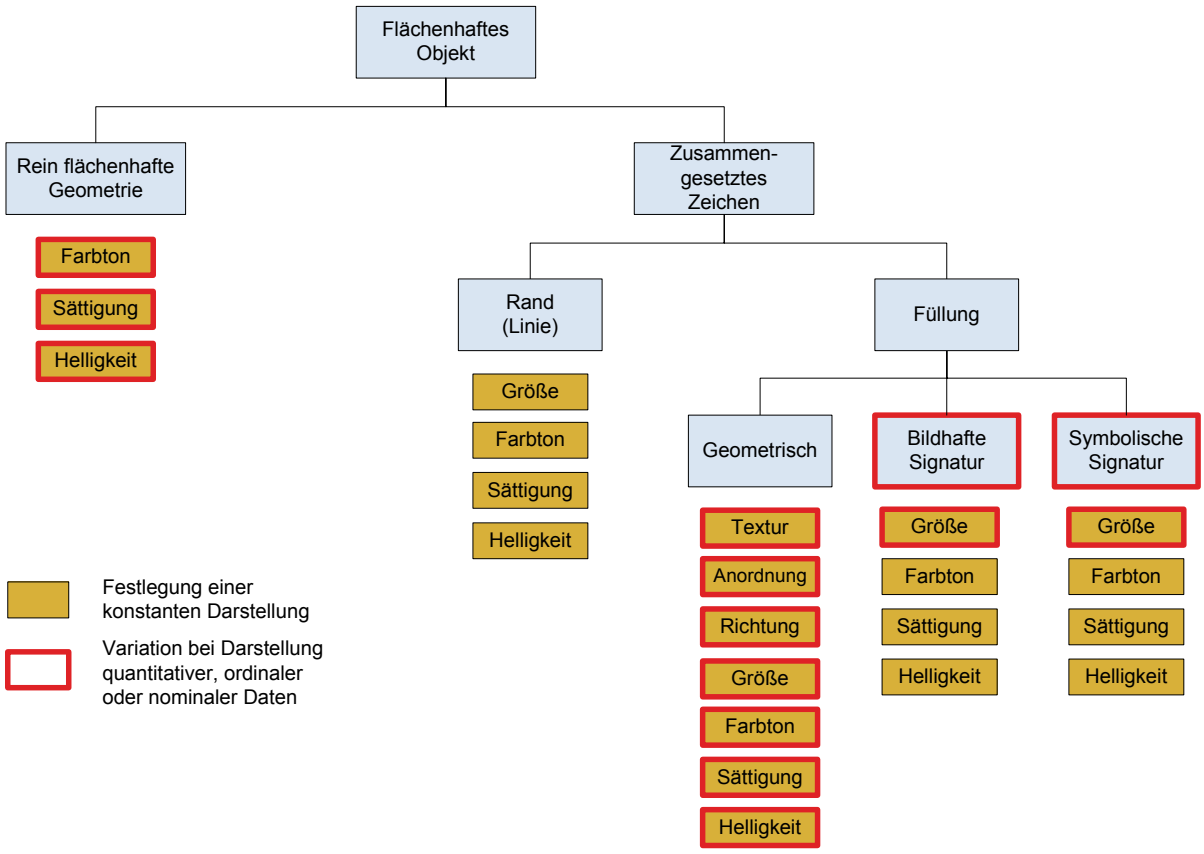


Abbildung 4-18: Darstellungsmöglichkeiten für flächenhafte Objekte in dieser Arbeit

Die Beschreibung einer systematischen Vorgehensweise zur konkreten Anwendung der Darstellungsmittel wird im Kapitel 8 gegeben.

4.5 Zusammenfassung

Im vorliegenden Kapitel wurden die Darstellungsmöglichkeiten durch graphische Elemente und Variablen vorgestellt und die Ergebnisse verschiedener Arbeiten, die die Eignung der Variablen zur Visualisierung nominaler, ordinaler und quantitativer Daten betrachtet haben, zusammengefasst. Daraus ging hervor, dass die Farbe bzw. ihre drei Dimensionen Farbton, Sättigung und Helligkeit insgesamt als vorrangige – effektivste – visuelle Gestaltungsmittel anzusehen sind.

Weiterhin wurde die Anwendung visueller Variablen auf punkt-, linien- und flächenhafte Objekte für die Erstellung der ad-hoc-Karten dieser Arbeit systematisch zusammengestellt. Daraus geht ebenfalls die Bedeutung der Farbdimensionen hervor. Im folgenden Kapitel werden deshalb die für den Einsatz von Farbe und zur Umsetzung prägnanter Darstellungen wesentlichen Aspekte der Wahrnehmung und zahlenmäßigen Beschreibung von Farbe vertiefend ausgeführt.

5 Nutzung von Farbe

Aus den Ausführungen des letzten Kapitels ging die Farbe als primäre visuelle Variable zur Erstellung von ad-hoc-Karten hervor. Im Vergleich zu anderen, ebenfalls vorgestellten, Variablen ist die Nutzung von Farbe allerdings ungleich komplexer; dies wird schon anhand einiger Möglichkeiten und Rahmenbedingungen von Farbdarstellungen deutlich:

- *Größe des Wertebereichs:* Gängige Bildschirme geben mehrere Millionen Farben wieder, das normalsichtige menschliche Auge kann ebenso viele Farbnuancen unterscheiden (Schumann & Müller 2000).
- *Äußere Einflüsse:* Die Wahrnehmung einer Farbe hängt immer von Umgebungsbedingungen (z.B. benachbarte Farben, Beleuchtung) ab (vgl. Abschnitt 5.1.2).
- *Geräteabhängigkeit*: Farben werden durch verschiedene technische Geräte unterschiedlich wiedergegeben (vgl. Abschnitt 5.6.1).
- *Sehschwächen:* Bei der Anwendung von Farben wird in der Regel von normalsichtigen Adressaten ausgegangen, allerdings sind annähernd 8 % der männlichen und 0,4 % der weiblichen Bevölkerung (Schläpfer 1993) in ihren Möglichkeiten Farben zu sehen bzw. zu unterscheiden eingeschränkt (vgl. Abschnitt 5.6.2).

Diese genannten Punkte verdeutlichen, dass ein korrekter und zweckmäßiger Einsatz von Farbe grundlegende Kenntnisse der Farbwahrnehmung und -theorie voraussetzt. Dabei wird sich zeigen, dass die Farbwahrnehmung zunächst ein subjektiver Vorgang ist. Die mathematische Modellierung der Farbauswahl im Rahmen eines Optimierungsmodells erfordert dagegen eine objektivierte Beschreibung durch ein geeignetes Bezugssystem (Farbraum). Auf diesem Bezugssystem wird eine Metrik benötigt, die eine Modellierung der effektiven Kommunikation durch die Bestimmung gut unterscheidbarer Farben ermöglicht.

Diese Anforderung an ein Bezugssystem macht für diese Arbeit die Nutzung verschiedener Farbräume notwendig. Der RGB- und der CMYK-Farbraum werden für die Reproduktion von Farben auf Bildschirmen und im Druck benötigt, bieten allerdings keine geeignete Metrik. Für eine solche, der menschlichen Wahrnehmung entsprechende, Metrik stehen der CIELUV- und CIELAB-Farbraum zur Verfügung. Hauptziel dieses Kapitels ist es, das Problem der Prägnanz in den letztgenannten Räumen beschreibbar zu machen.

Abschnitt 5.1 fasst nach der Definition des Begriffes Farbe wesentliche Aspekte der Entstehung und Wahrnehmung des Farbeindrucks zusammen, Abschnitt 5.2 führt in die Grundlagen der Farbmetrik ein. Im Abschnitt 5.3 werden nach dem Normvalenzsystem, dem international vereinbarten Basissystem für die Beschreibung von Farben, die angesprochenen Farbräume vorgestellt. Abschnitt 5.4 skizziert mit dem Munsell-System ein bekanntes Farbordnungssystem, Abschnitt 5.5 fasst wesentliche Aspekte der Farbwahl in Karten zusammen. Im Abschnitt 5.6 wird die Personalisierung der Farbdarstellung, d.h. die Anpassung auf Seheigen-

schaften und Anzeigemedium eines Nutzers, beschrieben, bevor im Abschnitt 5.7 eine kurze Zusammenfassung gegeben wird.

5.1 Grundlagen der Farbwahrnehmung

Die Tatsache, dass Farbe für einen normalsichtigen Menschen ein permanenter Bestandteil der Erfahrung seiner Umwelt ist, verschleiert die komplexen Abläufe bei der Entstehung und Wahrnehmung von Farbeindrücken. In diesem Abschnitt werden deshalb zunächst einige Grundlagen der Farbwahrnehmung zusammengefasst. Die Ausführungen beschränken sich dabei weitgehend auf wesentliches Vokabular zur Farbbeschreibung, die Entstehung eines Farbreizes, sowie auf verschiedene Effekte, die die Wahrnehmung beeinflussen können. Die detaillierten physiologischen Vorgänge im visuellen System des Menschen, die für ein vertieftes Verständnis der Farbtheorie bedeutsam sind, werden für die folgenden Ausführungen nicht zwingend benötigt. Ausführliche Darstellungen finden sich beispielsweise in Lang (1995) oder Purves & Lotto (2003).

5.1.1 Farbe

Der Begriff Farbe wird im Alltag in verschiedenen Bedeutungen gebraucht, so werden sowohl Farbstoffe als auch Farberscheinungen gemeinhin als Farbe bezeichnet (Schumann & Müller 2000). Charakteristisch für den verbalen Ausdruck von Farbe ist die Zuordnung zu Gegenständen unserer Umwelt („Mein Auto ist rot“) (Lang 1995). Eine Definition gibt der DIN-Standard 5033 „Farbmessung“. Dort wird Farbe als eine Gesichtsempfindung, als Sinneseindruck, der durch das Auge vermittelt wird, beschrieben (DIN 1979):

> *„Farbe im Sinne dieser Norm ist ein durch das Auge vermittelter Sinneseindruck, also eine Gesichtsempfindung. Die Farbe ist diejenige Gesichtsempfindung eines dem Auge strukturlos erscheinenden Teils des Gesichtsfeldes, durch die sich dieser Teil bei einäugiger Beobachtung mit unbewegtem Auge von einem gleichzeitig gesehenen, ebenfalls strukturlosen angrenzenden Bezirk allein unterscheiden kann.“*

Farbe wird also einem Betrachter durch das Auge vermittelt und ist keine Eigenschaft von Objekten oder Oberflächen. Der Sinneseindruck wird ausgelöst durch Licht (sichtbare elektromagnetische Strahlung, Abschnitt 5.1.3.1), das ins Auge eindringt. Diese Strahlung wird auch als *Farbreiz*, das Ergebnis der Farbwahrnehmung im Auge als *Farbvalenz* bezeichnet (Schläpfer 1993). Der eigentliche Sinneseindruck, die *Farbempfindung*, kommt schließlich im Gehirn zustande (ebd.). Die Beschreibung dieser drei Stationen der Farbwahrnehmung ist durch die Gesetze der Physik (Farbreiz), Physiologie (Farbvalenz) und Psychologie (Farbempfindung) möglich (ebd.). Während der Farbreiz damit als physikalische Größe messbar ist, ist die Farbempfindung als psychologische Größe Meßmethoden nicht unmittelbar zugänglich. Ziel der Farbwissenschaft bzw. dem Teilgebiet der Farbmetrik ist es, trotz dieser Schwierigkeit, einen Zusammenhang zwischen einem Farbreiz und der durch diesen ausgelösten Farbempfindung herzustellen (vgl. Lang 1995).

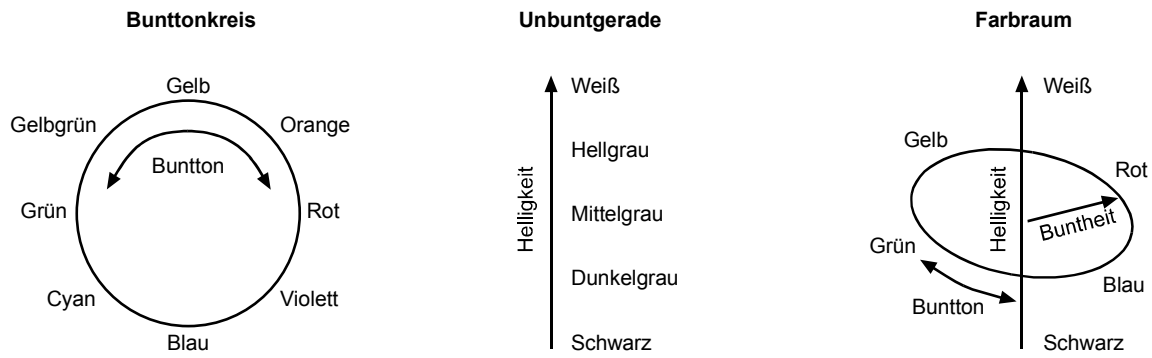


Abbildung 5-1: Anordnung von Farben auf dem Bunttonkreis, der Unbuntgeraden und im Farbraum (nach Lang 1995)

Eine Farbempfindung ist charakterisierbar als bunt oder chromatisch und unbunt oder achromatisch (Schumann & Müller 2000). Chromatische Farben lassen sich in einem sogenannten Farb- oder Bunttonkreis anordnen, unbunte Farben in ihrem Helligkeitsverlauf von Schwarz nach Weiß auf einer Geraden (Lang 1995). Die Vereinigung dieser beiden Darstellungen führt zu einer dreidimensionalen Darstellung, einem sogenannten Farbraum. In diesem lassen sich Farben durch die Begriffe *Buntton (Farbton)*, *Helligkeit* und *Buntheit* bzw. *Sättigung* beschreiben (vgl. Lang 1995). Abbildung 5-1 gibt einen Überblick über die Zusammenhänge.

5.1.2 Physiologische und psychologische Wahrnehmungseinflüsse

Die Farbwahrnehmung ist für das normalsichtige menschliche Auge nicht absolut, sondern abhängig von bestimmten Rahmen- und Beobachtungsbedingungen. Aus wahrnehmungspsychologischer Sicht können bei der Betrachtung von Farben verschiedene Effekte oder optische Täuschungen auftreten, die die Farbempfindung beeinflussen.

Sukzessivkontrast

Das Auftreten farbiger Nachbilder im Auge wird als Sukzessivkontrast bezeichnet (Lang 1995). Diese Nachbilder entstehen beim Betrachten einer farbigen Fläche als Gegenbild in den Komplementärfarben zur fixierten Farbe (Schumann & Müller 2000). Abbildung 5-2 zeigt ein Beispiel: Wenn ein Betrachter einige Zeit den Mittelpunkt des linken, farbigen Quadrats fixiert, erscheinen beim anschließenden Fixieren des Mittelpunkts des rechten Quadrats die jeweiligen Komplementärfarben der Teilflächen des ersten Quadrats.

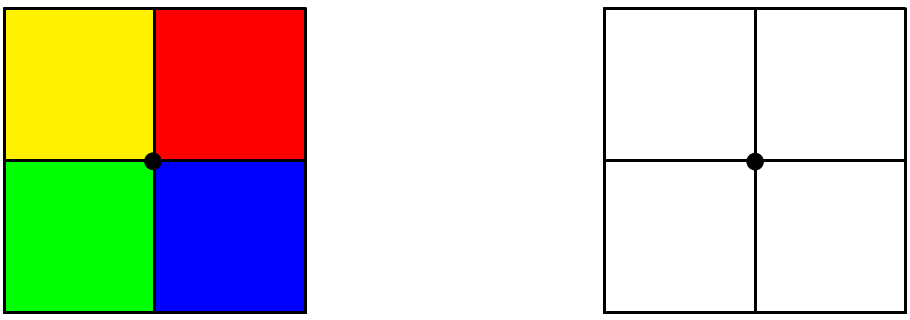


Abbildung 5-2: Wirkung des Sukzessivkontrasts (nach Schumann & Müller 2000)

Der Sukzessivkontrast entsteht durch eine lokale Veränderung betreffender Netzhautstellen des Auges; diese Veränderung benötigt eine gewisse Zeit, um sich auf- bzw. abzubauen (Lang 1995). Aufgrund des Sukzessivkontrastes wird empfohlen, die Anzahl von Farben im Verlauf einer Visualisierung möglichst gering und konstant zu halten (Schumann & Müller 2000).

Farbstimmung

Eine weitere zeitliche Veränderung der Farbempfindung tritt bei der Anpassung des Auges an unterschiedliche Beleuchtungsverhältnisse auf. Dieser Anpassungsvorgang, als Hell- bzw. Dunkeladaption bezeichnet, befähigt das Auge, seine Empfindlichkeit auf die jeweiligen Lichtverhältnisse abzustimmen (Lang 1995). Nur im helladaptierten Zustand besitzt das Auge die volle Fähigkeit des Farbensehens (ebd.). Die chromatische Adaption oder Farbstimmung bezeichnet die Anpassung des Auges an verschiedenfarbige Beleuchtungen, beispielsweise beim Wechsel vom Tageslicht zu Glühlampenlicht (ebd.).

Farbkonstanz

Die Tatsache, dass wechselnde Beleuchtung – und damit ein geänderter Farbreiz – keinen Einfluss auf die Erkennbarkeit einer Farbe hat, wird als Farbkonstanz bezeichnet (Schumann & Müller 2000). Die Farbkonstanz ist unmittelbare Konsequenz einer Farbumstimmung und sorgt dafür, dass beispielsweise ein rotes Auto unter wechselnder Beleuchtung immer rot erscheint (Lang 1995).

Persistenzsatz

Für die Durchführung von Farbvergleichen ist der Persistenzsatz von Bedeutung. Nach diesem ist das Urteil über die Gleichheit oder Ungleichheit zweier Farben unabhängig von der Farbstimmung (Lang 1995). Voraussetzungen sind allerdings, dass sich das Auge im helladaptierten Zustand befindet und die Umstimmungen nicht zu groß sind (ebd.).

Farbe und Größe

Der Zusammenhang zwischen Farbe und Größe wurde bereits im Abschnitt 4.3 thematisiert. Allerdings hängt nicht nur die Erkennbarkeit einer Farbe von der Größe eines Objektes ab, sondern auch die Reinheit der Farbe (Schumann & Müller 2000). Dieser Effekt wird dadurch bewirkt, dass ein größeres Objekt größere Teile der Netzhaut des menschlichen Auges bedeckt (ebd.).

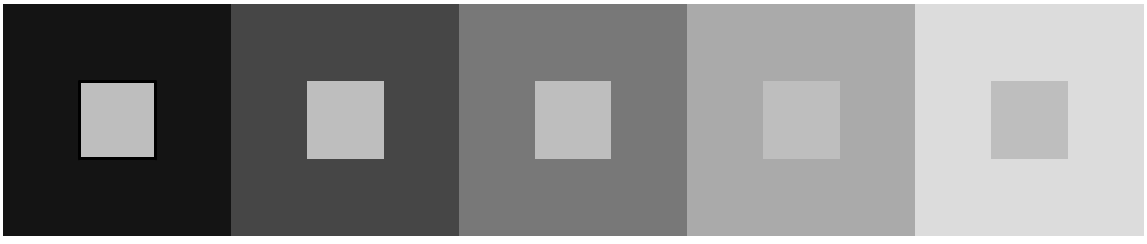


Abbildung 5-3: Wirkung des Simultankontrasts (nach Simon 2008)

Simultankontrast

Als Simultankontrast wird der Effekt bezeichnet, bei dem die Farbempfindung in einem bestimmten Teil des Sehfeldes von den Farben der Umgebung abhängt (Lang 1995). Abbildung

5-3 illustriert ein Beispiel: Die kleineren grauen Quadrate erscheinen – obwohl mit dem gleichen Grauwert belegt – umso heller, je dunkler der Hintergrund ist.

5.1.3 Licht

Ursprung jeglicher Farbwahrnehmung durch das menschliche Auge ist die Wahrnehmung von sichtbarer elektromagnetischer Strahlung in Form von Licht.

5.1.3.1 Licht und Spektrum

Licht und damit ein wahrgenommener Farbreiz kann direkt von einer Lichtquelle ausgehen (Selbstleuchter) oder von einem Objekt, das durch eine Lichtquelle beleuchtet wird, remittiert werden (Nicht-Selbstleuchter) (Schläpfer 1993). Im Falle der Nicht-Selbstleuchter wird die entstehende Farbe auch als Körperfarbe bezeichnet (ebd.).

Die für den Menschen als Licht sichtbare elektromagnetische Strahlung deckt lediglich den sehr geringen Bereich des elektromagnetischen Spektrums von einer Wellenlänge von ca. 380 Nanometern bis ca. 780 Nanometern ab (Lang 1995). In diesem Bereich rufen unterschiedliche Wellenlängen verschiedene Farbwahrnehmungen hervor: Vom kurzwelligen zum langwelligen Teil des Spektrums verlaufen die Farben über violett, blau, cyan, grün, gelb, orange und rot (Abbildung 5-4). Die Farbe der Strahlung einer einzelnen Wellenlänge wird als Spektralfarbe bezeichnet, durch Überlagerung von Licht unterschiedlicher Wellenlängen ergeben sich neue Farbreize. Die Überlagerung aller Wellenlängen des sichtbaren Spektrums ergibt weißes Licht.

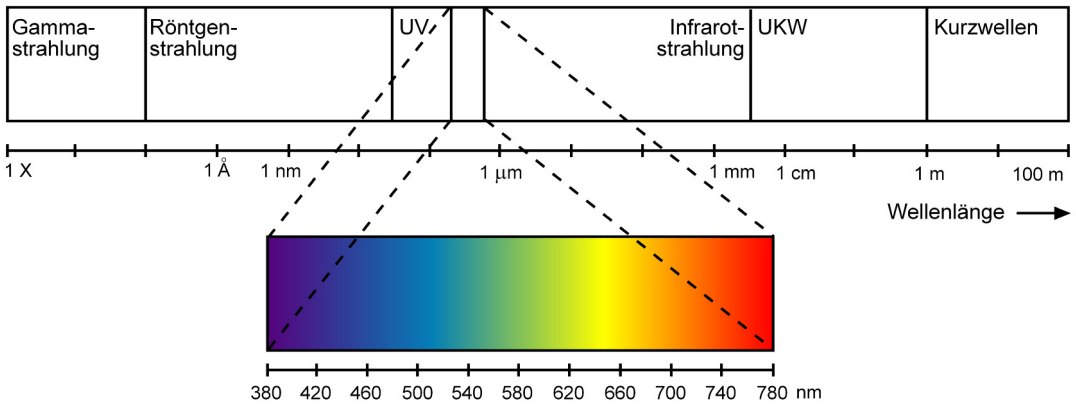


Abbildung 5-4: Spektrum der elektromagnetischen Strahlung mit dem für das menschliche Auge sichtbaren Teil (nach Homann 2007)

Die meisten Lichtquellen senden Strahlung verschiedener, über das Spektrum verteilter, Wellenlängen aus (Schläpfer 1993); dies deckt sich mit der Erfahrung, dass typische Lichtquellen des Alltags vom menschlichen Auge als weiß oder weißlich wahrgenommen werden. Allerdings gibt es auch Lichtquellen, die nur Licht weniger oder gar einer einzigen Wellenlänge abstrahlen; im letzten Fall spricht man auch von monochromatischem Licht (ebd.). Als weitere Eigenschaft einer Lichtquelle wird die Strahlung bei verschiedenen Wellenlängen mit unterschiedlicher Energie ausgesandt. Wird diese Strahlungsenergie, meist als relative Strahlungsenergie, d.h. durch den Energieanteil einer Wellenlänge an der Gesamtenergie, in Ab-

hängigkeit von der Wellenlänge in einem Diagramm aufgetragen, ergibt sich die sogenannte Strahlungsfunktion *S(λ)* (ebd.). *S(λ)* gibt damit die Verteilung der Strahlungsenergie über das sichtbare Spektrum an und charakterisiert eine Lichtquelle bzw. das von dieser emittierte Licht. Abbildung 5-5 zeigt als Beispiele die Strahlungsfunktionen der sogenannten Normlichtarten (vgl. folgenden Abschnitt).

5.1.3.2 Normlichtarten

Die im letzten Abschnitt genannten Körperfarben entstehen durch Beleuchtung eines Objekts durch eine Lichtquelle. Dabei wird die auf die Objektoberfläche auftreffende Strahlung durch Reflexion, Transmission oder Absorption in der Zusammensetzung ihrer Wellenlängen beeinflusst (Schläpfer 1993). Die eigentliche Körperfarbe entsteht durch Absorption, die in verschiedenen Wellenlängenbereichen unterschiedlich wirkt.

Damit hängt eine Körperfarbe also immer sowohl von der Oberfläche eines Objekts, als auch von der Lichtquelle und deren Strahlungsverteilung ab. Letzteres ist besonders für Farbvergleiche von Bedeutung: Körperfarben, die bei der einen Beleuchtung gleich sind, können bei einer anderen verschieden sein (Lang 1995). Für die Durchführung genauer Farbvergleiche ist deshalb die Kenntnis der Strahlungsverteilung der genutzten Lichtquelle eine wesentliche Voraussetzung. Um die Reproduzierbarkeit solcher Lichtquellen sicherzustellen und zu vereinfachen, hat die Commission Internationale de l'Eclairage (Internationale Beleuchtungskommission, kurz CIE[36]) durch die Festlegung von Strahlungsverteilungen sogenannte Normlichtarten eingeführt. Die Normlichtart A repräsentiert beispielsweise das Licht einer Glühlampe, die Normlichtart C das künstliche Tageslicht (Schläpfer 1993).

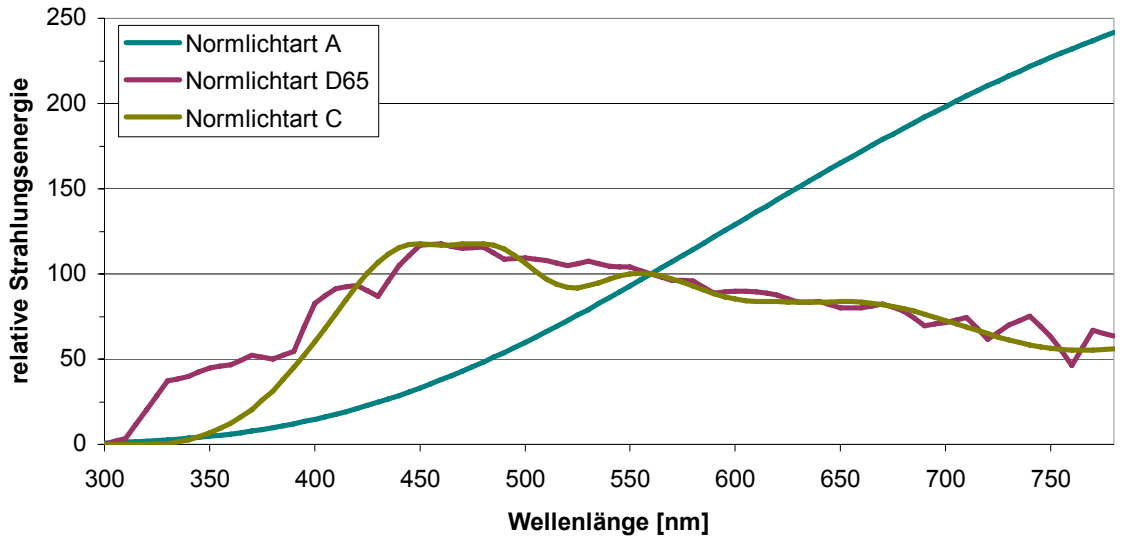


Abbildung 5-5: Strahlungsverteilung der Normlichtarten A, D65 und C (Werte aus Wyszecki & Stiles 1982)

[36] http://www.cie.co.at (Zuletzt geprüft am 13.11.2008)

Da in der Normlichtart C der ultraviolette Anteil des echten Tageslichts nicht enthalten ist, wurde mit der D65 eine weitere Normlichtart definiert, die das natürliche „mittlere“ Tageslicht einschließlich eines UV-Anteils repräsentiert (Schläpfer 1993). Abbildung 5-5 zeigt die Strahlungsverteilungen der drei genannten Normlichtarten. Bedeutung und Gebrauch werden in den späteren Ausführungen zur Farbmetrik deutlich werden.

5.1.4 Farbmischung

Die Reproduktion von Farben erfolgt immer über die Mischung von Farben, dabei wird zwischen *additiver Farbmischung* und *subtraktiver Farbmischung* unterschieden.

Additive Farbmischung

Bei der additiven Farbmischung werden Farbreize optisch gemischt; dies kann beispielsweise erfolgen durch die Übereinanderprojektion farbiger Lichter, durch schnell drehende Farbkreisel oder durch genügend dicht nebeneinander liegende und so vom Auge nicht mehr auflösbare Rasterpunkte (Schläpfer 1993). Abbildung 5-6 zeigt links die Übereinanderprojektion der Bilder dreier Projektoren mit jeweils einem Rot-, Grün- und Blaufilter. Die sich ergebenden Mischfarben sind in diesem Fall Magenta, Cyan, Gelb und Weiß. Die additive Farbmischung ist besonders in der Farbmetrik (vgl. Lang 1995) und für die Farbwiedergabe in Geräten der Kommunikations- und Informationstechnologie von Bedeutung (vgl. Abschnitte 5.3.2 und 5.6.1).

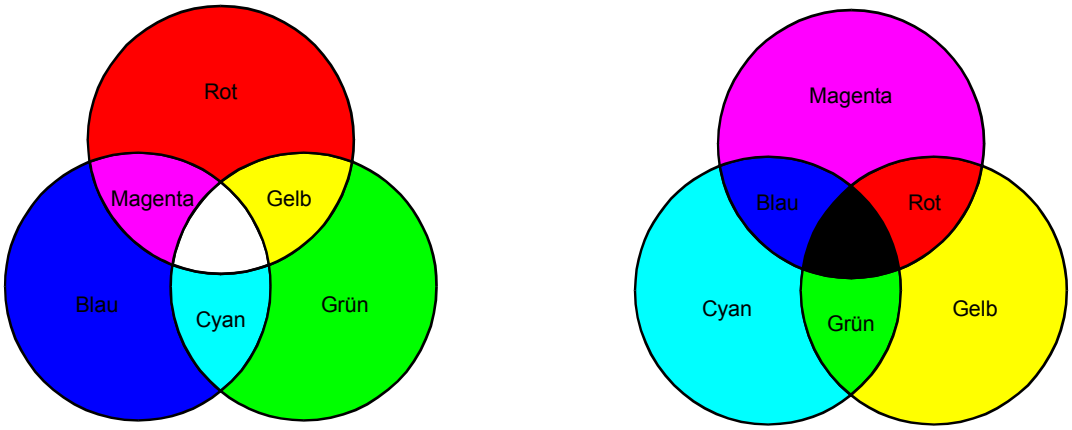


Abbildung 5-6: Additive (links) und subtraktive (rechts) Farbmischung

Subtraktive Farbmischung

Eine subtraktive Farbmischung liegt bei der materiellen Mischung von Farbstoffen vor; dabei kommt der Farbreiz durch die Absorption von Strahlung zustande (Schläpfer 1993). Beispiele für subtraktive Mischungen sind das Mischen von Pigmenten oder farbigen Lösungen sowie das Hintereinanderschalten farbiger Filter (ebd.). Abbildung 5-6 zeigt rechts die Wirkung farbiger Filter (Magenta, Cyan und Gelb) bei Bestrahlung mit weißem Licht: Die Mischfarben sind in diesem Fall Blau, Rot, Grün und Schwarz. Die subtraktive Farbmischung ist besonders für den Druck von Bedeutung.

5.2 Grundlagen der Farbmetrik

Voraussetzung für den Einsatz von Farbe ist deren gezielte Reproduzierbarkeit, d.h. zum einen müssen Farbreize zahlenmäßig beschreibbar werden, zum anderen muss bekannt sein, welcher Farbreiz welche Farbempfindung hervorruft. Die Beantwortung dieser Fragestellung ist Gegenstand der Farbmetrik (vgl. Lang 1995, Schläpfer 1993).

Eine zahlenmäßige Beschreibung von Farbreizen wird durch die Wahl bzw. Festlegung eines Bezugssystems möglich. Farben, die ein solches Bezugssystem bilden, werden als *Primärvalenzen* bezeichnet (Schläpfer 1993). Nach Festlegung von Primärvalenzen lässt sich ein unbekannter Farbreiz in diesem Bezugssystem beschreiben, indem er visuell mit einem aus den Primärvalenzen additiv nachgemischten Farbreiz verglichen wird (Lang 1995). Bei gleicher Farbvalenz beider Reize ist der erste Reiz durch die aktuellen Werte der Primärvalenzen beschrieben (Farbmessung nach dem *Gleichheitsverfahren*[37]).

Zunächst stellt sich demnach die Frage nach Art und Anzahl der für eine Farbbeschreibung notwendigen Primärvalenzen. Nach dem ersten Graßmannschen Gesetz von 1853 sind alle Farben aus drei beliebig wählbaren Primärvalenzen ermischbar (vgl. Lang 1995); Bedingung für die Wahl der Primärvalenzen ist deren Unabhängigkeit, d.h. eine Primärvalenz darf nicht durch die beiden anderen ermischbar sein (Schläpfer 1993). Die mathematische Darstellung einer Farbmischung bzw. Nachmischung aus drei Primärvalenzen wird meist als Addition von Vektoren dargestellt, als Primärvalenzen werden Rot ($\vec{R}$), Grün ($\vec{G}$) und Blau ($\vec{B}$) genutzt (Lang 1995). Für eine beliebige Farbe $\vec{F}$ gilt damit:

$$\vec{F} = R \cdot \vec{R} + G \cdot \vec{G} + B \cdot \vec{B} .$$

Die Forderung nach Unabhängigkeit der Primärvalenzen ist dann aus mathematischer Sicht die Forderung, dass die Vektoren $\vec{R}, \vec{G}, \vec{B}$ nicht komplanar sind.

Die Nachmischung kann sowohl als innere als auch als äußere Mischung vorkommen (Lang 1995). Als innere Farbmischung wird die oben dargestellte Addition von drei Primärvalenzen bezeichnet. Im Fall der äußeren Farbmischung kann eine Farbgleichheit nur erreicht werden, indem eine der Primärvalenzen der zu ermischenden Farbe beigemischt wird:

$$\vec{F} + R_n \cdot \vec{R} = G \cdot \vec{G} + B \cdot \vec{B} \quad \vee \quad \vec{F} + G_n \cdot \vec{G} = R \cdot \vec{R} + B \cdot \vec{B} \quad \vee \quad \vec{F} + B_n \cdot \vec{B} = R \cdot \vec{R} + G \cdot \vec{G} .$$

Die Wahl der genannten Primärvalenzen ist beliebig, einzige Forderung ist deren Unabhängigkeit. Sei $\vec{X}, \vec{Y}, \vec{Z}$ ein zweites Primärvalenzsystem. Es gilt dann (Lang 1995):

$$\vec{F} = R \cdot \vec{R} + G \cdot \vec{G} + B \cdot \vec{B} = X \cdot \vec{X} + Y \cdot \vec{Y} + Z \cdot \vec{Z} .$$

Der Übergang auf das zweite Primärvalenzsystem erfolgt durch eine affine Transformation, dafür müssen sich die Primärvalenzen des neuen Systems im alten System darstellen lassen (ebd.):

[37] Vertiefende Ausführungen finden sich bspw. in Lang (1995).

$$\vec{X} = R_X \cdot \vec{R} + G_X \cdot \vec{G} + B_X \cdot \vec{B}$$
$$\vec{Y} = R_Y \cdot \vec{R} + G_Y \cdot \vec{G} + B_Y \cdot \vec{B}$$
$$\vec{Z} = R_Z \cdot \vec{R} + G_Z \cdot \vec{G} + B_Y \cdot \vec{B}\,.$$

Werden die Koeffizienten in einer Matrix $\underline{A}$ zusammengefasst, erhält man:

$$\begin{pmatrix}\vec{X}\\ \vec{Y}\\ \vec{Z}\end{pmatrix} = \underline{A} \cdot \begin{pmatrix}\vec{R}\\ \vec{G}\\ \vec{B}\end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix}\vec{R}\\ \vec{G}\\ \vec{B}\end{pmatrix} = \underline{A}^{-1} \cdot \begin{pmatrix}\vec{X}\\ \vec{Y}\\ \vec{Z}\end{pmatrix}$$

für die inverse Transformation. Für die Transformation der Farbwerte gilt (ebd.):

$$\begin{pmatrix}X\\ Y\\ Z\end{pmatrix} = \left(\underline{A}^T\right)^{-1} \cdot \begin{pmatrix}R\\ G\\ B\end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix}R\\ G\\ B\end{pmatrix} = \underline{A}^T \cdot \begin{pmatrix}X\\ Y\\ Z\end{pmatrix}.$$

Die Koeffizienten der Matrix $\left(\underline{A}^T\right)^{-1}$ sind dann die Koordinaten der Primärvalenzen $\vec{R},\vec{G},\vec{B}$ im System der Primärvalenzen $\vec{X},\vec{Y},\vec{Z}$.

5.3 Farbsysteme und Farbräume

Die im letzten Abschnitt geschilderte Wahl eines Bezugssystems hat zu verschiedenen Festlegungen numerischer Modelle zur zahlenmäßigen Beschreibung von Farbreizen geführt. Die Grundlagen der Modelle, die für diese Arbeit von Bedeutung sind, werden im Folgenden zusammengefasst: Das Normvalenzsystem der CIE, das als grundlegendes System der Farbwahrnehmung eine Verbindung zwischen Farbreizen und Farbvalenzen herstellt und alle wahrnehmbaren Farben umfasst, das RGB-System zur Beschreibung von Farben in digitalen Medien, das CMY- bzw. CMYK-System für die Nutzung im Druck, sowie CIELUV und CIELAB als sogenannte gleichabständige Farbräume. Weiterhin von Bedeutung sind die Transformationen zwischen diesen Räumen.

5.3.1 CIE-Normvalenzsystem

Eine nach der im Abschnitt 5.2 beschriebenen Methode des Gleichheitsverfahrens bestimmte Verbindung zwischen definierten Farbreizen und der hervorgerufenen Farbempfindung ist durch das CIE-Normvalenzsystem von 1931 gegeben. Es beruht auf empirischen Daten von zwei Farbabgleichs-Messungen, die 1928/29 und 1931 an insgesamt 17 Personen durchgeführt und von der CIE als Standard übernommen wurden (Lang 1995, Wyszecki & Stiles 1982). Ziel dieser Messungen war es, Spektralreize (Licht bestimmter Wellenlängen) in Farbwerten eines Primärvalenzsystems auszudrücken. Als Primärvalenzen wurden bspw. in einer der Messungen monochromatische Farbreize mit den Wellenlängen λ = 650, 530 und 460 *nm* genutzt (Wyszecki & Stiles 1982). Die Messungen erfolgten für den sogenannten farbmetrischen 2°-Normalbeobachter, d.h. für einen normalsichtigen Beobachter, der Farbflächen von 2° Gesichtsfeldgröße betrachtet (ebd.).

Nach einer Transformation auf ein weiteres Primärvalenzsystem aus Spektralreizen wurde dieses von der CIE 1931 als einheitliche Basis festgelegt (Wyszecki & Stiles 1982):

$\vec{R}$: $\lambda = 700\,nm$ (Rot),

$\vec{G}$: $\lambda = 546{,}1\,nm$ (Grün),

$\vec{B}$: $\lambda = 435{,}8\,nm$ (Blau).

Die Nachmischung der Spektralreize durch diese Primärvalenzen ergibt die in Abbildung 5-7 dargestellten Spektralwertkurven $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$. Damit berechnet sich ein spektraler Farbreiz nach (vgl. Lang 1995):

$$\vec{F}(\lambda) = \bar{r}(\lambda) \cdot \vec{R} + \bar{g}(\lambda) \cdot \vec{G} + \bar{b}(\lambda) \cdot \vec{B}\,.$$

Anzumerken ist, dass die Spektralwerte der Abbildung 5-7 auf das energiegleiche Spektrum normiert wurden, d.h. die Farbwerte beziehen sich auf die gleiche relative Strahldichte 1,0 (Wyszecki & Stiles 1982).

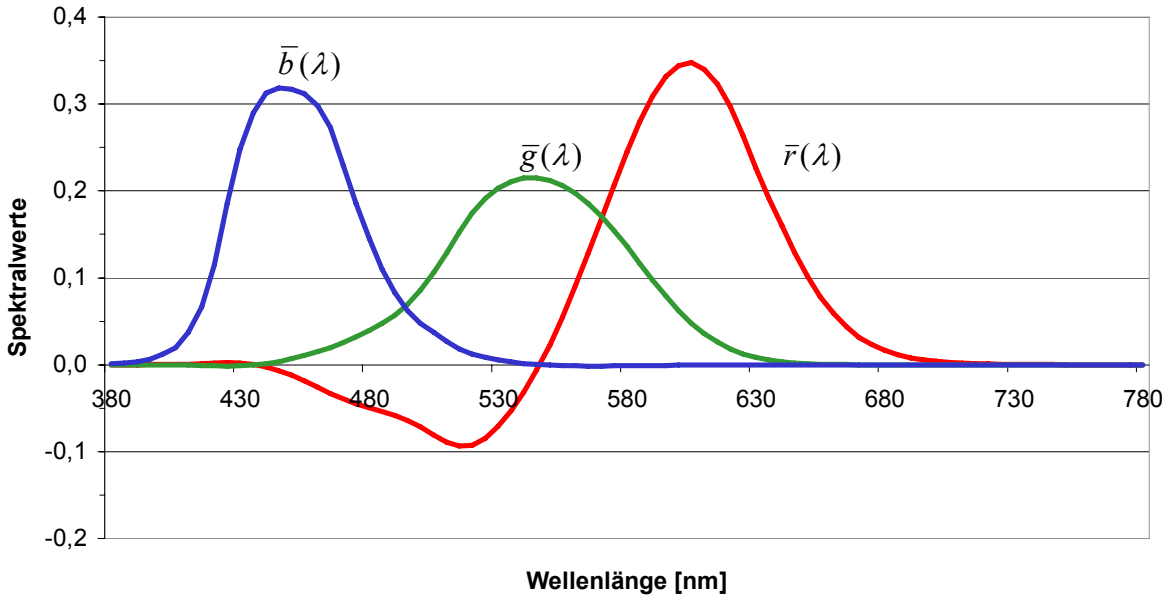


Abbildung 5-7: Spektralwertkurven für die Primärvalenzen der Wellenlängen 700 nm, 546,1 nm und 435.8 nm (Werte aus Wyszecki & Stiles 1982)

Aus Abbildung 5-7 geht hervor, dass auch negative Spektralwerte auftreten, d.h. eine Farbgleichheit kann für diese Spektralreize nur durch äußere Farbmischung erreicht werden. Um diesen Umstand zu vermeiden, wurden Primärvalenzen eingeführt, die den Ausdruck aller Farben ausschließlich durch innere Farbmischung erlauben. Die dafür gewählten Primärvalenzen mit der Bezeichnung $\vec{X}, \vec{Y}, \vec{Z}$ sind virtuelle Primärvalenzen, d.h. nicht physikalisch realisierbar (Schläpfer 1993). Das XYZ-Bezugssystem wird als *Normvalenzsystem* bezeichnet. Weitere Bedingungen für dessen Festlegung waren u.a. (Schläpfer 1993):

- Der Normfarbwert Y ist proportional zur Helligkeit eines Farbreizes.

- Beim energiegleichen Spektrum entsprechen gleiche Normfarbwerte unbunten Farbreizen; für das ideale Weiß gilt $X = Y = Z = 100$.

Die Umrechnung zwischen den Primärvalenzen bzw. den Farbwerten beider Systeme erfolgt nach den Transformationen im Abschnitt 5.2 (für die Transformationsparameter siehe Wyszecki & Stiles (1982)). Die Umrechnung der Spektralwertkurven $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ ergibt dann die Normspektralwertkurven $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ der Abbildung 5-8 (Die Farben der dargestellten Kurven stellen nicht die Farben der Primärvalenzen dar!). Gemäß der Festlegung Y ~ Leuchtdichte entspricht die Spektralwertkurve $\bar{y}(\lambda)$ der Hellempfindlichkeitskurve des helladaptierten menschlichen Auges.

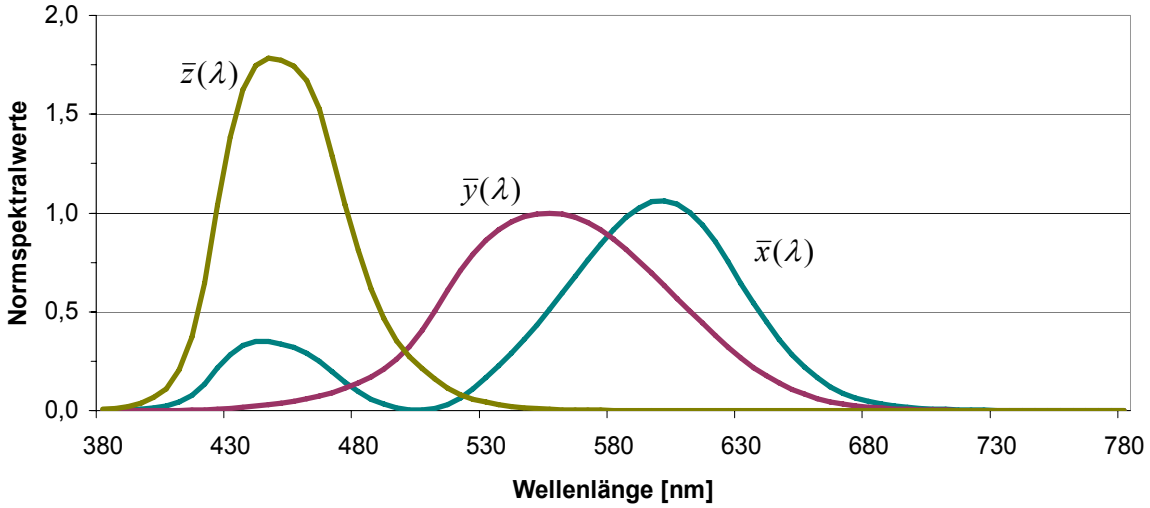


Abbildung 5-8: Normspektralwertkurven (Werte aus Wyszecki & Stiles 1982)

Die Normfarbwerte eines beliebigen Farbreizes, der ja als Mischung von Spektralfarben aufzufassen ist, lassen sich durch die Normspektralwertkurven rechnerisch bestimmen. Unter der Voraussetzung, dass die spektrale Strahlungsverteilung $\varphi(\lambda)$ des gesuchten Farbreizes bekannt ist, gilt (Lang 1995):

$$X = k \cdot \int_0^\infty \varphi(\lambda) \cdot \bar{x}(\lambda) \cdot d\lambda \,, \qquad Y = k \cdot \int_0^\infty \varphi(\lambda) \cdot \bar{y}(\lambda) \cdot d\lambda \,, \qquad Z = k \cdot \int_0^\infty \varphi(\lambda) \cdot \bar{z}(\lambda) \cdot d\lambda$$

mit

$$k = \frac{100}{\int_0^\infty \varphi(\lambda) \cdot \bar{y}(\lambda) \cdot d\lambda} \,.$$

Der Normierungsfaktor *k* stellt sicher, dass für die ideal weiße Fläche $Y = 100$ erhalten wird.

Identische Normfarbwerte $X = Y = Z = 100$ für das ideale Weiß ergeben sich nur dann, wenn die genutzte Lichtquelle ein energiegleiches Spektrum besitzt, für alle anderen Lichtquellen unterscheiden sich die Normfarbwerte (Schläpfer 1993). Für die Normlichtarten C und D65 gelten beispielsweise die in Tabelle 5-1 angegebenen Werte.

Normfarbwert	C	D65
X	98,07	95,04
Y	100,00	100,00
Z	118,23	108,89

Tabelle 5-1: Normfarbwerte des idealen Weißes für die Normlichtarten C und D65 (Werte aus Schläpfer 1993)

Das Normvalenzsystem wird häufig dahingehend kritisiert, dass es zwar möglich ist, durch den Normfarbwert Y eine Vorstellung von der Helligkeit eines Farbreizes zu bekommen, allerdings nur schwer von dessen Farbton und Sättigung (Schläpfer 1993). Aus diesem Grund werden durch eine Projektion der Normfarbwerte in die Ebene die sogenannten *Normfarbwertanteile* x, y, z erhalten (ebd.):

$$x = \frac{X}{X+Y+Z}, \qquad y = \frac{Y}{X+Y+Z}, \qquad z = \frac{Z}{X+Y+Z} = 1 - x - y. \tag{5.1}$$

Die Normfarbwertanteile x und y werden in einem ebenen rechtwinkligen Koordinatensystem aufgetragen; das Ergebnis – die *Normfarbtafel* – ist in Abbildung 5-9 links zu sehen.

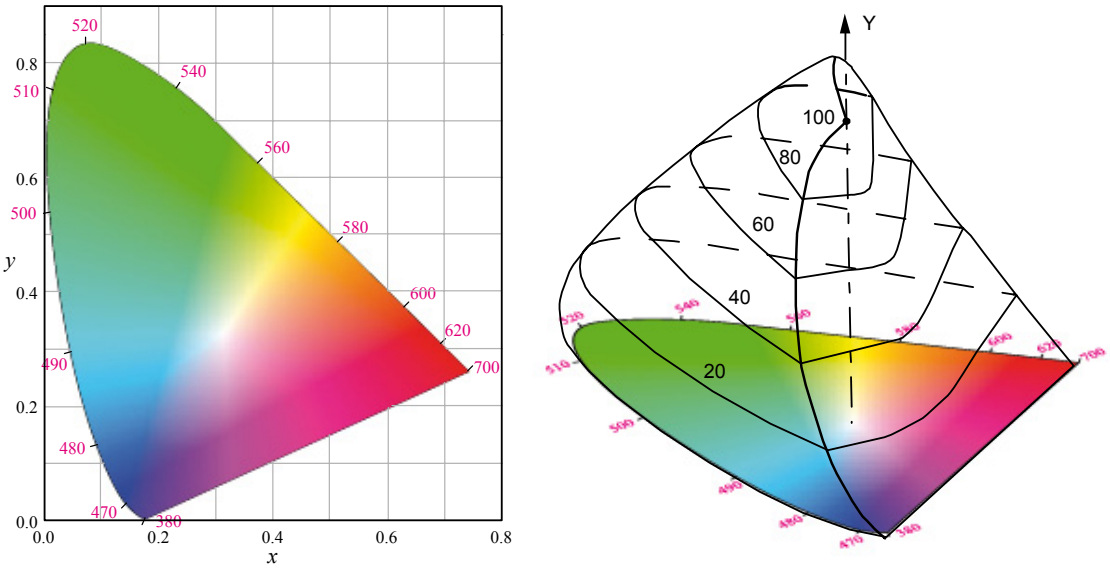


Abbildung 5-9: Normfarbtafel für den farbmetrischen 2°-Normalbeobachter (links, nach Schläpfer 1993, Wyszecki & Stiles 1982); Grenzen des Farbkörpers nach Rösch (rechts, nach Wyszecki & Stiles 1982)

Der Bereich der reellen Farbvalenzen wird von dem Spektralfarbenzug und der Purpurgeraden umschlossen (Schläpfer 1993): Der Spektralfarbenzug besteht aus den Normspektralwertanteilen der Spektralwerte und bildet einen hufeisenförmigen Kurvenzug; die Purpurgerade, auf der die reinst möglichen Purpurfarben liegen, schließt das Hufeisen. In dem so umschlossenen Bereich liegen alle Farbvalenzen, die sich durch additive Mischung aus den Spektralvalenzen erzielen lassen (ebd.).

Der Punkt mit den Normfarbwertanteilen $x = y = 0{,}333$ wird als Mittelpunktsvalenz bezeichnet; für das energiegleiche Spektrum liegt dort der Unbunton (Schläpfer 1993). Im Falle von

Körperfarben gilt die Farbvalenz des beleuchtenden Lichtes als unbunt, die Mittelpunktsvalenz verschiebt sich dementsprechend, beispielsweise gilt für die Normlichtart C x = 0,3101 und y = 0,3162 sowie für die Normlichtart D65 x = 0,3127 und y = 0,3290 (ebd.).

Die Normfarbtafel repräsentiert Sättigung und Farbton einer Farbvalenz (Schläpfer 1993). Als dritte Dimension, senkrecht zur x-y-Ebene, wird häufig der Normfarbwert Y als Helligkeit übernommen. Werden für die Normlichtart D65 die hellstmöglichen Körperfarben für jeden Farbton (Optimalfarben) über der Normfarbtafel aufgetragen, wird der Farbkörper nach Rösch erhalten (Wyszecki & Stiles 1982). Abbildung 5-9 illustriert rechts diesen Farbkörper: Es wird deutlich, dass die Helligkeit entlang des Spektralfarbenzugs nur geringe Werte annehmen kann; für die Mittelpunktsvalenz wird Y = 100 erreicht.

5.3.2 RGB- und sRGB-Farbraum

In den Abschnitten 5.1.4 und 5.2 wurde bereits die additive Farbmischung der drei Primärvalenzen Rot, Grün und Blau beschrieben. Mit der stetig steigenden Verbreitung von Geräten der Informations- und Unterhaltungsindustrie (z.B. Bildschirme, Digitalkameras) kam und kommt solchen RGB-Farbräumen eine wachsende Bedeutung zu, da diese Geräte RGB-Farbräume zur Repräsentation und Wiedergabe von Farbe nutzen. Allerdings ist anzumerken, dass die jeweils genutzten Primärvalenzen vom Gerät und zum Teil auch von dessen Hersteller abhängen und in der Regel nicht identisch sind mit den Primärvalenzen der CIE.

RGB-Farbräume werden typischerweise durch die Angabe der Normfarbwertanteile der Primärvalenzen ($x_R, y_R, x_G, y_G, x_B, y_B$) sowie des Weißpunktes (x_W, y_W) charakterisiert (Simon 2008).

Name	x_W	y_W	x_R	y_R	x_G	y_G	x_B	y_B
CIE-RGB	.3333	.3333	.7350	.2650	.2740	.7170	.1670	.0090
sRGB	.3127	.3290	.6400	.3300	.3000	.6000	.1500	.0600
EBU-Monitor	.3457	.3585	.6314	.3391	.2809	.5971	.1487	.0645
Adobe-RGB	.3127	.3290	.6250	.3400	.2800	.5950	.1550	.0700

Tabelle 5-2: Normfarbwertanteile und Gammawerte verschiedener RGB-Farbräume (nach Simon 2008)

Tabelle 5-2 fasst beispielhaft die Charakteristika einiger RGB-Farbräume zusammen:

- *CIE-RGB*, der von der CIE normierte Farbraum (Abschnitt 5.3.1),
- *sRGB* (Standard-RGB), die sehr weit verbreitete RGB-Spezifikation für Displays von Personal Computern (vgl. weitere Ausführungen dieses Abschnitts),
- *EBU-Monitor*, die in Europa spezifizierten Primärvalenzen für Farbfernsehbildröhren (Lang 1995),
- *Adobe-RGB*, ein von der Firma Adobe definierter RGB-Farbraum.

Die Transformation von Farbwerten eines RGB-Farbraums in einen beliebigen anderen Farbraum bzw. die inverse Transformation erfolgt über das Normvalenzsystem. Dafür ist es notwendig, dass die Normfarbwerte der RGB-Primärvalenzen bekannt sind (Lang 1995):

$$\vec{R} = \begin{pmatrix} X_R \\ Y_R \\ Z_R \end{pmatrix}, \qquad \vec{G} = \begin{pmatrix} X_G \\ Y_G \\ Z_G \end{pmatrix}, \qquad \vec{B} = \begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix} \tag{5.2}$$

Diese Normfarbwerte sind aus den Normfarbwertanteilen der Primärvalenzen ($x_R, y_R, x_G, y_G, x_B, y_B$) und des Weißpunktes ($x_W, y_W$) errechenbar (Simon 2008). Eine Beschreibung des Rechenwegs findet sich im Anhang A.3.1 dieser Arbeit.

Die Berechnung für den sRGB-Farbraum mit den Werten aus Tabelle 5-2 ergibt folgende Normfarbwerte der Primärvalenzen:

$$\vec{R} = \begin{pmatrix} 0{,}4124 \\ 0{,}2126 \\ 0{,}0193 \end{pmatrix}, \qquad \vec{G} = \begin{pmatrix} 0{,}3576 \\ 0{,}7152 \\ 0{,}1192 \end{pmatrix}, \qquad \vec{B} = \begin{pmatrix} 0{,}1805 \\ 0{,}0722 \\ 0{,}9505 \end{pmatrix}.$$

Im Abschnitt 5.2 wurde die Transformation von Farbwerten eines RGB-Farbraums in einen XYZ-Farbraum folgendermaßen angegeben:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \left(\underline{A}^T\right)^{-1} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Da die Koeffizienten der Matrix $\left(\underline{A}^T\right)^{-1}$ aus den Koordinaten der Primärvalenzen $\vec{R}, \vec{G}, \vec{B}$ im System der Primärvalenzen $\vec{X}, \vec{Y}, \vec{Z}$ bestehen, ergibt sich für die Transformation von RGB-Farbwerten in Normfarbwerte:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = 100 \cdot \begin{pmatrix} 0{,}4124 & 0{,}3576 & 0{,}1805 \\ 0{,}2126 & 0{,}7152 & 0{,}0722 \\ 0{,}0193 & 0{,}1192 & 0{,}9505 \end{pmatrix} \cdot \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}. \tag{5.3}$$

Die Umkehrung ist dementsprechend gegeben durch:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \frac{1}{100} \begin{pmatrix} 3{,}2406 & -1{,}5372 & -0{,}4986 \\ -0{,}9689 & 1{,}8758 & 0{,}0415 \\ 0{,}0557 & -0{,}2040 & 1{,}0570 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{5.4}$$

Die Farbwerte im System $\vec{R}, \vec{G}, \vec{B}$ sind in den Beziehungen (5.3) und (5.4) durch R', G', B' als temporäre Farbwerte gekennzeichnet, d.h. es muss jeweils ein weiterer Transformationsschritt erfolgen. Grund ist die sogenannte *Gammakorrektur*, deren Ursprung in der Funktionsweise von Kathodenstrahlbildschirmen liegt (Simon 2008). Diese Bildschirme nutzen zur Erzeugung des Lichtes Phosphore, die durch Beschuss mittels eines Elektronenstrahls zum Leuchten angeregt werden. Die Leuchtdichte wird durch die angelegte Beschleunigungsspannung geregelt, diese wird wiederum durch die Farbwerte des RGB-Farbraums gesteuert. Der Zusammenhang zwischen Beschleunigungsspannung und Leuchtdichte ist nicht linear, sondern genügt der Beziehung (vgl. Simon 2008):

$$L = L(P) = A \cdot (k_1 \cdot P - k_0)^{\gamma} \tag{5.5}$$

mit der maximal erzeugbaren Helligkeit A, einem einstellbaren Verstärkungsfaktor k_1, einem die Pixelhelligkeit repräsentierenden Spannungswert P und der Ausgangsspannung k_0. Der Gammawert γ liegt üblicherweise bei 2,2 (ebd.).

Da die Beschleunigungsspannung direkt durch RGB-Koordinaten geregelt wird, werden letztere durch die Gammakorrektur so verändert, dass sich ein linearer Zusammenhang zwischen RGB-Koordinaten und Leuchtdichte ergibt. In der Informatik wird dafür meist vereinfachend angenommen (Simon 2008):

$$A = 1, \qquad k_1 = 1, \qquad k_0 = 0\,.$$

Durch Einsetzen der Vereinfachungen in Gleichung 5.5 und Invertierung werden nichtlineare Koordinaten erhalten (Simon 2008):

$$R'' = (R')^{\frac{1}{\gamma}}, \qquad G'' = (G')^{\frac{1}{\gamma}}, \qquad B'' = (B')^{\frac{1}{\gamma}}.$$

Wird die Gammakorrektur bei der Transformation von Normfarbwerten nach sRGB berücksichtigt, müssen im Anschluss an die Transformation der Formel 5.4 die Verbesserungen

$$R'' = \begin{cases} 12{,}92 \cdot R' & \textit{für } R' \le 0{,}0031308 \\ 1{,}055 \cdot (R')^{\frac{1}{2{,}4}} - 0{,}055 & \textit{sonst} \end{cases}$$

angebracht werden (Simon 2008). Analoge Rechnungen müssen für G' und B' erfolgen.

Dementsprechend muss die Umkehrung vor Anwendung der Transformation (5.3) berücksichtigt werden (analog wieder für G'' und B''):

$$R' = \begin{cases} \dfrac{R''}{12{,}92} & \textit{für } R'' \le 0{,}04045 \\ \left(\dfrac{R''+0{,}055}{1{,}055}\right)^{2{,}4} & \textit{sonst} \end{cases}.$$

Der in diesen Gleichungen genutzte Gammawert $\gamma = 2{,}4$ entspricht zusammen mit der sogenannten Offset-Konstanten von 1,055 in etwa einem Gammawert von $\gamma = 2{,}2$ (Simon 2008).

Die Primärvalenzen eines RGB-Farbraums spannen ein dreidimensionales kartesisches Koordinatensystem auf; die Darstellung erfolgt häufig als Einheitswürfel (Abbildung 5-10), d.h. die Farbwerte für jede der Primärvalenzen liegen in einem Intervall [0,1].

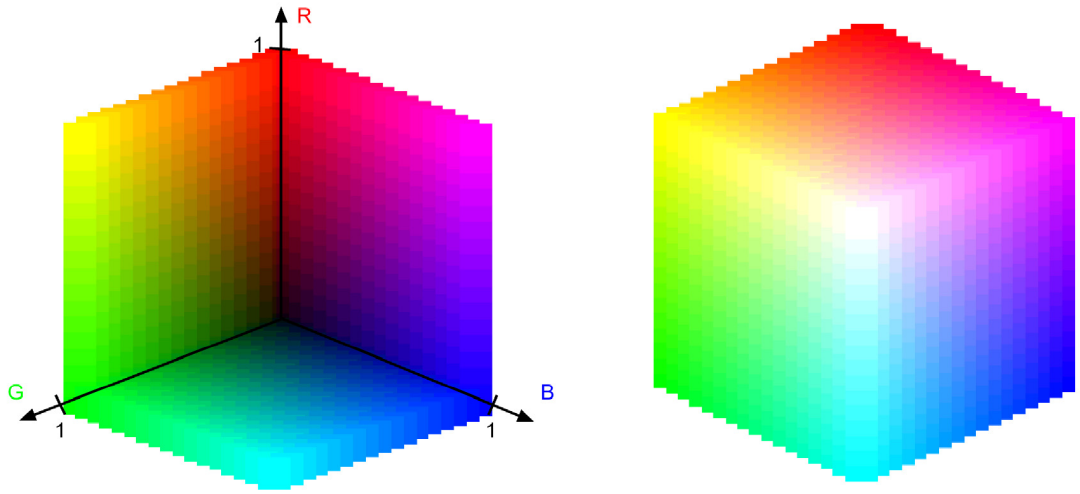


Abbildung 5-10: RGB-Farbwürfel, links mit Blick auf den Ursprung (schwarz), rechts auf den Punkt mit den Koordinaten (R,G,B) = (1,1,1) (weiß)

Bei der Verwendung des RGB-Modells auf einem Computer stehen für die Speicherung von Farben üblicherweise 8 Bit (1 Byte) für jede Primärvalenz zur Verfügung; jede Grundfarbe wird dann durch einen Wertebereich $0 \leq R, G, B \leq 2^8 - 1 = 255$ repräsentiert. Tabelle 5-3 zeigt die Farbwerte der Primärvalenzen und der Mischfarben 1. Ordnung in dieser 8-Bit-Kodierung.

	Weiß	Gelb	Cyan	Grün	Purpur/ Magenta	Rot	Blau	Schwarz
Rot	255	255	0	0	255	255	0	0
Grün	255	255	255	255	0	0	0	0
Blau	255	0	255	0	255	0	255	0

Tabelle 5-3: RGB-Anteile der Primärvalenzen und Mischfarben 1. Ordnung bei einer 8-Bit-Codierung

Vertiefende Ausführungen zur Farbrepräsentation auf digitalen Geräten erfolgen im Abschnitt 5.6.1.

5.3.3 CMY und CMYK

Der CMY bzw. CMYK-Farbraum wird für die Ziele dieser Arbeit nicht primär benötigt, aber aufgrund seiner allgemeinen Bedeutung für die persistente Farbwiedergabe durch Körperfarben (Drucker, Offsetdruck) kurz skizziert. CMY und CMYK bezeichnen dabei den *idealen* und *realen* Mehrfarbendruck. Der ideale Druck geht vom subtraktiven Farbmodell (Abschnitt 5.1.4) mit den Grundfarben Cyan (C), Magenta (M) und Yellow (Y) aus. Das Übereinanderdrucken dieser Grundfarben wirkt auf das Papierweiß wie das Übereinanderblenden von Farbfiltern (Lang 1995). Durch Mischung je zwei der Grundfarben entstehen die subtraktiven Mischfarben erster Ordnung: Rot, Grün und Blau (vgl. Abbildung 5-6).

Der ideale Druck ist allerdings technisch nur annähernd realisierbar (Simon 2008). Dies führt u.a. dazu, dass Grauwerte nur bedingt wiedergegeben werden können (ebd.). Aus diesem Grund werden C, M und Y im realen Mehrfarbendruck durch Schwarz zum CMYK-Farbraum ergänzt (Das K steht für den letzten Buchstaben in *black*). Schwarz ist damit in diesem Raum redundant repräsentiert.

5.3.4 Empfindungsgemäße Farbräume

In den bisherigen Ausführungen zur Ableitung des Normvalenzsystems lag das Ziel in der Bestimmung der Gleichheit von Farben. In vielen Fällen ist allerdings der Abstand von Farben bzw. die Berechnung von Farbdifferenzen aus Farbmaßzahlen von Bedeutung. Anhand der Normfarbtafel (Abbildung 5-9 links) werden aber bereits die ungleichen Flächenverhältnisse der dominierenden Farben deutlich. Daraus lässt sich schließen, dass rechnerisch ermittelte gleiche Abstände in vielen Fällen nicht als visuell gleichabständig empfunden werden. Abbildung 5-11 verdeutlicht die tatsächliche Situation für die Normfarbtafel. Die dargestellten Ellipsen (Schwellenellipsen), die von MacAdam experimentell bestimmt wurden, geben das Maß an, um das eine Farbart in einer bestimmten Richtung geändert werden kann, ohne dass das menschliche Auge einen Unterschied wahrnimmt (Lang 1995). In einer empfindungsgemäß gleichabständigen Farbtafel müssten dagegen alle Ellipsen gleichgroße Kreise sein.

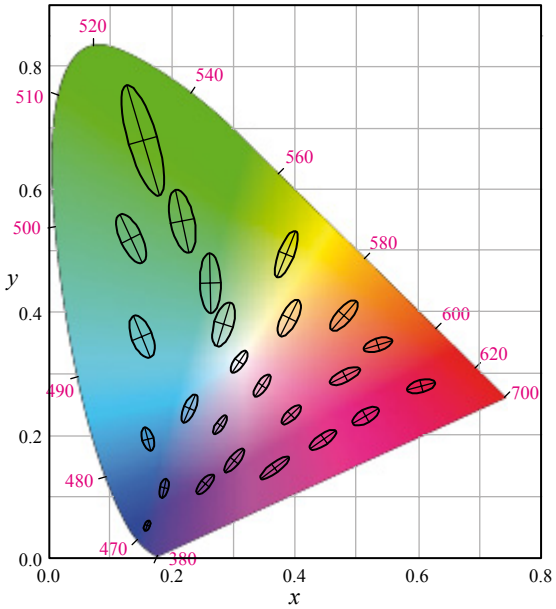


Abbildung 5-11: Normfarbtafel mit MacAdam-Ellipsen in 10-fach vergrößerter Darstellung (nach Schläpfer 1993, Wyszecki & Stiles 1982)

Ein visuell gleichabständiges System, das die direkte Messung von Farbabständen erlaubt, existiert allerdings nicht (Schläpfer 1993), d.h. insbesondere in den bisher vorgestellten Systemen ist diese Möglichkeit nicht gegeben. Somit müssen die XYZ-Werte des Normfarbsystems rechnerisch in einen gleichabständigen Farbraum transformiert werden.

Als gleichabständige Farbräume sind der CIELUV- und der CIELAB-Farbraum gebräuchlich. Beide Farbräume bieten keine 100%ige Gleichabständigkeit, sondern sind ein Kompromiss zwischen einer mathematisch einfachen Transformation und einer bestmöglichen Gleichabständigkeit (vgl. Schläpfer 1993).

5.3.4.1 CIELUV 1976

Das CIELUV-System wurde 1976 von der CIE als visuell gleichabständiger Farbraum empfohlen und hat seinen Anwendungsbereich vor allem bei Selbstleuchtern (Lang 1995). Die Berechnung der CIELUV-Koordinaten L^*, u^* und v^* erfolgt durch Transformation der Normfarbwerte XYZ bzw. der Normfarbwertanteile xyY.

Dazu werden zunächst durch projektive Transformation die Normfarbwerte bzw. die Normfarbwertanteile in die ebenen rechtwinkligen Koordinaten u', v' der CIE-UCS[38]-Farbtafel 1976 überführt (Schläpfer 1993):

$$u' = \frac{4X}{X+15Y+3Z} = \frac{4x}{-2x+12y+3}, \qquad v' = \frac{9Y}{X+15Y+3Z} = \frac{9y}{-2x+12y+3}.$$

Abbildung 5-12 gibt diese Farbtafel wieder.

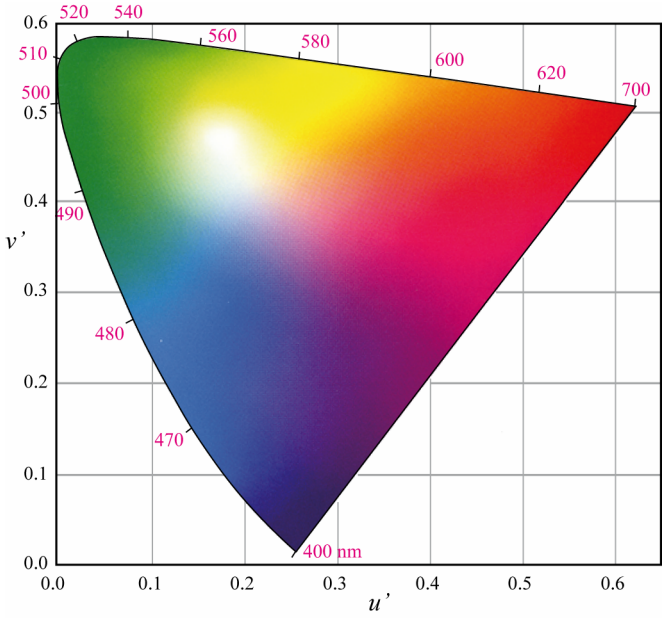


Abbildung 5-12: u'v'-Farbtafel (nach Schläpfer 1993, Wyszecki & Stiles 1982)

Durch Einbeziehung der Helligkeit wird die UCS-Farbtafel zu einem dreidimensionalen Farbraum erweitert. Dazu wird zunächst der Normfarbwert Y in eine empfindungsgemäß gleichabständig gestufte Helligkeitsskala, die *psychometrische Helligkeit* L^*, transformiert (Wyszecki & Stiles 1982):

$$L^* = \begin{cases} 903{,}29\left(\frac{Y}{Y_W}\right) & \text{für } \frac{Y}{Y_W} \le 0{,}008856 \\ 116\left(\frac{Y}{Y_W}\right)^{1/3} - 16 & \text{für } 0{,}008856 < \frac{Y}{Y_W}. \end{cases}$$

[38] Uniform Chromaticity Scale

Y_W bezeichnet den Normfarbwert des Bezugsweißes, dessen Einbeziehung die maximale Helligkeit auf 100 normiert (für Farbwerte des Bezugsweißes vgl. Abschnitt 5.3.1).

Im letzten Schritt werden die Koordinaten der UCS-Farbtafel unter Berücksichtigung der psychometrischen Helligkeit L^* in die Koordinaten u^* und v^* transformiert:

$$u^* = 13L^*(u' - u_W'), \qquad v^* = 13L^*(v' - v_W').$$

Darin sind u_W und v_W die Farbwerte des Referenzweißes im UCS-System (bei Körperfarben die Farbart der Beleuchtung, vgl. Abschnitt 5.3.1). Für unbunte Farben gilt dann $u^*, v^* = 0$ und die Repräsentation erfolgt ausschließlich durch die Helligkeit. Damit wird der Buntton im System L^*, u^*, v^* durch die Koordinaten u^* und v^* beschrieben: Die u^*-Achse verläuft in Richtung Rot-Grün, die v^*–Achse in Richtung Blau-Gelb (Lang 1995). Der Buntton ist rötlich für $u^* > 0$, grünlich für $u^* < 0$, gelblich für $v^* > 0$ und bläulich für $v^* < 0$ (ebd.).

Als weitere Größen des CIELUV lassen sich aus u^* und v^* die *psychometrische Buntheit*, die *psychometrische Sättigung* und der *Bunttonwinkel* angeben. Die Buntheit

$$C^*_{uv} = \sqrt{(u^*)^2 + (v^*)^2}$$

gibt den Abstand der Farbe von Unbunt an (Lang 1995). Von der Buntheit unterschieden werden muss hier die Sättigung, diese wird unter Berücksichtigung der Helligkeit errechnet (Schläpfer 1993):

$$s_{uv} = C^*_{uv} / L^*.$$

Der Buntton- oder Farbtonwinkel gibt einen Farbton in der u^*–v^*-Ebene durch einen Winkel an (Schläpfer 1993):

$$h_{uv} = \arctan(v^* / u^*).$$

5.3.4.2 CIELAB 1976

Als weiterer visuell gleichabständiger Farbraum wurde ebenfalls 1976 von der CIE das CIELAB-System empfohlen. Die Anwendung erfolgt hauptsächlich bei der Arbeit mit Körperfarben (Lang 1995). Das CIELAB-System 1976 wird durch die Koordinaten L^*, a^*, b^* gekennzeichnet. Die Berechnung aus den Normfarbwerten XYZ erfolgt durch folgende Formeln (Simon 2008, Wyszecki & Stiles 1982):

$$L^* = \begin{cases} 903{,}29\left(\dfrac{Y}{Y_W}\right) & \text{für } \dfrac{Y}{Y_W} \le 0{,}008856 \\ 116\left(\dfrac{Y}{Y_W}\right)^{1/3} - 16 & \text{für } 0{,}008856 < \dfrac{Y}{Y_W}. \end{cases}$$

$$a^* = 500\left[f\left(\frac{X}{X_W}\right) - f\left(\frac{Y}{Y_W}\right)\right], \qquad b^* = 200\left[f\left(\frac{Y}{Y_W}\right) - f\left(\frac{Z}{Z_W}\right)\right]$$

mit

$$f(w) = \begin{cases} \sqrt[3]{w} & \text{für } w > 0{,}008856 \\ 7{,}787w + 16/116 & \text{für } w \le 0{,}008856. \end{cases}$$

X_W, Y_W, Z_W bezeichnen darin wiederum die Normfarbwerte des Bezugsweißes, die Helligkeitsfunktion ist offensichtlich identisch mit der des CIELUV-Systems.

Abbildung 5-13 veranschaulicht den Aufbau des CIELAB-Farbraums. Daraus geht hervor, dass die a^*-Achse in Richtung Rot-Grün verläuft, die b^*-Achse in Richtung Blau-Gelb. Unbunte Farben werden, wie im CIELUV-System, durch die Helligkeit repräsentiert.

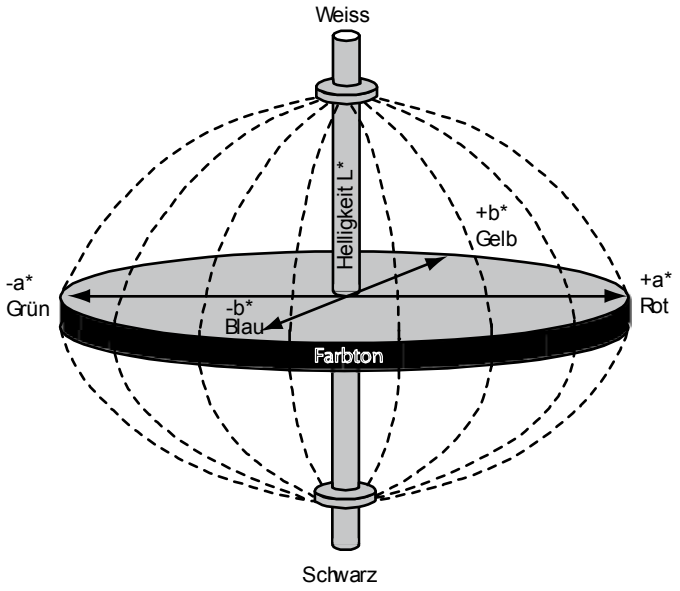


Abbildung 5-13: CIELAB-Farbraum (nach Schläpfer 1993)

Analog zum CIELUV-Farbraum lassen sich auch hier die *psychometrische Buntheit* und der *Bunttonwinkel* angeben. Erstere berechnet sich nach (Schläpfer 1993)

$$C^*_{ab} = \sqrt{(a^*)^2 + (b^*)^2},$$

letzterer nach

$$h_{ab} = \arctan(b^* / a^*).$$

Eine Sättigung ist im CIELAB-Farbraum nicht definiert.

5.3.4.3 Berechnung von Farbabständen

Die Beschreibung von CIELUV- und CIELAB-Farbraum wurde durch deren visuelle Gleichabständigkeit motiviert: Gleiche rechnerische Farbabstände entsprechen empfundenen Abständen. Ein solcher Farbabstand ΔE ist als Euklidische Distanz für den CIELUV-Farbraum durch (Lang 1995)

$$\Delta E_{uv} = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2}$$

und für den CIELAB-Farbraum durch

$$\Delta E_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$$

gegeben. Um für einen Farbabstand die einzelnen Komponenten der Farbempfindung identifizieren zu können, lässt sich dieser auch allgemein, d.h. gültig für CIELUV und CIELAB, durch die Helligkeitsdifferenz ΔL, die Farbtondifferenz ΔH und die Buntheitsdifferenz ΔC ausdrücken (Schläpfer 1993):

$$\Delta E = \sqrt{(\Delta L)^2 + (\Delta H)^2 + (\Delta C)^2} \ .$$

Die Berechnung der ΔL und ΔC ist nach den jeweiligen Ausführungen zum CIELUV- und CIELAB-System offensichtlich. Die Farbtondifferenz ist dagegen nur indirekt über den Farbabstand berechenbar (vgl. Schläpfer 1993):

$$\Delta H = \sqrt{(\Delta E)^2 - (\Delta L)^2 - (\Delta C)^2} \ .$$

5.4 Farbordnungssystem nach Munsell

Eine Sammlung von Farbmustern, die empfindungsgemäß gleichabständig klassifiziert sind, wird als Farbordnungssystem bezeichnet (Schläpfer 1993). Ein solches System ermöglicht eine Auswahl aus einer großen Anzahl materiell ausgeführter Farben.

Ein sehr bekanntes Farbordnungssystem ist das in den USA entwickelte Munsell-System (Simon 2008), die darauf basierende Farbmustersammlung ist das Munsell Book of Colors (Schläpfer 1993).

Die Farben des Munsell-Systems werden durch die Größen Value V (Helligkeit), Hue H (Buntton) und Chroma C (Buntheit, Sättigung) parametrisiert, der durch diese Parameter aufgespannte Farbraum wird durch einen Zylinder beschrieben (Simon 2008). Die Zylinderachse entspricht der Helligkeitsskala, die Farbtöne sind auf einem Farbkreis auf dem Zylindermantel angeordnet; die Sättigung wird von der Zylinderachse in Richtung Zylindermantel gezählt (ebd.). Abbildung 5-14 (links) deutet diesen Zylinder an.

Die Helligkeitsskala ist elffach unterteilt und verläuft von 0 (schwarz) bis 10 (weiß) (Simon 2008). Der Farbkreis wird zunächst durch fünf Hauptfarbtöne (Rot – R, Gelb – Y, Grün- G, Blau –B, Purpur – P) gleichmäßig unterteilt (Abbildung 5-14, rechts) (Schläpfer 1993). Durch Mischung benachbarter Töne entstehen fünf weitere Farben (Gelbrot – YR, Grüngelb – GY, Blaugrün – BG, Purpurblau – PB, Purpurrot – RP) und damit ein zehnteiliger Farbkreis. Jede dieser 10 Farben wird wiederum 10-fach unterteilt, so dass sich insgesamt 100 Farbtöne ergeben (ebd.). Für die Sättigung werden 14 Stufen unterschieden (Simon 2008): Ein Grauwert auf der Zylinderachse wird durch Chroma 0 repräsentiert, die maximale Sättigung auf dem Zylindermantel durch Chroma 14. Die maximale Sättigung wird allerdings für viele Farbtöne nicht erreicht (Schläpfer 1993).

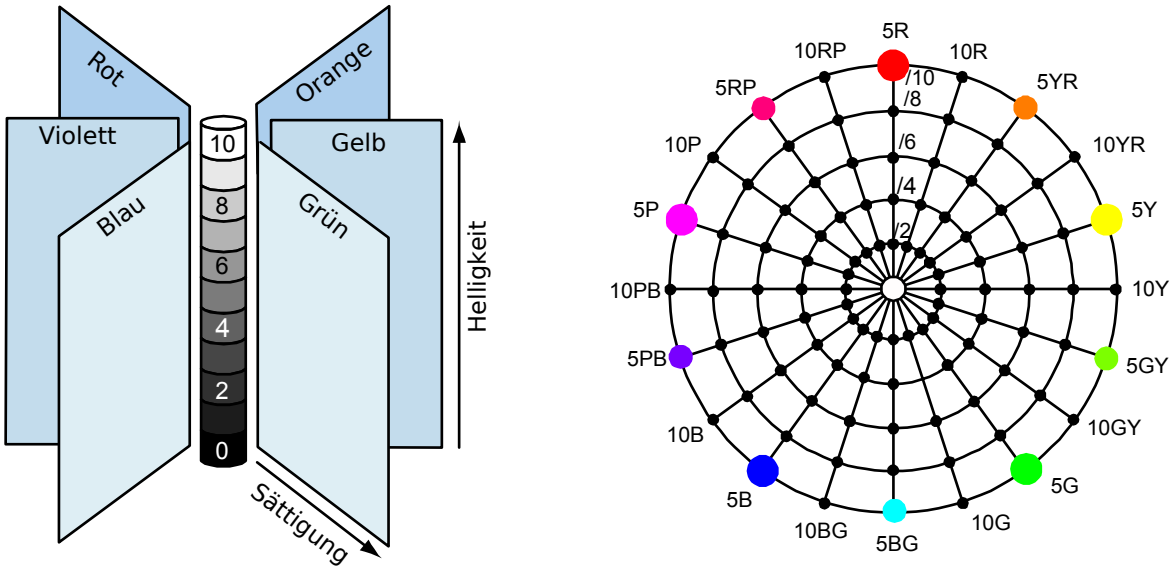


Abbildung 5-14: Aufbau des Munsell-Systems; Anordnung von Farbton, Buntheit und Sättigung in einem Zylinder (links), Einteilung des Farbkreises (rechts) (nach Schläpfer 1993)

Das Munsell Book of Colors umfasst lediglich einen Teil der im Munsell-System parametrisierten Farben, so werden statt der 100 möglichen nur 40 Farbtöne genutzt; die Sättigung wird in 2er-Schritten gezählt (Wyszecki & Stiles 1982). Weiterhin werden für das Munsell Book of Colors zwei Versionen unterschieden, mit glänzenden und mit matten Mustern. Für Erstere umfasst die Farbmustersammlung 1450 Muster, für Letztere 1277 Muster (Schläpfer 1993).

Jede Farbe im Munsell Book of Colors ist durch einen dreiteiligen Code eindeutig gekennzeichnet. So bezeichnet beispielsweise 5 R 8 / 2 ein mittleres Rot (H = 5 R) mit einer Helligkeit von 8 (V = 8) und einer Sättigung von 2 (C = 2) (Simon 2008).

5.5 Farbe in der Visualisierung

Für die graphische Darstellung von Daten durch Farben sind sowohl in der allgemeinen Visualisierung als auch in der Kartographie eine Reihe von Vorgehensweisen verfügbar. Dabei wird nicht nur die Frage betrachtet, welche Art Daten sich durch welche Farben kodieren lassen, sondern es werden auch Aspekte wie Farbsehschwächen, kulturspezifische Farbassoziationen oder Farbästhetik (vgl. Schumann & Müller 2000) berücksichtigt. In dieser Arbeit werden im Folgenden zunächst wesentliche Regeln der Kodierung durch Farbe beschrieben, die Einbeziehung von Farbsehschwächen erfolgt gesondert im nächsten Abschnitt. Kulturspezifische Farbassoziationen werden kurz angerissen, die Farbästhetik dagegen außer Acht gelassen.

Arbeiten, die die Anwendung von Farben in Karten betrachten, untersuchen meist thematische Karten (bspw. Choroplethenkarten). Gemäß der Unterscheidung im Abschnitt 4.4 zwischen Objekten und Klassen bzw. zwischen Variablen und Konstanten werden dabei Daten oder Objekte einer Klasse nach ihrer nominalen, ordinalen oder quantitativen Skala betrachtet und die Farbdimensionen im Sinne einer Variablen genutzt. Dementsprechend sind die verfügbaren Regeln zum Farbeinsatz vor diesem Hintergrund zu interpretieren. Im Abschnitt 4.4 wur-

de dagegen für diese Arbeit besonders die Unterscheidung von Objekten verschiedener Klassen und damit die Nutzung von Farbe als Konstante betont. Es wird sich allerdings zeigen, dass dieser Umstand durch die Darstellung nominaler Daten abgedeckt ist; die nominale Skala ist damit für diese Arbeit am bedeutendsten.

Die Methoden zur Auswahl von Farben werden in Abhängigkeit von möglichen Aufgaben oder Operationen, die anhand einer Karte oder Visualisierung gelöst bzw. durchgeführt werden sollen, unterschieden (vgl. Schumann & Müller 2000). Mögliche Aufgaben bei der Kartennutzung nennen z.B. Hake et al. (2002): Wahrnehmen (Erkennen, Identifizieren), Auszählen, Schätzen, Vergleichen und Deuten (Interpretieren, Analysieren). In dieser Arbeit ist vor allem die Wahrnehmung von Bedeutung. Hinzu kommen die von Schumann & Müller (2000) allgemein für Visualisierungen genannten Aufgaben „Übersicht“ und „Suchen“ (vgl. Anwendungsszenarien im Kapitel 1).

Ein anderer Aspekt bei der Nutzung von Farbe ist die Anzahl der Stufen, die höchstens unterschieden werden sollten, um die menschliche Wahrnehmung nicht zu überfordern. Schumann & Müller (2000) nennen maximal 5-8 Stufen für die Kodierung durch Helligkeit und Farbton bei parallelem Suchen. Schoppmeyer (1978) vergleicht Tonwertskalen verschiedener Arbeiten der Kartographie, die von 4 bis maximal 13 Stufen divergieren.

In dieser Arbeit wird dieser Aspekt der Anwendung untergeordnet: Erfahrungen, u.a. mit den im Abschnitt 3.2 vorgestellten Webportalen mit Raumbezug, haben gezeigt, dass die Zahlen für viele Aufgaben nicht einzuhalten sind: Typische farbige Karten, beispielsweise die in Abschnitt 3.2 genannten Kartenwerke, schöpfen eine Zahl von 5-8 Stufen bereits aus, eine Anreicherung der Karte durch zusätzliche punkt- und linienhafte Objekte überschreitet sie bei weitem.

5.5.1 Nominale Daten

Für die Darstellung nominaler Daten wurde in den Ausführungen des Abschnitts 4.2.4 aus den Dimensionen der Farbe der Farbton als sehr geeignet beschrieben. Dementsprechend werden bei der Kodierung solcher Daten Farben betrachtet, deren Farbtöne sich gut unterscheiden: Schumann & Müller (2000) nennen als verbreitete Strategien die Verwendung möglichst unterschiedlicher Farben aus dem HSV-Farbmodell[39] oder die Nutzung von Farben mit maximalem spektralen Abstand (violett, blau, grünblau, blaugrün, grün, gelbgrün, gelb, orange, orangerot, rot). Beide Ansätze bieten allerdings nur eine geringe Anzahl möglicher Farben.

Smallman & Boynton (1990) versuchen die ebenfalls geringe Zahl (6) von Farben, die in vorausgegangenen Arbeiten zur Unterstützung einer effizienten Lösung visueller Suchaufgaben bestimmt wurden, zu erhöhen. Dafür gehen sie von 7 sogenannten Grundfarben (basic colors) aus, die sprachlich prägnant benannt sind. Diese Liste erweitern sie um Mischungen

[39] Das HSV-Modell wurde in dieser Arbeit nicht gesondert beschrieben. Die Dimensionen dieses Modells sind die schon häufiger erwähnten Farbton (hue), Sättigung (saturation) und Helligkeit (value).

der Grundfarben (focal basic colors) auf 14 Farben. Die Hinzunahme von Weiß, Schwarz, Braun und einem leichten Grau würde die Liste auf 18 Farben erhöhen.

Die skizzierten Ansätze haben mit dieser Arbeit das Ziel der guten Unterscheidbarkeit von Farben gemeinsam. Allerdings gehen alle Skalen von frei wählbaren Farben aus, während hier auch gut unterscheidbare Farben zu bereits vorhandenen bestimmt werden sollen. Zu diesem letztgenannten Aspekt bietet die Arbeit von Nagy et al. (1990) einen allgemeineren Ansatz, der keine festen Farben, sondern Farbdifferenzen im CIELUV-Farbraum betrachtet. Für das parallele Suchen werden dort in Abhängigkeit von der Farbe 45 - 65 Längeneinheiten als gut unterscheidbar angegeben. Nagy et al. betonen allerdings auch die Abhängigkeit von der Größe des Farbreizes.

Die bisher genannten Arbeiten behandeln allgemein die Nutzung von Farben in der Darstellung. Im engeren Kontext der Kartographie sind einige weitere Rahmenbedingungen von Bedeutung (Robinson et al. 1995):

- Universelle oder kulturspezifische Konventionen ordnen bestimmten Objektarten bestimmte Farbtöne zu. Erstere betrifft das Blau von Wasserflächen und Gewässerlinien, letztere die Grüntöne für Vegetation, braun für Oberflächenstrukturen und Gelb für trockene Gebiete.
- Die Farbgebung in Karten ist durch bestimmte Vereinbarungen, insbesondere Zeichenvorschriften für Kartenwerke, restringiert.

5.5.2 Ordinale Daten

Die Kodierung ordinaler Daten erfolgt für die Operationen Suchen und Identifizieren ähnlich wie die der nominalen Daten (Schumann & Müller 2000). Die eigentliche Ordnung ist erst für die Aufgaben Übersicht und Vergleichen von Bedeutung (ebd.). Voraussetzung für den Ausdruck einer Ordnung ist, dass ein Betrachter für die genutzten Farben eine Ordnung empfindet (ebd.). Eine solche Empfindung kann besonders durch die Variation von Helligkeit und Sättigung erreicht werden (vgl. Abschnitt 4.2.4 zur Eignung der Variablen). Ausführlichere Informationen über so erhaltene *Farbskalen* werden im Zusammenhang mit dem spezielleren Fall der quantitativen Daten dargelegt.

5.5.3 Quantitative Daten

Die Quantität, die die Unterschiede innerhalb einer Ordnung zahlenmäßig angibt, erfordert dementsprechend, dass auch eine Metrik für die genutzten Farben empfunden wird. Im Abschnitt 4.2.4 divergierten die Einschätzungen zur Eignung der Farbdimensionen für diese Aufgabe. Insgesamt wurden die Helligkeit und Sättigung als besser geeignet eingeschätzt als der Farbton. Diese Einschätzung wird auch durch die Farbskalen deutlich, die Robinson et al. (1995) u.a. unterscheiden:

- Die *einfarbige Skala* entsteht bei gleichem Farbton durch die Variation von Sättigung und Helligkeit. Das Ergebnis ist ein Übergang von Weiß zu einem reinen Farbton.

- Eine *bi-polare Skala* verläuft von einem ersten gesättigten Farbton durch Variation von Sättigung und Helligkeit über Weiß zu einem zweiten gesättigten Farbton. Diese Skala eignet sich besonders zum Ausdruck eines positiv-negativ Trends.
- Eine *komplementäre Farbton-Skala* ist ebenfalls eine bi-polare Skala, an deren Enden zwei komplementäre Farben stehen.
- Die *partielle spektrale Farbton-Skala* besteht aus einem zusammenhängenden Teil der Farben des Spektrums; das gesamte Spektrum wird durch die *spektrale Skala* repräsentiert.
- Durch Überblenden zweier benachbarter Farben entsteht die *Farbübergangs-Skala*.
- Eine *Zweiteilige Skala* entsteht, wenn, beispielsweise ausgehend von Weiß, die Sättigungen zweier Farben jeweils erhöht und überlagert werden. Ergebnis ist eine Matrix von Farbwerten.

Bei der Anwendung von Farben auf quantitative Daten sind weiterhin gewisse Bedeutungen von Farben zu beachten. Beispielsweise wird ein dunkler Farbton vom menschlichen Betrachter als „mehr“ interpretiert, rot als „viel“ bzw. „hoch“ (Temperatur), blau als „wenig“ (vgl. Robinson et al. 1995). Ausführliche Betrachtungen zu diesem sehr umfangreichen Thema der Farbsymbolik bietet z.B. Heller (1999).

5.6 Personalisierung der Farbdarstellung

Einleitend zu diesem Kapitel wurden bereits die Abhängigkeit der Farbdarstellung vom Ausgabegerät und mögliche Wahrnehmungsschwächen eines Nutzers als limitierende Faktoren bei der Nutzung von Farbe angesprochen. Im Kontext der Kommunikationsmodelle des Abschnitts 4.1.2 stellen diese Faktoren Störungen des Kommunikationsprozesses dar, die ihren Grund in der Person des Nutzers und dem jeweils genutzten Ausgabemedium haben. Letzteres wird weiterhin von äußeren Bedingungen, in der Hauptsache dem Umgebungslicht, beeinflusst. Mit dieser Eingrenzung wesentlicher Störungen auf die Nutzerseite lassen sich durch eine Personalisierung Eigenschaften und Umgebungsbedingungen des Nutzers erfassen und bei der Nutzung der Farbe in der graphischen Darstellung berücksichtigen. Damit wird die Grundvoraussetzung einer effektiven und effizienten Kommunikation sichergestellt[40]: Eine klare und eindeutige Darstellung bzw. Erkennbarkeit der verwendeten Zeichen.

5.6.1 Farbwiedergabe und Color Management

Obwohl der Einsatz von Farbe durch Geräte der modernen Informationstechnologie mittlerweile zum Alltag gehört, ist eine professionelle Farbreproduktion aufgrund technischer Restriktionen nach wie vor eine anspruchsvolle Aufgabe: In digitalen Produktionsketten (der Ablauf von der Digitalisierung über die Bearbeitung bis zur Ausgabe eines Bildes) sind zum

[40] Die Effektivität einer auf Rot-Grün-Sehschwäche angepassten Kartengraphik wurde beispielsweise von Olson & Brewer (1997) durch Tests bestätigt.

einen die Farbumfänge der beteiligten Geräte nicht deckungsgleich, zum anderen werden Farbdaten gerätespezifisch interpretiert, d.h. die von einem Gerät ausgegebenen Farben entsprechen nicht den eingegebenen Ist-Werten (vgl. Fraser et al. 2005). Ziel eines *Color Managements* ist es, diese Probleme zu erfassen, zu beschreiben und in allen Arbeitsprozessen zu berücksichtigen (vgl. Homann 2007). Ein *Color Management System* (CMS) ist eine Software, die den Anwender dabei unterstützt.

5.6.1.1 Farbumfang des Ausgabegeräts

Der wesentliche limitierende Faktor in einer Produktionskette ist durch die unterschiedlichen Geräte-Gamuts, d.h. die Gesamtheit der von einem Ausgabegerät reproduzierbaren Farben gegeben (Simon 2008). Es treten in einem Reproduktionsprozess dann Probleme auf, wenn nicht alle Farben eines Bildes im Farbraum eines Ausgabegeräts darstellbar sind. Abbildung 5-15 zeigt beispielsweise die Farbumfänge eines Röhrenbildschirms und des Offsetdrucks in der Normfarbtafel.

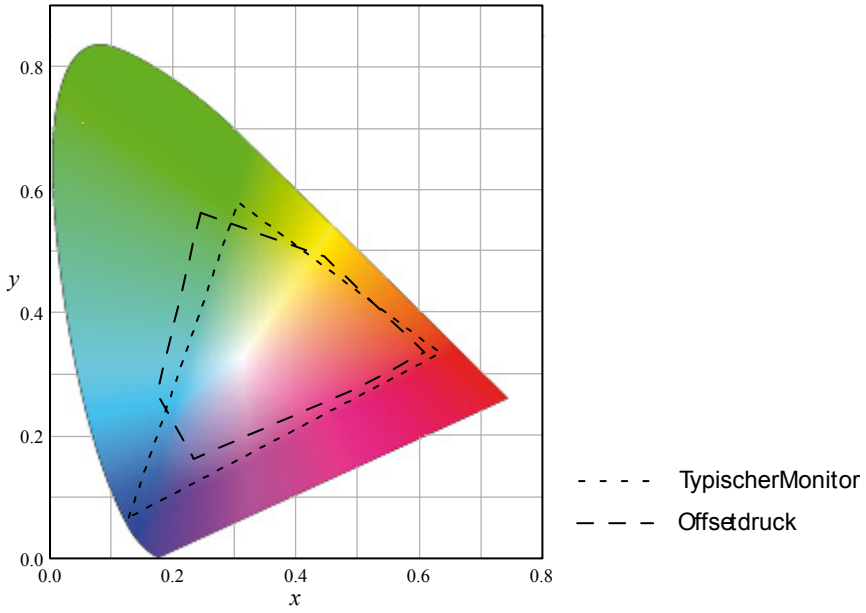


Abbildung 5-15: Farbumfang eines Monitors und des Offsetdrucks in der Normfarbtafel (nach Fraser et al. 2005)

Der Gamut des Bildschirms stellt ein Dreieck dar, das durch die Primärvalenzen aufgespannt wird. Der Farbumfang des Offsetdrucks bildet, abhängig von den Grundfarben Cyan, Magenta und Yellow, ein Sechseck, dessen Ecken diese Grundfarben und die subtraktiven Mischfarben erster Ordnung (Rot, Grün, Blau) sind (vgl. Abschnitt 5.3.3).

Color Management Systeme lösen dieses Problem der unterschiedlichen Farbumfänge durch Gamut Mapping: Die Farben eines Bildes werden an den Zielgamut angepasst (Simon 2008). Die einfachste Lösung ist die Abbildung der Farben außerhalb des Zielgamuts auf dessen Oberfläche, während Farben innerhalb des Gamuts unverändert bleiben (ebd.).

5.6.1.2 Austausch und Darstellung von Farbdaten

In den Ausführungen des Abschnitts 5.3 wurden verschiedene Farbräume als Bezugssysteme eingeführt, als Systemübergänge wurden affine oder projektive Transformationen angegeben. Diese rechnerische Exaktheit ist bei der Darstellung von Farben auf technischen Geräten hinfällig, so sind beispielsweise in einer digitalen Produktionskette unterschiedliche Geräte mit verschiedenen Farbräumen und einer jeweils gerätespezifischen Interpretation von Farbdaten beteiligt (vgl. Simon 2008):

- *Erfassung*: Geräte zur Erfassung von Daten (bspw. Scanner, Kameras) nutzen diverse Varianten von RGB-Farbräumen, sehr häufig Standard-RGB.
- *Display*: Diese Geräte, in der Hauptsache Bildschirme, nutzen als Farbraum meist Standard-RGB, höherklassige Modelle aber auch andere RGB-Farbräume.
- *Ausgabe*: Ausgabegeräte (bspw. Druck, Offsetdruck) verwenden im Normalfall den CMYK-Farbraum.

Color Management Systeme gleichen diese Unterschiede aus, indem sie zum einen Farben in systemunabhängigen Farbräumen (CIE XYZ oder CIELAB) angeben, zum anderen durch sogenannte Farbprofile eine Beziehung zwischen den Farben, die ein Gerät ausgeben sollte und den Farben, die es tatsächlich ausgibt, herstellen (Homann 2007). Ein Farbprofil ist dabei im Kern nichts anderes als eine Lookup-Tabelle, die Farbwerten im Gerätesystem (RGB oder CMYK) korrespondierende Werte in einem der genannten systemunabhängigen Farbräumen zuordnet (Fraser et al. 2005). Die Festschreibung erfolgt als ICC-Profil, dessen Struktur als offenes Format vom International Color Consortium (ICC)[41] spezifiziert ist (ICC 2004).

Um eine definierte Farbausgabe eines Gerätes herzustellen, sind zwei Prozesse von Bedeutung (Fraser et al. 2005):

- Eine konstante Farbwiedergabe wird durch *Kalibrierung* erreicht, die das Verhalten eines Gerätes ändert. Beispielsweise besteht die Kalibrierung eines Monitors aus Einstellungen des Anwenders und der Berücksichtigung von Farbprofilen durch das Betriebssystem oder Bildbearbeitungsprogramme.
- Das eigentliche Farbprofil wird durch eine *Profilierung* oder *Charakterisierung* erhalten. Diese stellt einen reinen Messvorgang dar: Das zu profilierende Gerät erzeugt bestimmte Farbwerte, deren tatsächliche Wiedergabe messtechnisch erfasst wird.

5.6.1.3 Kalibrierung von Drucker und Monitor

In dieser Arbeit, die ja einen nicht-professionellen, web-gestützten Kartengebrauch fokussiert (vgl. Abschnitt 1.2), ist vor allem die Ausgabe am Bildschirm von Bedeutung, in Einzelfällen kommt eine Druckausgabe infrage. Um sicherzustellen, dass sämtliche Farben einer Karte immer entsprechend dem Zweck ihrer Nutzung dargestellt werden, sollten die Geräte jedes

[41] Das ICC ist eine Vereinigung namhafter Hersteller von Geräten, die Farben nutzen; näheres unter www.color.org (Zuletzt geprüft am 20.11.2008).

Anwenders Farben zumindest ohne grobe Verfälschungen wiedergeben. Wünschenswert wäre darüber hinaus zweifelsohne, dass sämtliche Geräte kalibriert sind. Im Folgenden wird deshalb auf die Einstell- und Kalibrierungsmöglichkeiten von Bildschirmen und Druckern eingegangen. Für Letztere bedeutet dies einen erhöhten Aufwand und es ist fraglich, ob nichtprofessionelle Anwender diesen in Kauf nehmen. Die Kalibrierung von Druckern wird deshalb nur kurz skizziert, bevor detaillierter auf Einstellung und Kalibrierung von Bildschirmen eingegangen wird. Zu diesen sei noch erwähnt, dass deren Kalibrierung/Profilierung natürlich streng genommen eine Kalibrierung/Profilierung von Bildschirm und Graphikkarte darstellt, dieser Umstand ist hier allerdings ohne Bedeutung.

Druckerkalibrierung

Eine Druckausgabe ist immer von mehreren Parametern abhängig. Aus diesem Grunde muss eine Profilierung den gesamten Druckprozess berücksichtigen (Homann 2007). Dieser hängt u.a. ab vom Druckverfahren, dem verwendeten Papier, der Druckfarbe und von der Stärke des Farbauftrags (ebd.). Die Profilierung erfolgt, indem ein Testchart mit einer Vielzahl repräsentativer Farben (mindestens 1000, (Simon 2008)) gedruckt und mit Hilfe eines Spektralphotometers ausgemessen wird (Homann 2007). Die Ergebnisse werden als ICC-Profil gespeichert und bei der Druckausgabe berücksichtigt.

Monitorkalibrierung

Beschreibungen zur Kalibrierung von Monitoren basieren meist auf den technischen Grundlagen von Kathodenstrahlmonitoren (Röhrenbildschirmen). Die Erzeugung von Farben auf diesen Bildschirmen wurde bereits im Abschnitt 5.3.2 (RGB- bzw. sRGB-Farbraum) skizziert. Das Bild eines solchen Monitors hat einen rasterförmigen Aufbau und setzt sich vertikal aus Zeilen zusammen, die wiederum horizontal in Bildpunkte oder Pixel unterteilt sind (Homann 2007). Jeder Bildpunkt besteht aus drei Phosphoren, die durch Zuführen von Energie zum Leuchten in einer der drei Primärvalenzen Rot, Grün oder Blau angeregt werden (Lang 1995). Dem Betrachter erscheint dann die additive Mischung dieser Farben im RGB-Farbraum (vgl. Abschnitt 5.3.2). Flüssigkristallbildschirme versuchen das Farbverhalten von Röhrenmonitoren nachzuahmen (Homann 2007). Die Bild- bzw. Farberzeugung erfolgt durch eine Schicht Flüssigkristalle, deren Durchlässigkeit für das Licht einer Hintergrundbeleuchtung durch eine elektrische Spannung steuerbar ist (Fraser et al. 2005).

Für die Farbdarstellung eines Monitors sind außer den Primärvalenzen auch der Weiß- und Schwarzpunkt von Bedeutung. Die Einstellung des Weißpunktes gibt an, ob dargestelltes Weiß ins Warme oder Kalte tendiert (Homann 2007). Bei einem wärmeren Weiß, (eine Farbtemperatur von 5000 oder 5500 K) erscheint ein Monitor eher gelblich, bei einem Weiß von 6500 bis 9500 K eher bläulich (vgl. Fraser et al. 2005). Für Monitore, auf denen Farben für Internetanwendungen beurteilt werden, empfiehlt Homann (2007) eine Farbtemperatur von 6500 K, dies entspricht der Farbe der Normlichtart D65.

Der Schwarzpunkt gibt an, durch welche Farbwerte das Schwarz eines Gerätes beschrieben wird (Fraser et al. 2005). Im Falle eines Monitors muss dies nicht zwingend durch $S_0 = (R_0,G_0,B_0) = (0, 0, 0)$ gegeben sein, sondern in Abhängigkeit von der eingestellten Helligkeit

und dem Umgebungslicht können auch dunkle Grautöne bereits schwarz erscheinen. Auf Röhrenmonitoren ist der Schwarzpunkt durch den Helligkeitsregler einstellbar. Auf TFT-Monitoren ist meist nur die Helligkeit der Hintergrundbeleuchtung regelbar, diese beeinflusst aber ebenso die Leuchtdichte für Weiß (Fraser et al. 2005). Letztere wiederum ist bei Röhrenmonitoren separat durch den Kontrast steuerbar.

Ein weiteres Charakteristikum von Monitoren wurde bereits im Abschnitt 5.3.2 eingeführt, der Gammawert. Die zugehörige Gammakurve beschreibt über den Zusammenhang zwischen Beschleunigungsspannung und Leuchtdichte den Helligkeitsverlauf von dunkel nach hell (vgl. Homann 2007). Für größere Gamma verdunkeln sich die mittleren Töne eines Monitors (vgl. Abbildung 5-16).

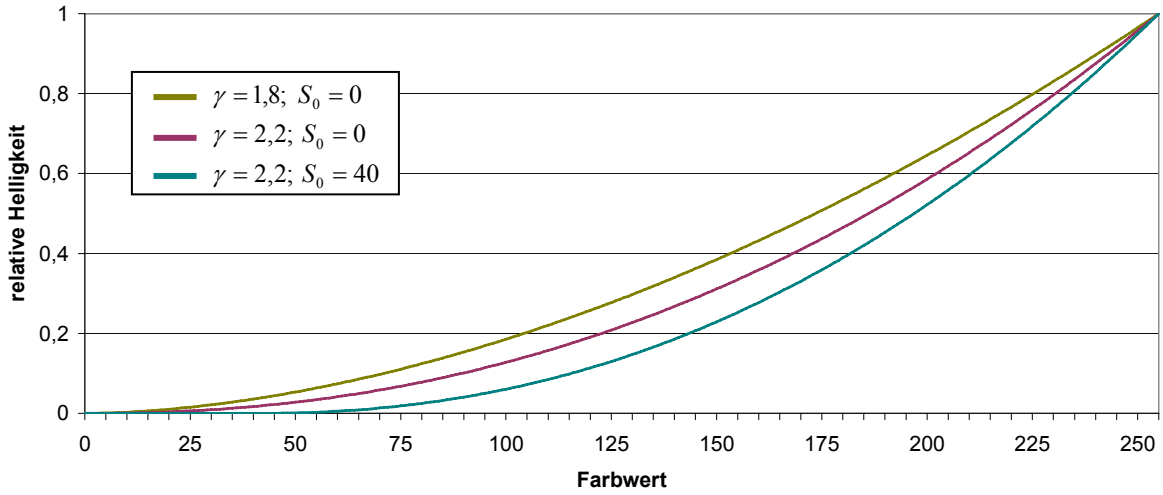


Abbildung 5-16: Beispiele für den Verlauf der Leuchtdichte bei verschiedenen Gammawerten und Schwarzpunkten

Der allgemeine Zusammenhang zwischen Spannung und Leuchtdichte wurde bereits durch Formel 5.5 angegeben. Mit den dort eingeführten Vereinfachungen und unter Berücksichtigung, dass bei 8 Bit-Kodierung die angelegte Spannung für jeden Farbkanal über die Farbwerte $R,G,B = 0\ldots255$ gesteuert wird, ergibt sich beispielsweise für den roten Farbkanal die Leuchtdichte L_R durch (Brainard et al. 2002):

$$L_R = \begin{cases} \left(\frac{R - R_0}{255 - R_0}\right)^{\gamma} & \text{für } R > R_0 \\ 0 & \text{sonst .} \end{cases} \qquad (5.6)$$

Die Formeln des grünen und blauen Kanals ergeben sich analog. Abbildung 5-16 zeigt den Verlauf dreier Gammakurven bei unterschiedlichen Gamma und Schwarzpunkten. Für das Color Management wird für die meisten Anwendungen eine Kalibrierung mit einem Gamma von 2,2 empfohlen (Fraser et al. 2005, Homann 2007).

Die eigentliche Kalibrierung und Profilierung lässt sich grob in instrumentenbasierte und visuelle Methoden unterscheiden (vgl. Fraser et al. 2005), die im Folgenden näher ausgeführt werden.

Vorab sind bei jeder Kalibrierung einige Grundbedingungen zu beachten: Der Monitor sollte unter stabilen äußeren Bedingungen (Umgebungslicht) kalibriert werden und zunächst mindestens 30 Minuten aufwärmen, die Farbtiefe mindestens 24 Bit umfassen (Fraser et al. 2005). Weiterhin sollte die Bildschirmoberfläche gesäubert sein.

Instrumentengestützte Kalibrierung und Profilierung: Die folgende Beschreibung gibt in Kürze den Kalibrierungs- und Profilierungsprozess mit der Software „eye-one match 3“ (Version 3.5) und dem Farbmessgerät „eye-one display LT“ der Firma Pantone/GretagMacbeth[42] wieder. Vergleichbare Systeme arbeiten ähnlich, allgemeiner gehaltene Beschreibungen des Prozesses finden sich z.B. in Fraser et al. (2005).

Nach dem Start der Kalibrierungssoftware wird der Anwender mit Hilfe eines interaktiven Assistenten durch die einzelnen Schritte des Kalibrierungsvorgangs geführt. Zunächst muss das zu kalibrierende Gerät (Röhrenmonitor, TFT-Monitor) ausgewählt und der Messkopf auf dem Bildschirm platziert werden. Anschließend fordert das System zur Auswahl der Farbtemperatur des gewünschten Weißpunkts auf. Im Verlauf der eigentlichen Kalibrierung muss dann zunächst interaktiv der Kontrast eingeregelt werden, d.h. das System misst den jeweils aktuellen Wert und erwartet Änderungen, bis ein Zielwert erreicht ist. Im nächsten Schritt wird durch Einregelung der RGB-Kanäle der tatsächliche Weißpunkt auf den gewünschten Weißpunkt eingestellt. Dies kann entweder durch Auswahl einer Voreinstellung oder, falls dies das jeweilige Monitormodell erlaubt, durch individuelle Einstellung der Farbkanäle erfolgen. Letzteres geschieht nach dem gleichen Prinzip wie die Einregelung des Kontrastes.

Nach diesen Einstellungen führt das System die Profilierung durch, indem auf dem Monitor definierte Farben dargestellt und durch den Messkopf erfasst werden. Der gesamte Kalibrierungsvorgang ist mit der Speicherung der gemessenen Daten als ICC-Profil beendet. Im Fall des Betriebssystems Windows erfolgt die Speicherung direkt im betreffenden Systemordner und das Farbprofil wird durch die Farbverwaltung der Graphikkarte berücksichtigt.

Obwohl Geräte zur Erfassung von Monitorprofilen in der Einstiegsklasse mit etwas über 100 € mittlerweile recht günstig verfügbar sind, lässt sich kaum voraussetzen, dass eine Mehrheit von Nutzern derart kalibrierte Monitore nutzt.

Visuelle Kalibrierung: Eine visuelle Kalibrierung lässt sich dagegen deutlich einfacher und ohne Einsatz spezieller Instrumente durchführen. Aufgrund der nicht sehr guten Ergebnisse ist eine solche Kalibrierung aber nicht empfehlenswert (Fraser et al. 2005) und lediglich als einfaches Hilfsmittel anzusehen. Die Methodik basiert auf der Funktionsweise von Röhrenbildschirmen, ist aber auch auf TFT-Monitore anwendbar. Voraussetzung für eine visuelle Kalibrierung sind ebenso die oben genannten Grundbedingungen (Fraser et al. 2005), allerdings ist der Einfluss der Umgebung ungleich kritischer, die Kalibrierung sollte deshalb möglichst in einem abgedunkelten Raum erfolgen (ebd.).

[42] http://www.pantone.com (Zuletzt geprüft am 21.11.2008)

Abbildung 5-17: Bildschirmfoto der Gamma-Einstellung mit Adobe Gamma

Das allgemeine Vorgehen bei der visuellen Kalibrierung beschreiben beispielsweise Fraser et al. (2005). In dieser Arbeit orientieren sich die Ausführungen am Programm Adobe Gamma (Adobe 2007), einem visuellen Kalibrierungsprogramm, das standardmäßig zusammen mit dem Bildbearbeitungsprogramm Adobe Photoshop installiert wird. Der Adobe Gamma-Assistent führt den Anwender in mehreren Schritten durch alle benötigten Einstellungen. Zunächst muss der Kontrast – sofern der jeweilige Monitor dies zulässt – auf die höchste Einstellung gesetzt werden, die Helligkeit muss über die visuelle Einschätzung eines Graustufenbildes eingeregelt werden. Im nächsten Schritt ist die Angabe der Phosphorfarben erforderlich, zur Auswahl stehen dabei mehrere vorgegebene Profile oder die Eingabe eigener Farbwerte. Der nächste Schritt, die Gamma-Einstellung, kann entweder für ein einzelnes Gamma oder für die drei getrennten Gammawerte des roten, grünen und blauen Farbkanals erfolgen (Abbildung 5-17). Im Fall der getrennten Gammawerte zeigt der Assistent drei Testfelder an, die im Hintergrund ein Muster enthalten. Dieses Muster besteht aus gleich breiten horizontalen schwarzen und farbigen (roten, grünen, blauen) Linien. Die farbigen Linien stellen dabei jeweils den reinsten Farbwert dar. Mit etwas Betrachtungsabstand zu diesem Muster verschwimmen die Linien zu jeweils gleichmäßig farbigen Flächen mit einem im Vergleich zur reinen Farbe halbierten Helligkeitswert. Über diesem Muster ist jeweils eine rote, grüne oder blaue Volltonfarbe sichtbar, die über Schieberegler in ihrer Helligkeit variierbar sind. Ein Anwender muss die Helligkeit der Volltonfläche nun möglichst gut mit der Helligkeit des Musters im Hintergrund in Einklang bringen.

Im letzten Schritt erfolgt die Einstellung des Weißpunktes. Hier hat der Anwender die Wahl zwischen verschiedenen vorgegebenen Farbtemperaturen und einer visuell bestimmten Eingabe. Die Ergebnisse aller Einstellungen werden von Adobe Gamma in ein ICC-Profil umgesetzt, im Systemordner gespeichert (Windows) und in der Farbverwaltung des Monitors als Standardprofil festgelegt.

Für Anwender, denen Adobe Gamma nicht zur Verfügung steht, existieren kostenlos nutzbare Alternativen (z.B. QuickGamma[43]). Damit ist das eben beschriebene Vorgehen der visuellen Kalibrierung ohne besonderen (finanziellen) Aufwand durchführbar.

Eine visuelle Kalibrierung in Form einer Gamma-Korrektur, die lediglich Einstellungen von Seiten des Anwenders und keinerlei eigene Software oder Systemeingriffe benötigt, ist in einer Client-Server-Konstellation möglich. Dazu werden nach Umformung der Formel 5.6 serverseitig die auszugebenden Farbwerte auf bestimmte Geräteeigenschaften umgerechnet, d.h. es wird jeweils derjenige Farbwert berechnet, der auf einem bestimmten Gerät eine gewünschte Leuchtdichte hervorruft (vgl. Brainard et al. 2002). Dazu muss ein Anwender zunächst den Schwarzpunkt $S_0 = (R_0,G_0,B_0)$ und die aktuellen Gamma-Einstellungen schätzen. Ersteres kann auf einer Skala von Graustufen (Abbildung 5-18) erfolgen, indem der niedrigste, noch als grau erkennbare, Balken auswählt wird.

Abbildung 5-18: Graustufen zur Festlegung des Schwarzpunktes

Weiterhin muss der Anwender, analog zu Adobe Gamma, die Helligkeit einer roten, grünen und blauen Volltonfläche auf die Helligkeit eines gemusterten Hintergrunds abgleichen. Mit den so erhaltenen Farbwerten $R_\gamma, G_\gamma, B_\gamma$ lässt sich dann nach Umstellung der Formel 5.6 zunächst für jeden Farbkanal ein Gammawert berechnen, beispielsweise ergibt sich für Rot:

$$\gamma_R = \frac{\log L}{\log\left(\frac{R_\gamma - R_0}{255 - R_0}\right)} = \frac{\log 0{,}5}{\log\left(\frac{R_\gamma - R_0}{255 - R_0}\right)}.$$

Für einen beliebigen Farbton mit Rotwert R', der für ein Gamma mit Standardwert (z.B. 2,2) und einem Schwarzwert von $R_0 = 0$ eine Leuchtdichte

$$L_{soll} = \left(\frac{R' - R_0}{255 - R_0}\right)^\gamma$$

hervorruft, ergibt sich dann ein verbesserter Farbwert nach (vgl. Brainard et al. 2002):

$$R_{neu} = (255 - R_0) L_{soll}^{\frac{1}{\gamma_R}} + R_0.$$

[43] http://www.quickgamma.de (Zuletzt geprüft am 21.11.2008)

Ein solches Vorgehen ist sicherlich nur eine Behelfsmaßnahme, allerdings lassen sich auf diese Weise eine geringe Monitorhelligkeit bzw. ein helles Umgebungslicht und eine Farbstichigkeit feststellen und zum großen Teil ausgleichen.

5.6.2 Farbfehlsichtigkeit

Die uneingeschränkte Nutzung von Farbe in graphischen Darstellungen berücksichtigt nicht, dass Farben nicht von allen Menschen gleich empfunden werden, sondern ein relativ hoher Prozentsatz von ca. 8% der männlichen und 0,4 % der weiblichen Bevölkerung von sogenannten Farbsinnstörungen oder Farbfehlsichtigkeit betroffen ist (vgl. Tabelle 5-4). Diese Störungen können von leichten Anomalien bis hin zur völligen Farbenblindheit reichen. Farbfehlsichtigkeit tritt – abgesehen von der kompletten Farbenblindheit – als Blindheit in Bezug auf die Rot-, Grün- oder Blauwahrnehmung auf. Der Grund für diese verschiedenen Formen der Fehlsichtigkeit liegt im Aufbau und der Funktion des Auges begründet: Das Auge besitzt für die Farbwahrnehmung drei verschiedene Arten von Rezeptoren, deren Empfindlichkeit sich auf verschiedene Bereiche des Spektrums beschränkt (Simon 2008). Je nach dem Wellenlängenbereich, in dem das Maximum der jeweiligen Sensitivität liegt, werden unterschieden (ebd.):

- L, long-wavelength (Maximum im roten Bereich bei 558,4 nm),
- M, middle-wavelength (Maximum im grünen Bereich bei 530,8 nm),
- S, short-wavelength (Maximum im blauen Bereich bei 419,0 nm).

Jede Farbvalenz kommt als Mischung der Signale dieser drei Arten von Rezeptoren zustande (Schläpfer 1993).

Die verschiedenen Arten der Farbfehlsichtigkeit kommen durch Störung eines Rezeptors oder die Funktionsuntüchtigkeit eines oder mehrerer Rezeptoren zustande. Im Einzelnen werden differenziert (vgl. Schläpfer 1993, Schumann & Müller 2000):

- Die *Anomale Trichromasie* bezeichnet eine gestörte Funktion eines Farbrezeptors; in Abhängigkeit vom Rezeptor gilt:
  - *Protanomalie*: Schwäche der Rot-Wahrnehmung,
  - *Deuteranomalie*: Schwäche der Grün-Wahrnehmung,
  - *Tritanomalie*: Schwäche der Blau-Wahrnehmung.
- *Dichromasie* bezeichnet das Zweifarbensehen bei einem kompletten Ausfall eines Rezeptors, dabei werden wiederum unterschieden:
  - *Protanopie*: Rot-Blindheit,
  - *Deuteranopie*: Grün-Blindheit,
  - *Tritanopie*: Blau-Blindheit.
- *Monochromasie* bezeichnet den Ausfall von zwei Rezeptoren; dieser sehr seltene Fall ist allerdings medizinisch kaum beschrieben (Schläpfer 1993).

- Als *Achromasie* wird der Ausfall aller Rezeptoren und damit die völlige Farbenblindheit bezeichnet.

Tabelle 5-4 fasst die verschiedenen Formen der Farbfehlsichtigkeit und die Häufigkeit ihres Auftretens zusammen.

Bezeichnung der Fehlsichtigkeit	**Rezeptorfunktion**			**Häufigkeit in Prozent**	
	Rot	**Grün**	**Blau**	**Männer**	**Frauen**
Anomale Trichromasie					
Protanomalie	(+)	+	+	1	0
Deuteranomalie	+	(+)	+	4,9	0,38
Tritanomalie	+	+	(+)	<0,001	<0,001
Dichromasie					
Protanopie	-	+	+	1	0,02
Deuteranopie	+	-	+	1,1	0,01
Tritanopie	+	+	-	0,002	0,001
Achromasie	-	-	-	0,003	0,002

Tabelle 5-4: Formen und Häufigkeit der Farbfehlsichtigkeit (nach Schläpfer 1993); „+“ bezeichnet die volle, „(+)“ die eingeschränkte Funktion eines Rezeptors, „-“ den Ausfall eines Rezeptors

Die unterschiedliche Häufigkeit der Farbfehlsichtigkeit bei Männern und Frauen hat ihren Grund darin, dass die Gene, die die Entwicklung der Farbrezeptoren steuern, mit dem X-Chromosom vererbt werden (Schläpfer 1993). Da das weibliche Erbgut zwei X-Chromosomen enthält, das männliche lediglich eins, ist es für Frauen wahrscheinlicher, Erbgut ohne Genstörung zu erhalten (ebd.).

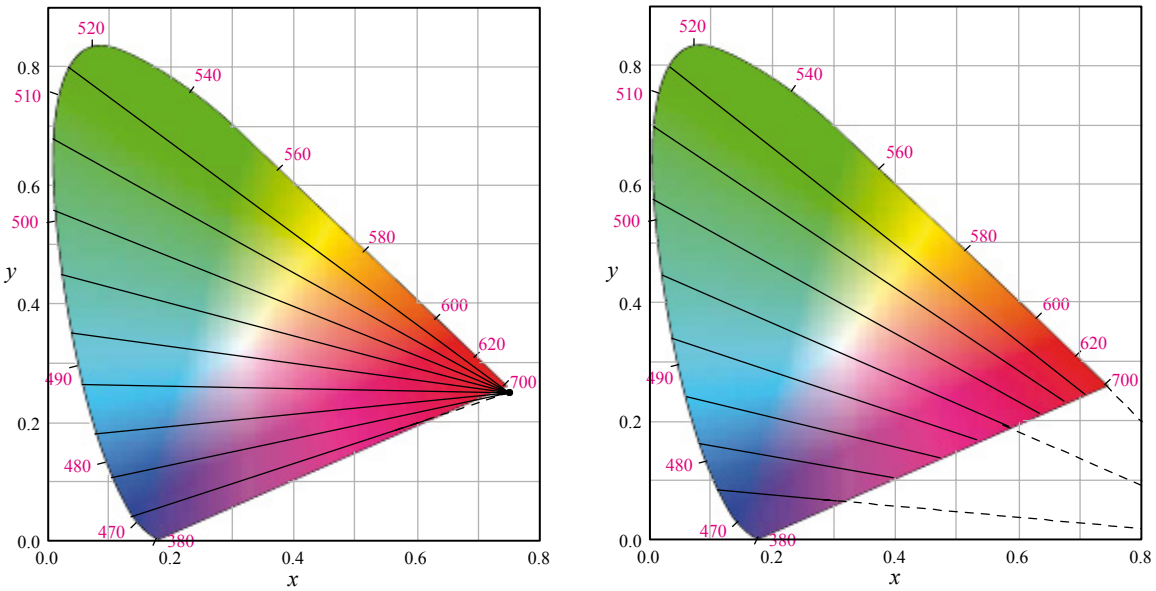


Abbildung 5-19: Verwechslungsgeraden für Protanope (links) und Deuteranope (rechts) in der Normfarbtafel (nach Wyszecki & Stiles 1982)

Die drei Formen der Dichromasie lassen sich in der Normfarbtafel durch sogenannte *Verwechslungsgeraden* darstellen (Schumann & Müller 2000). Farben, die auf jeweils der gleichen Geraden liegen, können durch die entsprechenden Dichromaten nicht unterschieden werden (Wyszecki & Stiles 1982). Abbildung 5-19 gibt links Verwechslungsgeraden für Pro-

tanope wieder, rechts für Deuteranope. Abbildung 5-20 stellt Verwechslungsgeraden für Tritanope dar. In allen drei Fällen schneiden sich die Geraden jeweils in einem ausgezeichneten Punkt, dem *Verwechslungspunkt* (Wyszecki & Stiles 1982). Zu beachten ist, dass die in den Abbildungen dargestellten Geraden lediglich eine Auswahl darstellen: Jede Gerade in der Normfarbtafel, die durch den Verwechslungspunkt verläuft, ist eine Verwechslungsgerade.

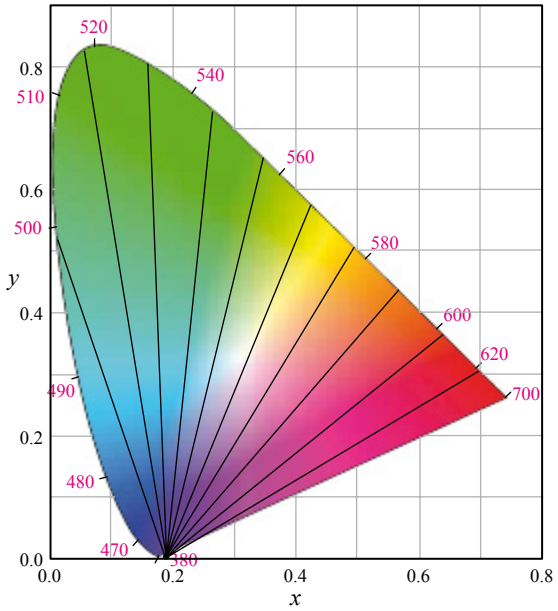


Abbildung 5-20: Verwechslungsgeraden für Tritanope in der Normfarbtafel (nach Wyszecki & Stiles 1982)

Für die Verortung der Verwechslungspunkte in der Normfarbtafel wird in einer Vielzahl von Arbeiten von unterschiedlichen Werten ausgegangen (vgl. Meyer & Greenberg 1988); Tabelle 5-5 gibt beispielhaft die Normfarbwerte nach Wyszecki & Stiles (1982) an.

Bezeichnung der Fehlsichtigkeit	Normfarbwerte des Verwechslungspunkts	
	x	y
Protanopie	0,747	0,253
Deuteranopie	1,080	-0,080
Tritanopie	0,171	0,000

Tabelle 5-5: Normfarbwerte der Verwechslungspunkte für die verschiedenen Dichromaten (Werte nach Wyszecki & Stiles 1982)

Aus Tabelle 5-4 ging hervor, dass das Auftreten von Tritanomalie und Tritanopie vernachlässigbar ist. Deshalb werden in den folgenden Ausführungen – sofern die Differenzierung nach der Art der Dichromasie von Bedeutung ist – nur die sehr viel häufiger auftretenden Formen der Rot- und Grün-Blindheit (Protanomalie, Deuteranomalie, Protanopie, Deuteranopie) betrachtet. Abbildung 5-19 illustriert anhand der Richtungen der Verwechslungsgeraden, dass diese Formen in ihrer Auswirkung recht ähnlich sind und im Wesentlichen rote und grüne Farbwerte betreffen.

Die Berücksichtigung von Farbsehschwächen in der graphischen Gestaltung erfordert für Normalsichtige offensichtlich Möglichkeiten, das eingeschränkte Sehvermögen bzw. den

damit eingeschränkten Farbraum zu simulieren. Eine einfache Lösung besteht darin, in Bildbearbeitungsprogrammen angepasste Farbpaletten (z.B. Rigden 1999) oder entsprechende Simulationsprogramme zu nutzen. Beispiele für solche Programme sind Visolve[44] und Color Oracle[45]. Zu Letzterem findet sich Näheres in Jenny & Kelso (2007).

Die Umrechnung von Farben auf den Farbraum von Dichromaten erfolgt meist unter Nutzung des SML-Raums, einem dreidimensionalen, rechtwinkligen Raum, der durch die oben genannten Farbrezeptoren des menschlichen Auges (S, M, L) aufgespannt wird (vgl. z.B. Meyer & Greenberg 1988).

Abbildung 5-21 verdeutlicht die Zusammenhänge zwischen Normfarbtafel, Normvalenzsystem und SML-Raum am Beispiel der Protanopie. Links ist die Repräsentation der Verwechslungsgeraden in der Normfarbtafel samt ihrer Einbettung in den XYZ-Normfarbraum abgebildet. Die Farbverwechslung wird im Normfarbraum durch die dargestellten Ebenen repräsentiert. Aus der Abbildung rechts ist die Situation nach linearer Transformation in das SML-System ersichtlich. Der Raum reduziert sich offensichtlich bei Ausfall eines Rezeptors auf eine Ebene, im Fall der Protanopie auf die SM-Ebene. Die Ebenen der Farbverwechslung stehen in diesem Fall senkrecht auf der SM-Ebene und verlaufen durch die L-Achse. Die Farbtafel reduziert sich auf die Gerade $S + M = 1$.

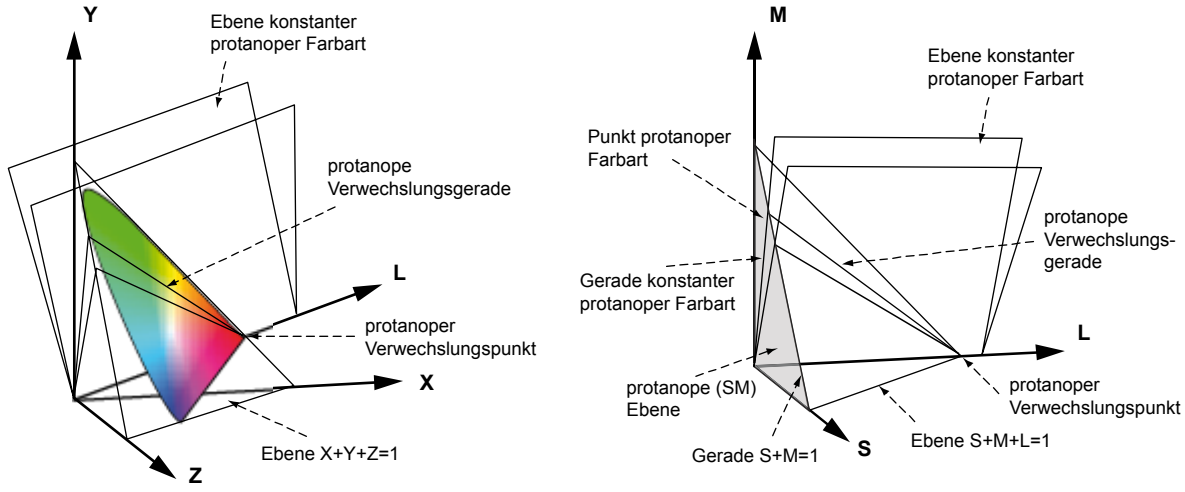


Abbildung 5-21: Modell der Farbwahrnehmung durch Protanope im XYZ- und SML-Raum (nach Meyer & Greenberg 1988)

Algorithmen, die durch Transformation in den SML-Raum RGB-Farbwerte eines Normalsichtigen auf die Wahrnehmung eines Dichromaten umrechnen, werden beispielsweise in Brettel et al. (1997) und Viénot et al. (1999) beschrieben. Eine kurze Zusammenfassung der benötigten Formeln findet sich im Anhang A.3.2 dieser Arbeit.

Abbildung 5-22 zeigt abschließend die Simulation der Normfarbtafel für Protanope und Deuteranope. Besonders deutlich werden aus dieser Darstellung die Ähnlichkeit der Farbseh-

[44] http://www.ryobi-sol.co.jp/visolve/en/ (Zuletzt geprüft am 27.11.2008)

[45] http://colororacle.cartography.ch (Zuletzt geprüft am 27.11.2008)

schwäche und die Dominanz von blauen und gelben Farbtönen im Farbraum dieser Sehschwächen.

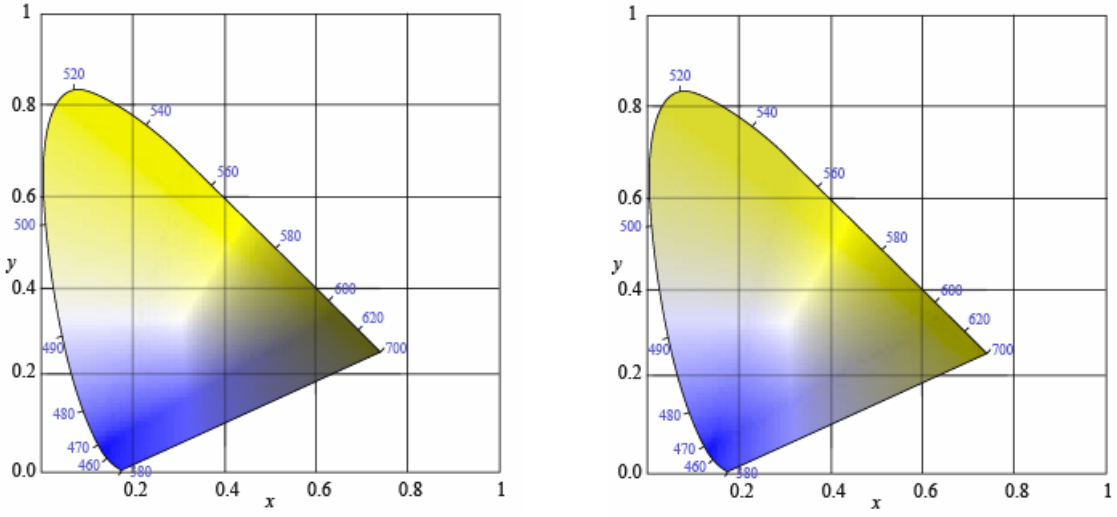


Abbildung 5-22: Normfarbtafel, simuliert für Protanope (links) und Deuteranope (rechts) (erstellt mit Color Oracle)

5.6.3 Personalisierte Karten

Die Anpassung der graphischen Darstellung einer Karte auf die Geräte- und Seheigenschaften eines Anwenders erfordert zum einen die Erfassung und Vorhaltung dieser Eigenschaften, zum anderen die Berücksichtigung bei der graphischen Umsetzung.

Das Vorhalten dieser Informationen ist im Kontext des WWW mit Hilfe einer Personalisierung offensichtlich unschwer zu erreichen. Eine Vielzahl von Webportalen bietet bzw. erwartet von ihren Nutzern die Einrichtung von Nutzerkonten und die Eingabe persönlicher Daten (vgl. Abschnitt 3.1). Im vorliegenden Fall muss die Erhebung der persönlichen Daten dann im Wesentlichen eine Strategie zur Erfassung von Geräteeigenschaften und Sehschwächen einschließen. Diese Strategie kann im einfachsten Fall aus Fragen nach Kalibrierungszustand des Monitors und nach den Seheigenschaften des Nutzers bestehen. Eine bessere Einschätzung wird aber sicherlich durch genaue Vorgaben zur Monitoreinstellung und/oder Tests erreicht.

Ein mögliches Vorgehen beschreiben Kuchenbecker et al. (2007) mit der Untersuchung eines webbasierten Farbsehtests für Screeninguntersuchungen des Farbensehens: Zu Beginn des Testablaufs wurden die Teilnehmer aufgefordert, den Raum abzudunkeln und bestimmte Monitoreinstellungen vorzunehmen (Weißpunkt, Leuchtdichte der Farbkanäle, Helligkeit, Farbtiefe). Mit einer visuellen Kalibrierung anhand einer Testgraphik (Abbildung 5-23) musste der Kontrast eingestellt werden, außerdem sollte eine Größenkalibrierung vorgenommen werden. Im weiteren Verlauf erfolgte dann die Prüfung des Farbensehens mit Hilfe von Velhagen-Broschmann- und Ishihara-Farbtafeln (Kuchenbecker et al. 2007, für vertiefende Ausführungen zu Farbsehtests siehe bspw. auch Dain (2004)).

Die Untersuchung dieses Ablaufs ergab, dass sich ein solcher Test durchaus für Siebtests des Farbensehens im WWW eignet (Kuchenbecker et al. 2007).

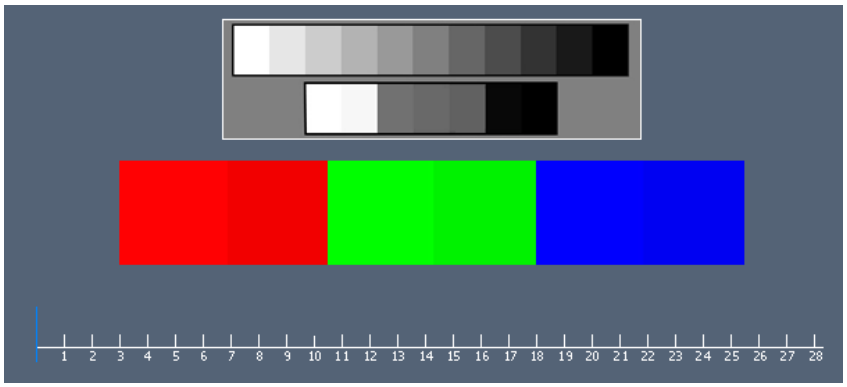


Abbildung 5-23: Testgraphik zur visuellen Kalibrierung eines Monitors (Quelle: http://www.farbsehtest.de, zuletzt geprüft am 26.11.2008)

Die Adaption dieses Tests würde für die Ziele dieser Arbeit sicherlich ausreichen. Darüber hinaus ist eine Erweiterung um die im Abschnitt 5.6.1.3 beschriebene Möglichkeit der visuellen Kalibrierung in einer Client-Server-Umgebung denkbar. Gerade für Monitore die keine Einstellung von Weißpunkt, Leuchtdichte, Kontrast und Helligkeit zulassen, würde sich auf diese Weise eine Einschätzung des Gerätezustands ermitteln lassen.

Für die Berücksichtigung der Sehschwächen in Karten schlagen Jenny & Kelso (2007) die Auswahl eindeutiger Farben, die Nutzung redundanter Variablen und die Beschriftung von Objekten vor. Abbildung 5-24 zeigt am Beispiel linienhafter Objekte eine Gegenüberstellung und Bewertung verschiedener Darstellungsmöglichkeiten.

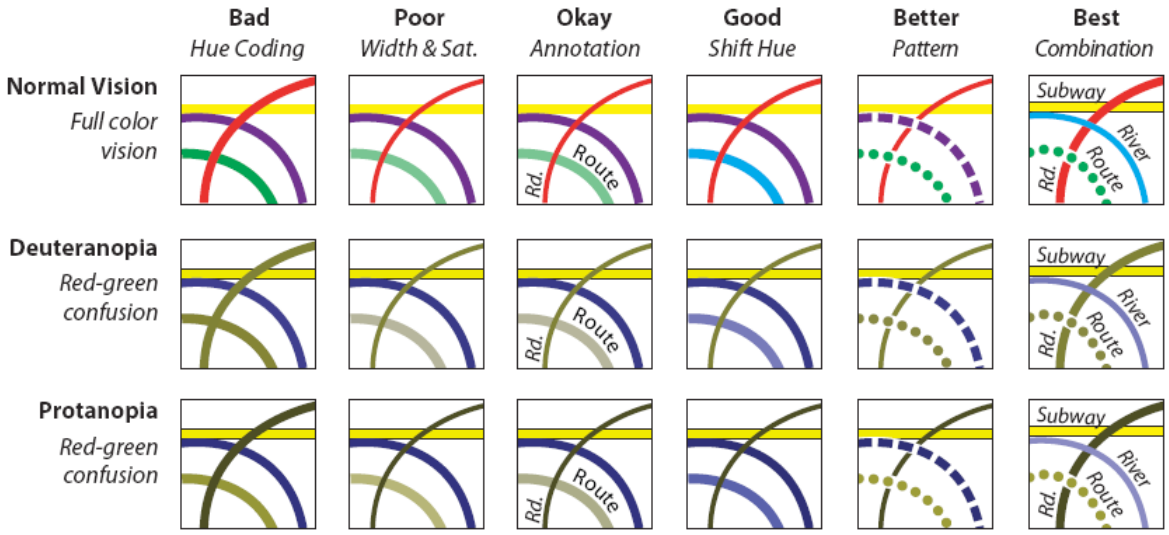


Abbildung 5-24: Darstellungsmöglichkeiten für linienhafte Objekte bei Farbfehlsichtigkeit (Quelle: Jenny & Kelso 2007)

5.7 Zusammenfassung

In diesem Kapitel wurden zunächst die Grundzüge der Farbtheorie und die Beschreibung von Farben in Bezugssystemen (Farbräumen) zusammengefasst. Mit dem CIELUV wurde der Farbraum beschrieben, der für die Modellierung des Optimierungsproblems im nächsten Kapitel genutzt wird. Diese allgemeinen Beschreibungen wurden anschließend auf die Anwen-

dung von Farbe unter bestimmten technischen Restriktionen und bei Farbsehschwächen eines Nutzers vertieft und Möglichkeiten einer Berücksichtigung dieser Einflüsse in Abhängigkeit von einem bestimmten Nutzer und seinem Anzeigegerät aufgezeigt.

6 Optimale Platzierung von Punkten im Farbraum

Im letzten Kapitel wurden durch die Ausführungen zur Farbwahrnehmung und zu verschiedenen Farbräumen die Grundlagen für die numerische Beschreibung und Reproduktion von Farben gelegt. Mit CIELUV und CIELAB wurden zwei Farbräume vorgestellt, auf denen eine Metrik definiert ist, die der visuellen Wahrnehmung des Menschen entspricht. Auf dieser Basis lassen sich die für das Mapping on demand benötigten gut unterscheidbaren Farben bestimmen (Dieses Problem wird im Folgenden kurz als „Farbproblem“ bezeichnet.). Die Forderung der guten Unterscheidbarkeit von Farben führt darin zu dem äquivalenten aber mathematisch formulier- und modellierbaren Problem der Identifikation von Farben, die einen möglichst großen Abstand besitzen. Es handelt sich dann dabei um ein Distanzproblem, das – auf Basis der Euklidischen Distanz – die Platzierung von Objekten im Raum zum Ziel hat. Probleme dieser Art werden im Kontext der *Algorithmischen Geometrie* und der *Mathematischen Optimierung*[46] behandelt, insbesondere letztere bietet auch einen geeigneten Formalismus zur präzisen Beschreibung und Modellierung dieser Probleme.

Es wird sich zeigen, dass die Lösung von Distanzproblemen durch effiziente[47] Standardverfahren nicht das gewünschte Ergebnis – eine eindeutige und global optimale Lösung – ergibt, sondern aufgrund der Nicht-Konvexität des Problems die global optimale Lösung nicht von der Vielzahl lokal optimaler Lösungen, die u.a. durch Effekte der Symmetrie und Kombinatorik zustande kommen, zu differenzieren ist. Eine gesicherte global optimale Lösung lässt sich nur durch NP-vollständige Algorithmen bestimmen.

In diesem Kapitel werden die für die Formulierung und Lösung von Distanzproblemen, speziell des Farbproblems, benötigten methodischen Grundlagen zusammengestellt. Auf deren Basis lässt sich ein Verfahren entwickeln, das für den Fall des Farbproblems eine effiziente Lösung ermöglicht. Die Beschreibung dieses Verfahrens ist Teil des nächsten Kapitels.

Im Abschnitt 6.1 erfolgt zunächst eine sprachliche Formulierung, anschließend die mathematische Beschreibung des Farbproblems. Abschnitt 6.2 führt mit dem Voronoi-Diagramm und den Formalismen der Mathematischen Optimierung in die genannten Ansätze zur Lösung von Distanzproblemen ein. Abschnitt 6.3 diskutiert deren Komplexität anhand des Farbproblems und einiger verwandter Probleme. In den Abschnitten 6.4 bis 6.6 werden grundlegende Lösungsverfahren vorgestellt, die im weiteren Verlauf benötigt werden: Abschnitt 6.4 beschreibt ein geometrisches Verfahren auf Basis des Voronoi-Diagramms, Abschnitt 6.5 Verfahren der Mathematischen Optimierung, die die Bestimmung einer lokal optimalen Lösung zum Ziel

[46] Besonders im englischsprachigen Raum wird auch häufig von Mathematischer Programmierung gesprochen. Der Begriff Programmierung bezeichnet dabei nicht das Erstellen eines Computerprogramms, sondern wird nach seiner Herkunft aus dem Bereich der Planung („Programm“) auch in diesem Sinne genutzt.

[47] Als effizient wird in dieser Arbeit ein Verfahren bezeichnet, das ein Problem von der Größenordnung des Farbproblems (zur Größe des Farbproblems vgl. Abschnitt 6.1) on demand, d.h. in der Zeit einer typischen Datenabfrage im WWW, löst.

haben. Dort ist besonders das SQP-Verfahren von Bedeutung, das im Kapitel 7 zur Lösung des Farbproblems beiträgt. Abschnitt 6.6 beschreibt zwei Verfahren der Mathematischen Optimierung, die die Bestimmung einer global optimalen Lösung zum Ziel haben. Abschnitt 6.7 skizziert Arbeiten, die sich ebenfalls mit der Bestimmung gut unterscheidbarer Farben beschäftigen. Eine abschließende Zusammenfassung dieses Kapitels gibt Abschnitt 6.8.

6.1 Problembeschreibung und -modellierung

Ziel der Lösung des Farbproblems ist es, raumbezogene Objekte vor einem Kartenhintergrund farblich so darzustellen, dass die verwendeten Farben sowohl eine gute visuelle Differenzierbarkeit der Objekte untereinander als auch gegenüber dem Kartenhintergrund ermöglichen. Der Kartenhintergrund ist dabei in seiner Darstellung festgelegt (z.B. als Teil eines Kartenwerks) und die darin enthaltenen Farben sind als gegeben und nicht änderbar anzusehen. Abbildung 6-1 zeigt beispielhaft die Darstellung zweier linienhafter Objekte vor dem Hintergrund einer topographischen Karte.

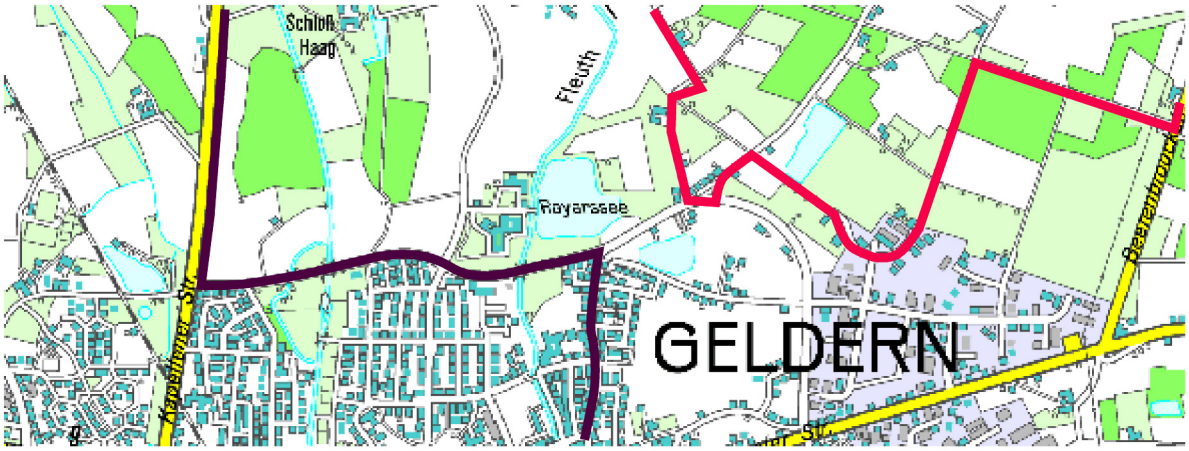


Abbildung 6-1: Platzierung linienhafter Objekte vor dem Hintergrund einer topographischen Karte (Quelle der topographischen Karte: Regionalverband Ruhr)

Die Aufgabe der Bestimmung gut unterscheidbarer Farben wird nun als Distanzproblem formuliert. Dafür wird von folgender Situation ausgegangen:

GEGEBEN — *Gegeben ist eine Menge F von m Farben mit ihren Farborten. Jeder Farbort ist durch Farbwerte in einem gebräuchlichen dreidimensionalen Farbraum (Anwendungsfarbraum) festgelegt und nicht änderbar.*

Die Formulierung des eigentlichen Farbproblems erfolgt als sogenanntes MAXMIN-Problem:

MAXMIN — *Gesucht ist eine Menge X von n Farben. Der minimale Abstand jedes Farbortes aus X soll sowohl zu jedem anderen Farbort aus X als auch zu jedem Farbort aus F in einem gleichabständigen Farbraum maximal werden.*

Für spezifische Anforderungen (vgl. Abschnitt 5.5 zur Farbe in der Visualisierung und Abschnitt 5.6 zur Personalisierung der Farbdarstellung) sind weitere Bedingungen erforderlich:

- *Die Farben sollen einen gewissen Grad an Buntheit erreichen, d.h. insbesondere nicht auf der Unbuntgeraden liegen.*
- *Die Farben sollen entlang einer Dimension des Farbraums variieren.*
- *Bestimmte Farben sind aufgrund von Restriktionen des Ausgabemediums oder Sehschwächen für einen bestimmten Anwender nicht darstellbar.*

Als Alternative zum Problem MAXMIN wurden in Arbeiten, die vergleichbare Distanzprobleme betrachten (vgl. Abschnitt 6.3.2), auch MAXSUM-Probleme formuliert:

MAXSUM — *Gesucht ist eine Menge X von n Farben. Die Summe der Abstände aller Farborte aus X untereinander und zwischen allen Farborten aus X und F soll dabei maximal werden. Weiterhin soll jeder Farbort aus X einen bestimmten Mindestabstand*

- *zu jedem anderen Farbort aus X nicht unterschreiten,*
- *zu den Farborten aus F nicht unterschreiten.*

Das Farbproblem soll nun als Problem der Mathematischen Optimierung formuliert werden. Dafür wird von folgender allgemeiner Form ausgegangen (Boyd & Vandenberghe 2004):

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \le 0, \quad i = 1,\ldots,m \\ &&& h_i(x) = 0, \quad i = 1,\ldots,p\,. \end{aligned}$$

Die Lösung des Problems minimiert den Funktionswert $f_0(x): \Re^n \to \Re$ unter allen $x = (x_1,\ldots,x_n)$, die den Nebenbedingungen $f_i(x) \le 0$, $i = 1,\ldots,m$, und $h_i(x) = 0$, $i = 1,\ldots,p$, genügen.

Eine Modellierung in dieser Form erfordert zunächst drei Vereinbarungen:

- Die Festlegung eines Anwendungsfarbraums, der den Umfang der zur Darstellung verfügbaren Farben angibt,
- die Definition eines Abstandsmaßes und
- die Modellierung des Anwendungsfarbraums in einem auf dem Abstandsmaß basierenden gleichabständigen Farbraum (Optimierungsfarbraum).

Als Anwendungsfarbraum ist nach den bisherigen Ausführungen in dieser Arbeit der sRGB-Farbraum prädestiniert, da er direkt die Farbdarstellung auf Bildschirmen (vgl. Abschnitt 5.3.2) repräsentiert.

Als Funktion zur Berechnung von Farbabständen wurde im Abschnitt 5.3.4.3 die Euklidische Distanz angegeben, die auch für das Farbproblem genutzt wird (im Folgenden wird für diese

Distanz die Notation $\|\cdot\|_2$ gebraucht; weitere Ausführungen zum Abstandsbegriff finden sich im Anhang A.4).

Als geeigneter Optimierungsfarbraum wurde bereits der CIELUV-Farbraum, der besonders bei Selbstleuchtern als gleichabständiger Farbraum genutzt wird (vgl. Abschnitt 5.3.4.1), genannt. Die Modellierung des Anwendungsfarbraums im Optimierungsfarbraum verdeutlicht Abbildung 6-2. Die Graphik zeigt links den dreidimensionalen Körper, der durch die Transformation des sRGB-Würfels (vgl. Abbildung 5-10) in den CIELUV-Farbraum entsteht. Dieser Körper gibt den Umfang der verfügbaren Farben an. Für die Formulierung als Optimierungsproblem ist die Beschreibung dieses Körpers durch Nebenbedingungen erforderlich. Abbildung 6-2 zeigt rechts anhand einer Triangulation der konvexen Hülle des Körpers, dass dies näherungsweise durch ein konvexes Polyeder erfolgen kann. Dieses Polyeder ist als Schnitt von Halbräumen der Form

$$\{x \mid a^T x \le b\}$$

mit $a \in \Re^n$, $a \neq 0$ und $b \in \Re$ modellierbar.

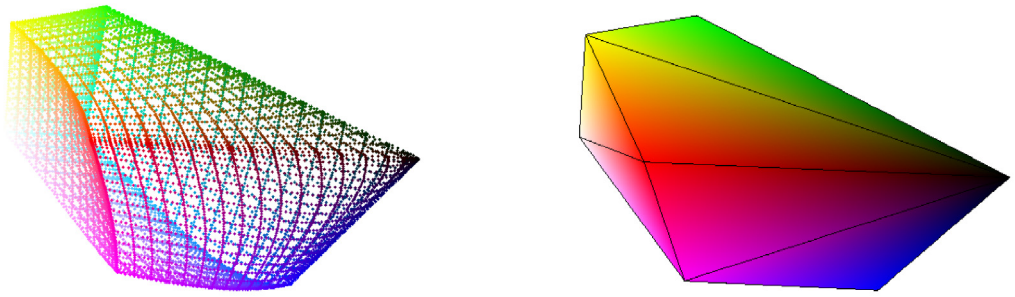


Abbildung 6-2: Transformation des sRGB-Würfels in den CIELUV-Farbraum: Darstellung des Farbkörpers (links) und Approximation durch eine Triangulation der konvexen Hülle (rechts)

Damit ist das Problem MAXMIN formulierbar als:

$$\text{MAXMIN} \qquad \text{maximize} \qquad \min\left(\left\|X_i, Y_j\right\|_2\right), \quad i = 1,\ldots,n,\ j = 1,\ldots,m+n,\ j > i$$

$$\text{subject to} \qquad a_k^T X_i \le b_k, \qquad i = 1,\ldots,n,\ k = 1,\ldots,9\,.$$

Darin bezeichnet $Y = (X, F)$ die Gesamtmenge der gesuchten und der vorab gegebenen Farben. Jede Farbe Y_j wird durch ihre Koordinaten L_j, u_j, v_j im CIELUV-Farbraum repräsentiert. Das Farbraum-Polyeder ist als Schnitt von 9 Halbräumen modellierbar.

Das MAXSUM-Problem stellt sich folgendermaßen dar:

$$\text{MAXSUM} \qquad \text{maximize} \qquad \sum_{i=1}^{n} \sum_{j=1,\ j>i}^{m+n} \left\|X_i, Y_j\right\|_2$$

$$\text{subject to} \qquad a_k^T X_i \le b_k, \qquad i = 1,\ldots,n,\ k = 1,\ldots,9$$

$$d_{\min} \le \left\|X_i, Y_j\right\|_2, \qquad i = 1,\ldots,n,\ j = 1,\ldots,m+n,\ j > i\,.$$

Dieses Problem ist im Folgenden nicht von Bedeutung, da die Forderung einer guten Unterscheidbarkeit einen möglichst großen Abstand jeder Farbe zu jeder anderen Farbe nötig macht. Die Gesamtsumme der Abstände ist demnach ohne Belang.

Die Anzahl der Farben der Menge $Y = (F, X)$ ist offensichtlich mit der Größe des Farbumfangs des sRGB-Farbraums nach oben begrenzt, d.h. es gibt für die Anzahl der Farben der Menge Y bei einer bestimmten Mindestdistanz, die für eine gute Unterscheidbarkeit immer eingehalten werden sollte (z.B. 45 Längeneinheiten, vgl. Abschnitt 5.5.1), eine obere Schranke M mit $m + n \leq M$. Damit existieren zwei Extremfälle der Farbbestimmung:

- Die Menge F der vorgegebenen Farben ist die leere Menge ($F = \varnothing$). Damit sind $n \leq M$ zu bestimmende Farben frei platzierbar, zu berücksichtigen sind lediglich die Abstände zwischen diesen Farben.
- Mit wachsender Anzahl gegebener Farben der Menge F sinkt die mögliche Anzahl der neu zu bestimmenden. Für $m = M$ macht eine Berechnung weiterer Farben keinen Sinn ($X = \varnothing$).

Der Wert der oberen Schranke M hängt bei nichtleerer Menge F von der Verteilung der gegebenen Farben im Farbraum ab. Für $F = \varnothing$ gilt bei einem Mindestabstand von 45 Längeneinheiten $m = M = 25$.

Im Verlauf dieses und des nächsten Kapitels wird sich zeigen, dass das Farbproblem besonders durch die verwendete Distanzfunktion und die Modellierung des Optimierungsfarbraums als konvexes Polyeder charakterisiert ist. Während die Distanzfunktion, insbesondere die Nicht-Linearität der Euklidischen Distanz, die Komplexität und damit die Lösbarkeit des Problems bestimmt, bietet die Konvexität des Farbraums eine Möglichkeit, diese Komplexität zu verringern.

6.2 Ansätze zur Lösung von Distanzproblemen

In diesem Abschnitt werden zur Lösung des Farbproblems zwei Herangehensweisen aus verschiedenen wissenschaftlichen Kontexten eingeführt. Im Bereich der *Algorithmischen Geometrie* (Computational Geometry) wird die Charakterisierung von Nachbarschaftsbeziehungen mit Hilfe der geometrischen Struktur des Voronoi-Diagramms genutzt, die *Mathematische Optimierung* sucht eine numerische Lösung des im vorangegangenen Abschnitt formulierten Optimierungsproblems. Für die Vertiefung dieser Ansätze sind jedoch zunächst noch einige mathematische Grundlagen von Bedeutung.

Die verwendeten Symbole und mathematischen Bezeichnungen folgen der im Kontext der Mathematischen Optimierung verbreiteten Notation, so wird dort insbesondere auf die Kennzeichnung von Vektoren und Matrizen verzichtet.

6.2.1 Mathematische Grundlagen

Im Rahmen der Problembeschreibung des Abschnitts 6.1 wurde mit der *Konvexität* ein wesentliches Charakteristikum des Farbproblems genannt. Für die weiteren Ausführungen wird

nun dieser Begriff in seiner Bedeutung für die Beschreibung von Mengen und Funktionen präzisiert.

6.2.1.1 Konvexe Mengen und konvexe Hülle

Die Inhalte dieses Abschnitts sind entnommen aus Boyd & Vandenberghe (2004), dort finden sich auch vertiefende Ausführungen.

Konvexe Menge und konvexe Kombination

Eine Menge C ist *konvex*, wenn die Verbindungsstrecke zweier beliebiger Punkte aus C ganz in C liegt. Für alle $x_1, x_2 \in C$ und jedes t mit $0 \le t \le 1$ muss damit gelten

$$tx_1 + (1-t)x_2 \in C .$$

Abbildung 6-3 zeigt die geometrische Interpretation einer konvexen und einer nicht-konvexen Menge im $\Re^2$.

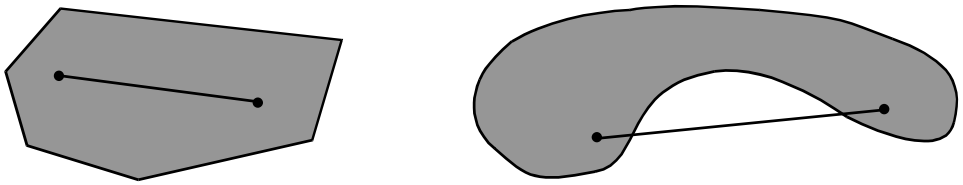


Abbildung 6-3: Konvexe (links) und nicht-konvexe (rechts) Menge im $\Re^2$

Eine *konvexe Kombination* von Punkten $x_1, \ldots, x_k \in C$ ist ein Punkt der Form $t_1 x_1 + \ldots + t_k x_k$, wobei $t_1 + \ldots + t_k = 1$ und $t_i \ge 0$ für $i = 1, \ldots, k$.

Konvexe Hülle

Die *konvexe Hülle* einer Menge C ($conv\, C$) ist die Menge aller konvexen Kombinationen von Punkten in C:

$$conv\, C = \{ t_1 x_1 + \ldots + t_k x_k \mid x_i \in C,\ t_i \ge 0,\ i = 1, \ldots, k,\ t_1 + \ldots + t_k = 1 \} .$$

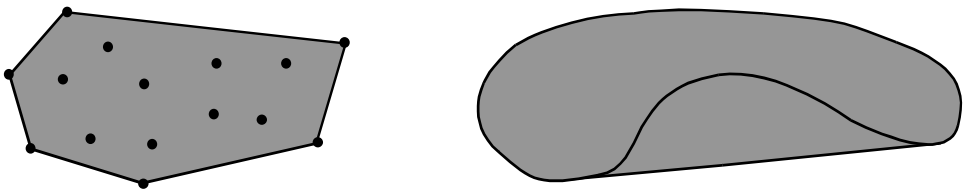


Abbildung 6-4: Konvexe Hülle einer konvexen (links) und nicht-konvexen (rechts) Menge im $\Re^2$

Abbildung 6-4 zeigt die konvexen Hüllen einer konvexen und einer nicht-konvexen Menge im $\Re^2$.

6.2.1.2 Konvexe Funktion

Eine Funktion $f : \Re^n \to \Re$ ist *konvex*, wenn der Definitionsbereich von f ($dom\ f$) eine konvexe Menge ist und für alle $x, y \in dom\ f$ und t mit $0 \le t \le 1$ gilt:

$$f(tx + (1-t)y) \le\ tf(x) + (1-t) f(y) .$$

Eine Funktion $f : \Re^n \to \Re$ ist *strikt konvex*, wenn darüber hinaus für $x \neq y$ und $0 < t < 1$ gilt:

$$f(tx + (1-t)y) < tf(x) + (1-t)f(y).$$

Eine Funktion f ist *konkav*, falls $-f$ konvex ist, und *strikt konkav*, wenn $-f$ strikt konvex ist.

Abbildung 6-5 zeigt die geometrische Interpretation einer konvexen bzw. konkaven Funktion: Alle Linien-Segmente zwischen zwei Punkten $(x, f(x))$ und $(y, f(y))$ liegen „oberhalb" bzw. „unterhalb" des Graphen von f.

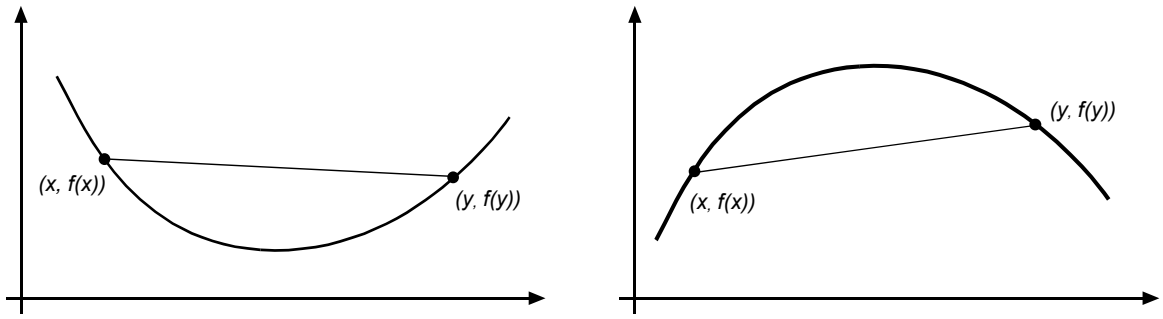


Abbildung 6-5: Konvexe (links) und konkave (rechts) Funktion

Konvexitätsbedingungen

Eine Funktion f ist konvex, falls die Konvexitätsbedingung erster oder zweiter Ordnung erfüllt ist.

Konvexitätsbedingung erster Ordnung

Die Funktion f sei auf *dom f* differenzierbar und *dom f* sei offen. f ist dann und nur dann konvex, wenn *dom f* konvex ist und

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

für alle $x, y \in dom\ f$ gilt.

Darin bezeichnet $\nabla f(x)$ den Gradienten der Funktion f. Der Gradient ist ein Spaltenvektor, dessen Komponenten die partiellen Ableitungen $\partial f(x) / \partial x_i, i = 1, \ldots, n,$ sind:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)^T.$$

Konvexitätsbedingung zweiter Ordnung

Die Funktion f sei auf *dom f* zweimal differenzierbar und *dom f* sei offen. f ist dann und nur dann konvex, falls *dom f* konvex und die Hesse-Matrix positiv semidefinit ist:

$$\nabla^2 f(x) \succeq 0$$

für alle $x \in dom\ f$.

Die Hesse-Matrix enthält die zweiten partiellen Ableitungen $\frac{\partial^2 f(x)}{\partial x_i \partial x_j},\ i,j=1,\ldots,n,$ der Funktion $f:\ \nabla^2 f(x)=\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & & \vdots \\ \vdots & & & \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$.

Die Bestimmung der Konvexität wird im weiteren Verlauf zur Charakterisierung der Funktionen des Farbproblems notwendig.

Nach Einführung dieser allgemeinen mathematischen Grundlagen werden im Folgenden die beiden Lösungsansätze für das Farbproblem, Voronoi-Diagramme und Mathematische Optimierung, beschrieben.

6.2.2 Voronoi-Diagramme

Das Voronoi-Diagramm bezeichnet eine geometrische Struktur zur Charakterisierung der Nachbarschaft von Elementen im Raum. Die Beschreibung erfolgt meist für die Euklidische Ebene (z.B. Preparata & Shamos 1985, Ottmann & Widmayer 1996), ist aber auf n-dimensionale Räume erweiterbar (vgl. Okabe et al. 2000). Für diese Arbeit sind Voronoi-Diagramme im dreidimensionalen Raum von Bedeutung.

6.2.2.1 Definition und Anwendung

Ein Voronoi-Diagramm im $\Re^n$ ist wie folgt definiert (Okabe et al. 2000):

Sei $P=\{p_1,\ldots,p_m\}$ mit $2 \le m < \infty$ eine Menge von Punkten im $\Re^n$. Die Punkte aus P sind gegeben durch ihre Ortsvektoren $x_1,\ldots,x_m$ mit den kartesischen Koordinaten $(x_{11},\ldots,x_{1n}),\ldots,(x_{m1},\ldots,x_{mn})$. Es gelte $x_i \neq x_j$ für $i \neq j$ mit $i,j=1,\ldots,m$. Die Region

$$V(p_i)=\{x \mid \|x-x_i\|_2 \le \|x-x_j\|_2 \ \textit{für}\ j \neq i,\ j=1,\ldots,m\}$$
$$= \bigcap_{j=1,\ j\neq i}^{m} H(p_i,p_j)$$

wird als das n-dimensionale Voronoi-Polyeder des Punktes p_i, und die Menge $\tilde{V}(P)=\{V(p_1),\ldots,V(p_m)\}$ als das n-dimensionale Voronoi-Diagramm der Menge P bezeichnet. Der Halbraum $H(p_i,p_j)$ ist der geometrische Ort aller Punkte des $\Re^n$, die näher an p_i als an p_j liegen:

$$H(p_i,p_j)=\{x \mid \|x-x_i\|_2 \le \|x-x_j\|_2\} \quad j \neq i.$$

Das Voronoi-Polyeder des Punktes p_i ist also der Durchschnitt aller Halbräume, den p_i mit allen anderen Punkten aus P bildet.

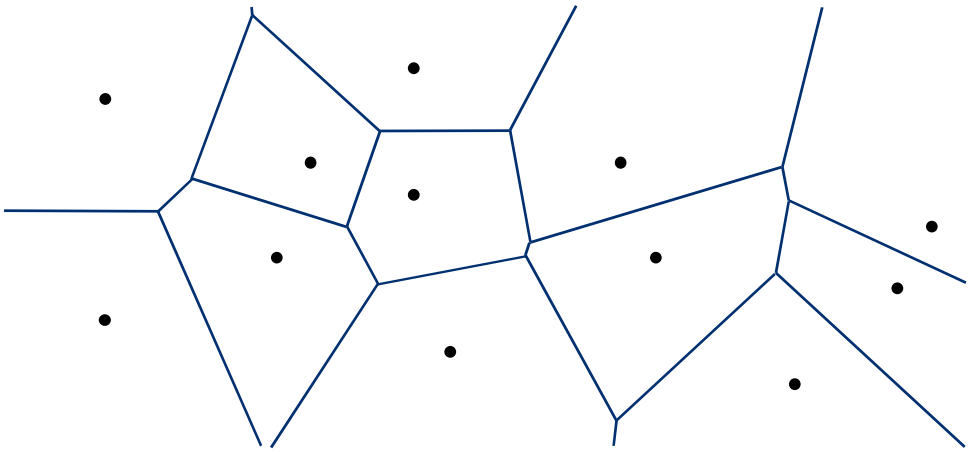


Abbildung 6-6: Voronoi-Diagramm in der Ebene

Abbildung 6-6 zeigt ein Voronoi-Diagramm in der Euklidischen Ebene. Bei den Voronoi-Polyedern handelt es sich dann offensichtlich um Polygone, die auch als *Voronoi-Regionen* bezeichnet werden (Ottmann & Widmayer 1996). In höherdimensionalen Räumen wird auch von V*oronoi-Zellen* gesprochen (Okabe et al. 2000). Aus Abbildung 6-6 geht auch hervor, dass ein Voronoi-Diagramm in einem unbeschränkten Raum aus beschränkten und unbeschränkten Regionen oder Zellen besteht.

Voronoi-Diagramme sind vielfältig einsetzbar. Eine Reihe von Möglichkeiten aus den Bereichen Informatik, Mathematik und Naturwissenschaften beschreibt beispielsweise Aurenhammer (1991). Die Anwendung von Voronoi-Diagrammen bei der Lösung von Optimierungsproblemen, die die Platzierung von Objekten in der Ebene zum Ziel haben, fassen Okabe & Suzuki (1997) zusammen. Für *m* Objekte, die in einer Region *S* bereits platziert sind, lassen sich u.a. unterscheiden (ebd.):

- Bestimmung eines Ortes, dessen Abstand zum nächsten Objekt maximal wird,
- Bestimmung eines Ortes, dessen Abstand zum entferntesten Objekt minimal wird,
- Optimierung der Lage der *m* Objekte durch Minimierung der maximalen Distanz zum nächsten Objekt,
- Optimierung der Lage der *m* Objekte durch Minimierung der durchschnittlichen Distanz zum nächsten Objekt.

Ein konkretes Vorgehen zur Lösung des erstgenannten Problems – das dem Farbproblem entspricht – wird im Abschnitt 6.4 beschrieben.

6.2.2.2 Berechnung von Voronoi-Diagrammen

Die Berechnung eines Voronoi-Diagramms kann über dessen dualen Graphen, die sogenannte Delaunay-Tessellation, erfolgen. Eine n-dimensionale Delaunay-Tessellation einer Menge P von Punkten im $\Re^n$ ist wie folgt definiert (Okabe et al. 2000):

Sei $\widetilde{V}(P)$ das Voronoi-Diagramm der Menge $P = \{p_1, \ldots, p_m\} \subset \Re^n$ mit $n+1 \leq m < \infty$; die Punkte aus P seien nicht kollinear. Weiterhin sei $Q = \{q_1, \ldots, q_{m_V}\}$ die Menge der Voronoi-

Knoten in $\widetilde{V}(P)$, $V(p_{i1}),\ldots,V(p_{ik_i})$ die Voronoi-(n-1)-Flächen, die zu q_i inzident sind, und T_i die n-dimensionale konvexe Hülle der Punkte $p_{i1},\ldots,p_{ik_i}$. Falls $k_i = n+1$ für alle $i = 1,\ldots,m_V$, besteht die Menge $\widetilde{D}(P) = \{T_1,\ldots,T_{m_v}\}$ aus *n*-dimensionalen Simplizes und wird als *n*-dimensionale Delaunay-Tessellation der konvexen Hülle von P bezeichnet.

Ein Simplex ist als Sonderfall einer konvexen Hülle definiert (Okabe et al. 2000): Im $\Re^n$ wird die konvexe Hülle einer Menge von $n+1$ Punkten, die nicht alle auf einer Hyperebene liegen, als Simplex bezeichnet. Für $n = 0$ ist der Simplex ein Punkt (0-Simplex), $n = 1$ ergibt ein Liniensegment, das durch zwei Punkte begrenzt wird (1-Simplex). Für $n = 2$ ist der Simplex ein Dreieck (2-Simplex), das durch 0- und 1-Simplizes begrenzt wird, für $n = 3$ ein Tetraeder.

Abbildung 6-7 zeigt eine Delaunay-Tessellation in der Ebene. Diese wird auch als Delaunay-Triangulation bezeichnet.

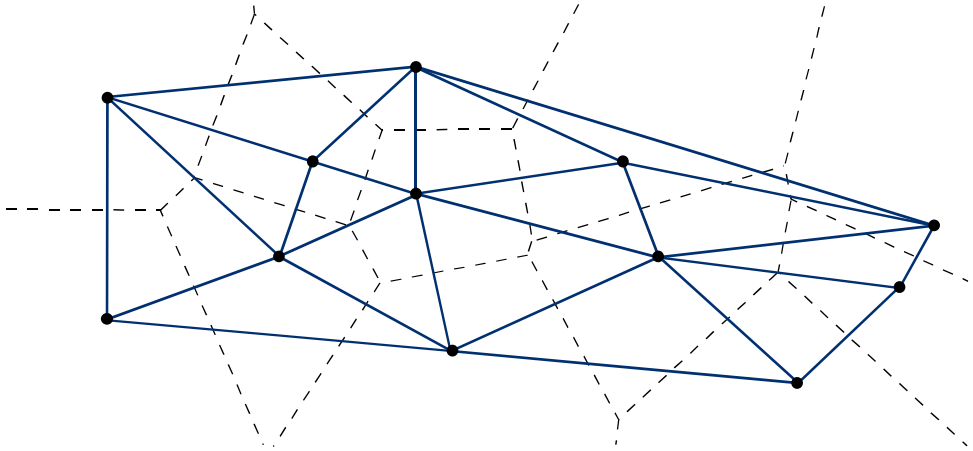


Abbildung 6-7: Delaunay-Triangulation und Voronoi-Diagramm (gestrichelt dargestellt) in der Ebene

Die Berechnung eines Voronoi-Diagramms bzw. einer Delaunay-Tessellation kann über den Zusammenhang einer Delaunay-Tessellation im *n*-dimensionalen Raum mit einer konvexen Hülle im *(n+1)*-dimensionalen Raum erfolgen (Okabe et al. 2000). Dafür sei eine Menge P von Punkten $\{p_1,\ldots,p_m\} \subset \Re^n$ gegeben; p_i habe die kartesischen Koordinaten $(x_{i1}, x_{i2},\ldots,x_{in})$. In einem ersten Schritt werden die Punkte aus P auf ein Paraboloid im *(n+1)*-dimensionalen Raum projiziert:

$$\hat{p}_i : (x_{i1}, x_{i2},\ldots,x_{in}) \mapsto (x_{i1}, x_{i2},\ldots,x_{in}, x_{i1}^2 + x_{i2}^2 + \ldots + x_{in}^2).$$

Im *(n+1)*-dimensionalen Raum wird anschließend die konvexe Hülle der Punkte berechnet. Von Bedeutung sind dann die „unteren Flächen" dieser Hülle („untere konvexe Hülle"). Eine untere Fläche zeichnet sich dadurch aus, dass unter der Annahme, dass eine Hyperebene durch diese Fläche gelegt wird, alle anderen Punkte $\hat{p}_i$, die nicht zur betrachteten Fläche inzident sind, oberhalb der Hyperebene liegen.

Die Projektion der unteren konvexen Hülle zurück in den Raum der Dimension *n* ist dann die Delaunay-Tessellation der Punkte in P. Aus der Delaunay-Tessellation lässt sich problemlos

der duale Graph, das Voronoi-Diagramm, berechnen. Abbildung 6-8 verdeutlicht den Vorgang der Projektion in den *(n+1)*-dimensionalen Raum und zurück.

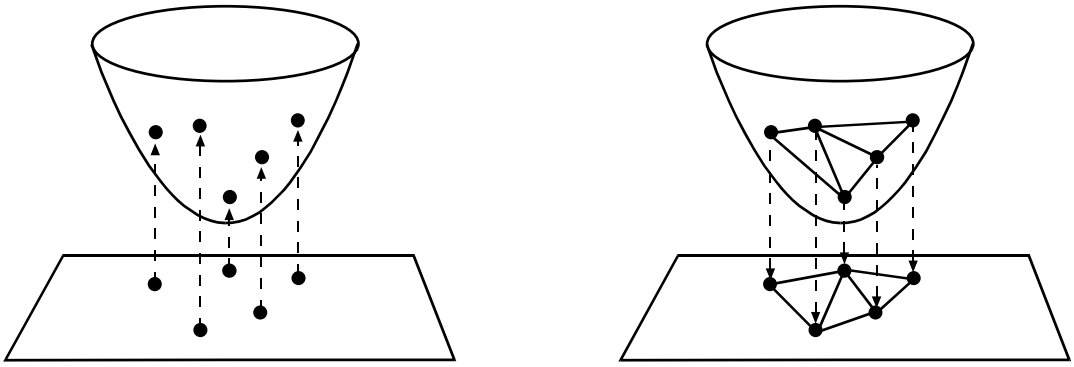


Abbildung 6-8: Konstruktion eines Voronoi-Diagramms in der Ebene; Projektion von Punkten auf ein Paraboloid im dreidimensionalen Raum(links); Projektion der unteren konvexen Hülle zurück in die Ebene (rechts)

Die Berechnung eines Voronoi-Diagramms lässt sich damit im Wesentlichen auf die Berechnung einer konvexen Hülle zurückführen. Die Implementierung eines geeigneten Algorithmus beschreiben beispielsweise mit „Quickhull“ Barber et al. (1996).

6.2.3 Mathematische Optimierung

Ziel einer Optimierung im mathematischen Kontext ist es, die Variablen eines Problems so zu wählen, dass der Wert einer Funktion „optimal“, d.h. minimal oder maximal wird. Der Lösungsraum der Variablen kann dabei durch Nebenbedingungen eingeschränkt sein.

Optimierungsprobleme sind in vielen Bereichen von Wissenschaft und Wirtschaft von Bedeutung. Typische Anwendungen sind beispielsweise die optimale Auslastung begrenzter Produktionskapazitäten oder die Erstellung von Einsatzplänen in Unternehmen oder Behörden. Eine Vielzahl von Anwendungsmöglichkeiten findet sich z.B. in Williams (1999).

In diesem Abschnitt wird zunächst eine allgemeine Beschreibung von Optimierungsproblemen gegeben und grundlegendes Vokabular eingeführt. Konkrete Lösungsverfahren folgen in den Abschnitten 6.5 und 6.6.

6.2.3.1 Terminologie von Optimierungsproblemen

Die folgenden Ausführungen sind, sofern nicht anders angegeben, entnommen aus Boyd & Vandenberghe (2004). Die deutschsprachige Terminologie findet sich beispielsweise in Alt (2002).

Im Abschnitt 6.1 wurde bereits die allgemeine Form eines Optimierungsproblems angegeben:

$$
\begin{array}{lll}
\text{SFP} & \text{minimize} & f_0(x) \\
 & \text{subject to} & f_i(x) \le 0, \quad i = 1,\ldots,m \\
 & & h_i(x) = 0, \quad i = 1,\ldots,p\,.
\end{array}
$$

Gesucht ist ein $x = (x_1, \ldots, x_n)$, das $f_0(x)$ unter allen x, die den Bedingungen $f_i(x) \le 0$, $i = 1, \ldots, m,$ und $h_i(x) = 0$, $i = 1, \ldots, p,$ genügen, minimiert. $x \in \Re^n$ wird als *Entscheidungsvariable* oder *Optimierungsvariable* bezeichnet, $f_0 : \Re^n \to \Re$ als *Ziel-* oder *Kostenfunktion*. Die Ungleichungen $f_i(x) \le 0$ bzw. die Gleichungen $h_i(x) = 0$ sind die *Nebenbedingungen* oder *Restriktionen*. Ein Problem mit Nebenbedingungen wird auch als *restringiertes* Optimierungsproblem bezeichnet, sind keine Nebenbedingungen vorhanden, handelt es sich um ein *un-* oder *nichtrestringiertes* Problem.

Die Menge der Punkte, für die Zielfunktion und Nebenbedingungen definiert sind, wird als *Domain* D des Problems bezeichnet:

$$D = \bigcap_{i=0}^{m} dom\ f_i \cap \bigcap_{i=1}^{p} dom\ h_i\ .$$

Ein Punkt $x \in D$, der den Nebenbedingungen genügt, ist ein *zulässiger Punkt*. Ein Problem, für das mindestens ein zulässiger Punkt existiert, wird ebenfalls *zulässig* genannt, andernfalls *unzulässig*. Die *zulässige Menge* Z umfasst alle zulässigen Punkte. Das *Optimum* des Problems SFP ist definiert als

$$\overline{p} = \inf\{ f_0(x) \mid f_i(x) \le 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ i = 1, \ldots, p \}.$$

Für ein unzulässiges Problem gilt $\overline{p} = \infty$. Falls zulässige Punkte x_k mit $f_0(x_k) \to -\infty$ für $k \to \infty$ existieren, gilt $\overline{p} = -\infty$ und das Problem SFP ist nach unten nicht beschränkt.

Eine Ungleichung $f_i(x) \le 0$ ist *aktiv* oder *bindend*, wenn für einen zulässigen Punkt x gilt $f_i(x) = 0$, bei $f_i(x) < 0$ ist die Ungleichung *inaktiv*. Eine Nebenbedingung wird als *redundant* bezeichnet, falls ihr Entfernen die zulässige Menge nicht ändert.

Globale und lokale Optima

Ein zulässiger Punkt $\overline{x}$ ist ein *optimaler Punkt*, wenn $f_0(\overline{x}) = \overline{p}$. Für die Menge aller optimalen Punkte, die *optimale Menge*, gilt

$$X_{opt} = \{ x \mid f_i(x) \le 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ i = 1, \ldots, p,\ f_0(x) = \overline{p} \}.$$

Existiert ein optimaler Punkt, ist das Problem SFP *lösbar*.

Der obige Begriff des Optimums definiert *global* optimale Punkte, daneben sind *lokal* optimale Punkte von Bedeutung. Ein Punkt ist lokal optimal, falls für ein z und ein $r > 0$ gilt

$$f_0(x) = \inf\{ f_0(z) \mid f_i(z) \le 0,\ i = 1, \ldots, m,\ h_i(z) = 0,\ i = 1, \ldots, p, \| z - x \|_2 \le r \},$$

d.h. x minimiert f_0 in einer Umgebung von z. Die Umgebung wird durch die geschlossene Kugel repräsentiert: $B(x, r) =: \{ z \in \Re^n \mid \| z - x \|_2 \le r \}$.

Ein Optimum wird durch ein Minimum der Zielfunktion bestimmt. Anhand des unrestringierten Problems

$$\text{minimize} \qquad f_0(x)$$

lassen sich unterscheiden (nach Alt 2002):

Ein Punkt $\bar{x} \in Z$ heißt *lokales Minimum* von f_0 auf Z, falls ein $r > 0$ mit

$$f_0(x) \geq f_0(\bar{x}) \qquad \forall x \in Z \cap B(\bar{x}, r)$$

existiert. Ein Punkt $\bar{x} \in Z$ heißt *striktes lokales Minimum* von f_0 auf Z, falls ein $r > 0$ mit

$$f_0(x) > f_0(\bar{x}) \qquad \forall x \in Z \cap B(\bar{x}, r), x \neq \bar{x}$$

existiert. Ein Punkt $\bar{x} \in Z$ heißt *globales Minimum* von f_0 auf Z, falls

$$f_0(x) \geq f_0(\bar{x}) \qquad \forall x \in Z$$

gilt. Ein Punkt $\bar{x} \in Z$ heißt *striktes globales Minimum* von f_0 auf Z, falls

$$f_0(x) > f_0(\bar{x}) \qquad \forall x \in Z, x \neq \bar{x}$$

gilt. Abbildung 6-9 illustriert die verschiedenen Minima.

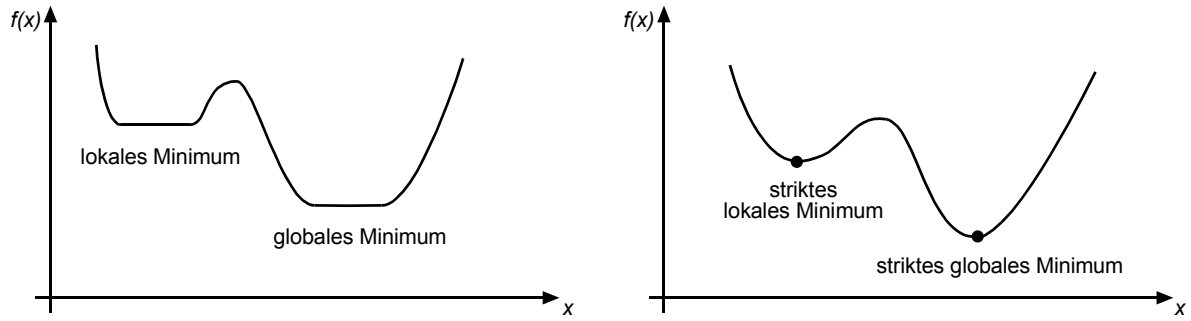


Abbildung 6-9: Arten von Minima einer Funktion

Offensichtlich ist für eine optimale und eindeutige Lösung eines Optimierungsproblems ein striktes globales Minimum wünschenswert.

Standardform

Die Formulierung des Problems SFP stellt ein Problem in Standardform (Standardform Problem) dar. Diese Form erfordert, dass die rechten Seiten der Ungleichungen und Gleichungen „Null“ sind. Die Darstellung der Ungleichungen muss als „kleiner gleich“ („$\leq$“) erfolgen. Die Standardform wird häufig von Optimierungsprogrammen, beispielsweise der MATLAB „Optimization Toolbox“, für die Eingabe von Optimierungsproblemen gefordert.

Maximierung

Weiterhin besteht häufig die Konvention, Optimierungsprobleme als Minimierung zu formulieren (z.B. in gängigen Optimierungsprogrammen wie MATLAB). Maximierungsprobleme der Form

$$\begin{aligned} &\text{maximize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0, && i = 1, \ldots, m \\ &&& h_i(x) = 0, && i = 1, \ldots, p \end{aligned}$$

sind dann durch eine Minimierung der Zielfunktion $-f_0(x)$ lösbar.

Im Folgenden wird der Begriff der „Maximierung“ lediglich genutzt, sofern es für den prägnanten Ausdruck eines Problems notwendig ist (z.B. für das Farbproblem in der Form MAXMIN). Für die Lösung eines Problems wird immer von einer Minimierung ausgegangen, insbesondere werden in diesem Zusammenhang die Begriffe „Minimierung“ und „Optimierung“ synonym verwendet.

6.2.3.2 Charakterisierung von Optimierungsproblemen

Die Form SFP stellt die allgemeine Form eines Optimierungsproblems dar. In Abhängigkeit von der Art der Entscheidungsvariablen, der Art der Zielfunktion und der Existenz und Art der Nebenbedingungen lassen sich verschiedene Typen von Problemen identifizieren. Für diese Typen existieren nicht nur spezifische, leistungsfähige Lösungsverfahren, sondern die Art des Problems und das gewählte Lösungsverfahren bestimmen auch maßgeblich die Effizienz und Güte – lokale oder globale Optimalität – einer Lösung. Die möglichst treffende Charakterisierung eines Problems ist damit Voraussetzung für die Wahl eines adäquaten Verfahrens und eine effiziente Berechnung.

Unrestringierte Probleme

Falls für ein Problem keine Nebenbedingungen vorliegen, handelt es sich um ein unrestringiertes Problem:

$$\text{UP} \qquad \text{minimize} \qquad f_0(x).$$

Es ist also lediglich die Zielfunktion zu minimieren. Die zur Lösung benötigten Optimalitätsbedingungen sind aus der Analysis bekannt und werden im Abschnitt 6.5.1.1 angegeben.

Lineare Probleme

Ein Problem, bei dem Zielfunktion und Nebenbedingungen affin sind, wird als lineares Problem oder Pogramm bezeichnet. Nach oben eingeführter Notation wird die allgemeine Form beschrieben durch

$$\begin{aligned} &\text{LP} && \text{minimize} && c^T x + d \\ &&& \text{subject to} && Gx \preceq h \\ &&&&& Ax = b \end{aligned}$$

mit $G \in \Re^{m \times n}$ und $A \in \Re^{p \times n}$ (Boyd & Vandenberghe 2004).

Konvexe Probleme

Das Problem SFP ist ein konvexes Optimierungsproblem, falls die Zielfunktion und die Ungleichungen der Nebenbedingungen konvex, die Gleichungen der Nebenbedingungen affin sind (Boyd & Vandenberghe 2004):

COP	minimize	$f_0(x)$
	subject to	$f_i(x) \leq 0, \quad i = 1,\ldots,m$
		$Ax = b$

mit $A \in \Re^{p \times n}$. Die Konvexität der Nebenbedingungen führt dazu, dass die zulässige Menge Z konvex ist. Ein konvexes Optimierungsproblem minimiert damit eine konvexe Zielfunktion auf einer konvexen Menge.

Die bereits vorgestellten linearen Programme lassen sich als Sonderfall konvexer Probleme auffassen, sind doch lineare Funktionen sowohl konvex als auch konkav.

Als weiterer Sonderfall konvexer Probleme werden *quadratische Programme* differenziert. Diese liegen vor, falls die Zielfunktion (konvex) quadratisch und die Nebenbedingungen affin sind (Boyd & Vandenberghe 2004):

QP	minimize	$\frac{1}{2} x^T P x + q^T x + r$
	subject to	$Gx \preceq h$
		$Ax = b$

mit einer symmetrischen positiv semidefiniten $n \times n$ Matrix P, $G \in \Re^{m \times n}$ und $A \in \Re^{p \times n}$. Ein quadratisches Programm minimiert eine konvexe quadratische Zielfunktion über eine zulässige Menge in Form eines Polyeders.

Sind außer der Zielfunktion auch die Ungleichungsnebenbedingungen konvex quadratisch, wird von einem *quadratisch restringierten quadratischen Programm* gesprochen (Boyd & Vandenberghe 2004):

QCQP	minimize	$\frac{1}{2} x^T P_0 x + q_0^T x + r_0$
	subject to	$\frac{1}{2} x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1,\ldots,m$
		$Ax = b$.

Die zulässige Menge ist hier für den Fall positiv definiter P_i ($P_i \succ 0$), $i = 1,\ldots,m,$ als Schnitt von Ellipsoiden interpretierbar (ebd.).

Die Besonderheit konvexer Probleme liegt in der Eigenschaft, dass jedes lokale Minimum von f_0 auf Z ein globales Minimum ist. Falls darüber hinaus f_0 strikt konvex ist, stellt eine Lösung $\bar{x}$ ein striktes globales Minimum dar und ist eindeutig bestimmt (vgl. Alt 2002).

Obwohl der Bereich der konvexen Optimierung seit über einem Jahrhundert bekannt ist (Boyd & Vandenberghe 2004), haben jüngere Entwicklungen seit den 1980er Jahren zu einer veränderten Wahrnehmung der gesamten Optimierung geführt. Diesen Umstand stellt beispielsweise Rockafellar (1993) heraus:

> *„In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."*

Zu einer ähnlichen Einschätzung kommt Fylstra (2005), der insbesondere den Einfluss der konvexen Optimierung auf die Probleme der Wirtschaftswissenschaften betont.

Die genannten Entwicklungen bestanden in der (Weiter)-Entwicklung von Innere-Punkte-Verfahren, die zunächst als Alternative zum etablierten Simplex-Verfahren für die Lösung linearer Programme genutzt wurden. Die Verallgemeinerung auf konvexe Programme ermöglichte auch für diese Probleme eine effiziente Lösbarkeit (vgl. Fylstra 2005, Boyd & Vandenberghe 2004). In diesem Zusammenhang stehen das Simplex- und Innere-Punkte-Verfahren auch für zwei wesentliche Paradigmen von Lösungsverfahren: Während der Simplex den Rand der zulässigen Menge nach der optimalen Lösung absucht, nähern sich Innere-Punkte-Verfahren der optimalen Lösung aus dem Inneren der zulässigen Menge heraus. Das Simplex-Verfahren wird in dieser Arbeit in Abschnitt 6.6.1 beschrieben, als Beispiel für ein Innere-Punkte-Verfahren wird das Barriere-Verfahren im Abschnitt 6.5.3.1 eingeführt.

Nichtlineare Probleme

Die allgemeine Form des Problems SFP wird als nichtlineares Optimierungsproblem bezeichnet, falls die Zielfunktion oder Nebenbedingungen nicht linear und nicht konvex sind bzw. ihre Konvexität unbekannt ist (Boyd & Vandenberghe 2004):

$$\begin{array}{lll} \text{NLP} & \text{minimize} & f_0(x) \\ & \text{subject to} & f_i(x) \le 0, \quad i = 1,\ldots,m \\ & & h_i(x) = 0, \quad i = 1,\ldots,p\,. \end{array}$$

Nichtlineare Probleme unterscheiden sich von den bisher vorgestellten dadurch, dass nicht ohne weiteres eine Aussage über die Güte einer gefundenen lokalen Lösung möglich ist: Aufgrund der Nicht-Konvexität der Funktionen bzw. des Problems sind lokale Minima nicht zwingend globale Minima.

Ganzzahlige Probleme

Enthält die Modellierung von Optimierungsproblemen eine Beschränkung von Variablen auf die Menge der ganzen Zahlen, wird allgemein von Ganzzahliger oder Diskreter Programmierung gesprochen. Eine weitere Differenzierung kann in *Rein Ganzzahlige* und *Gemischt Ganzzahlige* Programme erfolgen (Williams 1999). Erstere nutzen ausschließlich ganzzahlige, letztere daneben auch kontinuierliche Variablen. Eine häufige Anwendung sind beispielsweise ja-nein-Entscheidungen, die mit einer Beschränkung der Optimierungsvariablen auf $\{0,1\}$ modellierbar sind (ebd.).

Kombinatorische Probleme

Ein Sonderfall der Ganzzahligen Programme sind die Kombinatorischen Probleme, bei denen die zulässige Menge aus einer großen aber endlichen Menge zulässiger Lösungen besteht, die durch unterschiedliche Anordnung einer endlichen Anzahl von Objekten zustande kommt (Williams 1999). Eines der bekanntesten Probleme der Kombinatorischen Optimierung ist das

Travelling Salesman Problem (Cook et al. 1998): Ein Handelsreisender möchte *n* Orte in der Reihenfolge besuchen, die die dabei zurückgelegte Distanz minimiert.

6.3 Analyse des Farbproblems

Nach den bisherigen Ausführungen zur Mathematischen Optimierung und der Differenzierung verschiedener Arten von Problemen erfolgt nun eine Charakterisierung und Einordnung des Farbproblems. Eine weitergehende Analyse und die Betrachtung verwandter Fragestellungen wird die Komplexität des Problems im Hinblick auf die Lösbarkeit durch numerische Verfahren verdeutlichen.

6.3.1 Einordnung

Im Abschnitt 6.1 wurde das Farbproblem in der Form MAXMIN modelliert:

$$\begin{array}{lll} \text{MAXMIN} & \text{maximize} & \min\left(\left\|X_i, Y_j\right\|_2\right), \quad i=1,\ldots,n,\ j=1,\ldots,m+n,\ j>i \\ & \text{subject to} & a_k^T X_i \le b_k, \qquad i=1,\ldots,n,\ k=1,\ldots,9. \end{array}$$

Für die weiteren Betrachtungen wird zunächst eine gängige Umformung angewandt, die die nicht-glatte Minimumfunktion aus der Zielfunktion eliminiert und stattdessen neue Nebenbedingungen einführt:

$$\begin{array}{lll} \text{MAXMIN 1} & \text{maximize} & d_{\min} \\ & \text{subject to} & d_{\min} \le \left\|X_i, Y_j\right\|_2, \quad i=1,\ldots,n,\ j=1,\ldots,m+n,\ j>i \\ & & a_k^T X_i \le b_k, \qquad i=1,\ldots,n,\ k=1,\ldots,9. \end{array}$$

Nach den Ausführungen des letzten Abschnitts und der Betonung der Bedeutung konvexer Probleme ist zunächst zu klären, ob es sich bei MAXMIN 1 um ein solches Problem handelt.

Für die Zielfunktion und die Ungleichungen $a_k^T X_i \le b_k$, gilt offensichtlich Linearität und damit Konvexität. Anders verhält es sich mit den Ungleichungen $d_{\min} \le \left\|X_i, Y_j\right\|_2$. Aus der Definition für eine konvexe Funktion (Abschnitt 6.2.1.2) folgt zunächst mit der Dreiecksungleichung (Anhang A.4.1), dass die $L_2 - Norm$ – wie alle Normen auf dem $\Re^n$ (Boyd & Vandenberghe 2004) – konvex ist:

$$f(tx+(1-t)y) \le f(tx)+f((1-t)y) = tf(x)+(1-t)f(y)$$

für $0 \le t \le 1$. Weiterhin ist die offene Kugel

$$B(x,r) = \{z \in \Re^n \mid \| z-x \|_2 < r\}$$

auf dem $\Re^n$ eine konvexe Menge. Offensichtlich legen dann die Nebenbedingungen $d_{\min} \le \left\|X_i, Y_j\right\|_2$ als Komplement zur offenen Kugel eine nicht-konvexe zulässige Menge für

das Problem MAXMIN 1 fest. Damit ist das Problem nicht-konvex und als allgemeines nichtlineares Problem einzuordnen.

Nichtlineare Programme wurden im letzten Abschnitt – insbesondere in Abgrenzung zu konvexen Programmen – dadurch gekennzeichnet, dass lokale Minima nicht zwingend globale Minima sind. Aus diesem Grund wird zwischen *lokaler* und *globaler* Optimierung differenziert (Boyd & Vandenberghe 2004):

- Die lokale Optimierung verzichtet auf die Suche einer global optimalen Lösung und bestimmt eine lokal optimale Lösung, deren globale Optimalität zwar möglich, aber nicht sichergestellt ist. Ebenso wenig ist es möglich, eine Lösung sicher auf Globalität zu prüfen. Für die Berechnung lokaler Minima sind zwar sehr effiziente lokale Optimierungsverfahren verfügbar, allerdings gehen diese Verfahren von einem Startpunkt aus, dessen Wahl wesentlichen Einfluss auf die gefundene Lösung hat.
- Ziel der globalen Optimierung ist die Bestimmung einer Lösung, deren globale Optimalität sichergestellt ist. Das Finden einer solchen Lösung muss allerdings durch großen Rechenaufwand erkauft werden: Die worst-case-Komplexität von Verfahren der globalen Optimierung wächst in den meisten Fällen exponentiell mit der Zahl der Variablen und der Nebenbedingungen.

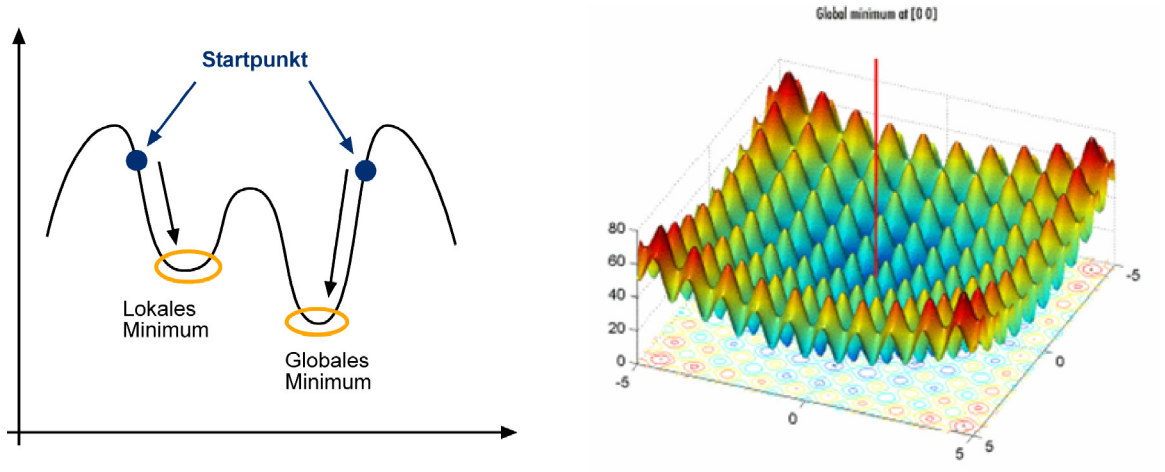


Abbildung 6-10: Auffinden eines Minimums in Abhängigkeit vom Startpunkt bei Anwendung lokaler Verfahren (links); rechts Rastrigins Funktion (Quelle: Matlab 2007)

Abbildung 6-10 verdeutlicht die Problematik der nichtlinearen Optimierung. Links ist die Situation bei Anwendung eines lokalen Verfahrens dargestellt: Das gefundene Minimum und damit die Optimalität der Lösung ist abhängig vom Startpunkt des Verfahrens. Der rechte Teil der Abbildung illustriert die mögliche Komplexität am Beispiel von Rastrigins Funktion

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right)$$

für $n = 2$:

$$f(x) = 20 + x_1^2 + x_2^2 - 10\left(\cos 2\pi x_1 + \cos 2\pi x_2\right)$$

mit $-5{,}12 \le x_1, x_2 \le 5{,}12$ (Weicker 2007).

Der Graph dieser Funktion hat die Form eines im Zentrum durchhängenden Eierkartons: Das globale Minimum befindet sich damit im Ursprung zwischen einer Vielzahl lokaler Minima. Aufgrund dieser Eigenschaft dient Rastrigins Funktion häufig als Benchmark zum Vergleich von Algorithmen (Weicker 2007, Salomon 1996).

Unabhängig von der lokalen oder globalen Optimalität ist das Farbproblem weiterhin durch die Kombinatorik und symmetrische Konstellationen möglicher Lösungen gekennzeichnet. Bevor diese Charakteristika vertiefend betrachtet werden, erfolgt im nächsten Abschnitt zunächst die Beschreibung einiger verwandter Probleme.

6.3.2 Verwandte Probleme

Als verwandte Probleme werden zwei sehr bekannte Beispiele, *Packprobleme* und *Probleme der Facility Location*, betrachtet. Neben ähnlichen bzw. identischen Problemformulierungen bieten die Packprobleme besonders eine für die weiteren Ausführungen geeignete Modellvorstellung für das Farbproblem.

6.3.2.1 Packprobleme

In den bisherigen Ausführungen wurde von Distanzen zwischen Punkten im dreidimensionalen Raum ausgegangen. Die Betrachtung von Packproblemen legt dagegen zunächst eine Modellvorstellung von Kugeln nahe: Der Optimierungsfarbraum mit dem Volumen V_{OR} wird durch Kugeln des Volumens $V_i, i = 1,\ldots,n,$ aufgefüllt. Der Radius r jeder Kugel entspricht dabei dem halben Abstand zweier Farborte, der Mittelpunkt jeder Kugel ist ein gesuchter Farbort.

Probleme dieser Art gehören der allgemeinen Klasse der *Knapsack-*, *Rucksack* oder *Packprobleme* an (Kellerer et al. 2004): Dem Wanderer, der seinen Rucksack packt, erscheinen eine Vielzahl von Dingen für seine Tour sinnvoll, sein Rucksack besitzt allerdings nur ein begrenztes Fassungsvermögen. Er muss sich demnach einerseits für die Dinge entscheiden, die ihm den größten Nutzen versprechen, andererseits die Kapazität seines Rucksacks möglichst gut ausnutzen. In der Terminologie der Mathematischen Optimierung handelt es sich also um ein Kombinatorisches Problem. Die allgemeine Beschreibung eines Knapsack-Problems im Formalismus der Mathematischen Optimierung lautet (Kellerer et al. 2004):

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \le c$$

$$x_i \in \{0,1\}, \qquad i = 1,\ldots,n\,.$$

Die Summe der Objekte x_i mit ihrem jeweiligen Nutzen p_i soll also maximal werden, dabei ist die Summe der Gewichte w_i durch die Kapazität c begrenzt. Die Entscheidungsvariablen

x_i sind in diesem Fall als boolsche Variablen modelliert und drücken die Entscheidung über „einpacken“ oder „nicht einpacken“ des Objekts x_i aus.

Dieses allgemeine Knapsack-Modell ist in dieser Form lediglich eindimensional, speziellere Anwendungen betrachten auch höherdimensionale Probleme. Ein bekanntes Beispiel sind *Cutting-Probleme*, deren Ziel es bspw. ist, aus einer rechtwinkligen Platte eines Materials rechtwinklige Stücke in vorab gegebener Größe so auszuschneiden, dass das Material möglichst optimal genutzt wird (Kellerer et al. 2004).

Ein weiteres Charakteristikum des Knapsack-Modells ist, dass die Größen der zu packenden Objekte von vorneherein feststehen müssen. Dies ist für das Farbproblem allerdings nicht der Fall, sondern der Radius der zu packenden Kugeln wird als variabel bzw. als zu maximierend angenommen. Probleme dieser Art werden im Kontext einer spezielleren Klasse kombinatorischer Probleme, dem „*Circle Packing*“ (für den ebenen Fall) betrachtet: Es sollen *n* Kreise so in einem Container (Einheitsquadrat, Rechteck, Dreieck oder Kreis) angeordnet werden, dass sich einerseits keine zwei Kreise überlappen, andererseits den Container aber möglichst gut ausfüllen (Castillo et al. 2008). Dabei besitzen alle Kreise entweder einheitliche oder verschiedene Radien. Im ersten Fall kann die Formulierung als Optimierungsproblem analog zur Formulierung MAXMIN 1 in dieser Arbeit erfolgen (vgl. Szabó et al. 2005, Castillo et al. 2008).

Diese auf den ersten Blick recht trivial erscheinende Anwendung des Circle Packings zeigt sehr gut, dass dies zwar für kleinere n ($\sim n \leq 5$) zutrifft und eine Lösung noch durch bloße Vorstellung möglich ist, die Komplexität und damit der Rechenaufwand für größere *n* aber sehr schnell ansteigt (vgl. Szabó et al. 2005).

Castillo et al. (2008) zeigen auch die Verbindung des Circle Packings zu Problemen der *Facility Location* und zum *p-dispersion-Problem*, die im nächsten Abschnitt vertieft werden. Dort werden auch mögliche bzw. in vergleichbaren Arbeiten genutzte Lösungsverfahren aufgezeigt.

6.3.2.2 Facility Location und p-dispersion-Problem

Ausführliche Betrachtungen zur Distanzoptimierung sind im Bereich der Platzierung von Objekten in der Realwelt, meist in Form der Standortplanung von (Produktions-)anlagen (engl. allgemein Facility Location), verfügbar; dabei sind sowohl Minimierungs- als auch Maximierungsmodelle von Bedeutung.

Unter dem allgemeinen Begriff der Facility Location werden hauptsächlich Probleme der Standortbestimmung, die eine Minimierung von Distanzen zum Ziel haben, subsumiert. Ein typisches Beispiel ist die Suche eines Standortes für ein Lager. Dieses Lager soll so platziert werden, dass Kunden an verschiedenen Orten möglichst kostengünstig beliefert werden können, d.h. dass z.B. die maximale Distanz vom Lager zu diesen Kunden minimal wird. Einen Überblick über verschiedene Probleme und Modelle dieser Art geben Eiselt & Laporte (1995).

Dem Problem dieser Arbeit am nächsten kommen Modelle der Standortplanung, die sich mit der Maximierung von Distanzen befassen. Diese Probleme werden unter dem spezielleren Begriff der *Obnoxious Facility Location* oder *Undesirable Facility Location* zusammengefasst. Ein typisches Beispiel für solch eine gefährliche oder unerwünschte Anlage ist ein Kernkraftwerk, das möglichst weit entfernt von bewohnten Gebieten platziert werden soll.

Im Kontext der Obnoxious Facility Location wurden unterschiedlichste Problemformulierungen und Lösungsverfahren betrachtet. Als Klassifizierungskriterien nennen Erkut & Neuman (1989) u.a.:

- *Anzahl* der zu platzierenden Anlagen,
- *Lösungsraum*: $\Re^n$ oder Netzwerk,
- Art der *zulässigen Menge*: Diskret oder kontinuierlich; bei kontinuierlichen Mengen innerhalb konvexer oder nicht-konvexer Polygone,
- *Distanzfunktion*: Euklidische Distanz, Manhattan Distanz oder Distanz entlang eines Netzwerks (vgl. Definition von Distanzfunktionen im Anhang A.4),
- *Abhängigkeit* zwischen vorhandenen und zu bestimmenden Facilities,
    - Distanzen zwischen neuen Facilties,
    - Distanzen zwischen neuen und vorhandenen Facilities,
    - Distanzen zwischen neuen und zwischen neuen und vorhandenen Facilities,
- Art der *Zielfunktion*: MAXMIN, MAXSUM.

Aufgrund dieser Vielzahl von Kriterien sind eine große Zahl ausdifferenzierter Probleme betrachtet worden. Einen Überblick geben beispielsweise Erkut & Neuman (1989) und Cappanera (1999). Dementsprechend vielfältig sind die genutzten Lösungsverfahren, ein effizientes Standardverfahren ist allerdings nicht verfügbar.

Die Arbeiten, die Erkut & Neuman (1989) zur Platzierung mehrerer Facilities mit der Formulierung MAXMIN zusammenfassen, betrachten überwiegend den speziellen Fall der p-dispersion-Probleme. Diese Probleme haben zum Ziel, aus einer Menge von n Kandidaten p Punkte so auszuwählen, dass die minimale Distanz der p Punkte maximiert wird (Erkut 1990). Als Verfahren werden beispielsweise Branch-and-Bound (Abschnitt 6.6.2) und Heuristiken eingesetzt. Jüngere Arbeiten (Erkut 1990, Pisinger 1999) nutzen zur Lösung von diskreten p-dispersion-Problemen ebenfalls das Branch-and-Bound-Verfahren. Die dabei betrachtete Menge von Kandidaten ist allerdings – gerade im Vergleich zum Farbproblem mit rein rechnerisch 16,7 Millionen verfügbaren Farben – als klein zu bezeichnen: Erkut nutzt maximal 40 Kandidaten, Pisinger 200. Anhand von Berechnungen für diese Probleme wird allerdings der rapide Anstieg der Lösungszeit mit wachsenden n und p deutlich.

Zur Bestimmung von Startpunkten für das Branch-and-Bound-Verfahren nutzt Pisinger (1999) eine Heuristik in Form eines Greedy-Algorithmus. Dabei wird zunächst die gesamte Menge der n Kandidaten ausgewählt und anschließend sukzessiv der jeweils am schlechtesten

platzierte Punkt entfernt. Dies erfolgt so lange, bis lediglich *p* Punkte übrig sind. In einem zweiten Schritt wird geprüft, ob das gegenseitige Austauschen eines der *p* Punkte mit einem der bisher nicht gewählten Punkte eine Verbesserung der Zielfunktion ergibt. Erkut (1990) beschreibt das gleiche Vorgehen mit dem Ziel, auf eine exakte aber teure Lösung des Problems durch Branch-and-Bound zu verzichten, und stattdessen effizient eine lediglich „gute" Lösung zu berechnen. Durch Beispielrechnungen wird gezeigt, dass eine solche Heuristik in fast 95% der Fälle sogar eine global optimale Lösung findet.

Drezner & Erkut (1995) verallgemeinern das diskrete p-dispersion-Problem auf kontinuierliche Werte, d.h. es werden *p* Punkte aus einem konvexen Raum ausgewählt. Als Anwendung wird das Circle-packing-Problem mit dem Einheitsquadrat als Container betrachtet. Die Lösung erfolgt als nichtlineares Programm für 10-24 Kreise mit MINOS[48], einer Software zur Lösung nichtlinearer Probleme. Um möglichst eine global optimale Lösung zu erhalten, wurden jeweils 1000 Berechnungen mit zufällig bestimmten Startpunkten durchgeführt. Die Anzahl der Berechnungen, die dabei tatsächlich eine global optimale Lösung ergeben haben, schwankt zwischen knapp 2 Prozent bei 23 Kreisen und 33 Prozent bei 12 Kreisen.

Ein Branch-and-Bound-Verfahren nutzen Welch et al. (2006) für die Lösung eines diskreten Problems, das Facilities in einem Quadrat mit der Seitenlänge 125 platziert. Die Autoren geben Berechnungen mit unterschiedlicher Anzahl bereits bestehender (20 – 120) und zu platzierender Facilities (3 – 5) an. Die Ergebnisse zeigen, dass der Rechenaufwand sehr stark von der Anzahl neu zu platzierender Objekte abhängt und im Mittel bereits für vier Objekte schon deutlich über 40 Sekunden Rechenzeit benötigt.

Andere Arbeiten der letzten Jahre zeigen, dass durch den Einsatz spezieller Verfahren für bis zu zwei neu zu platzierende Facilities (Moshe et al. 2000, Katz et al. 2002, Tamir 2006) in Abhängigkeit von der Anzahl der bereits vorhandenen Objekte eine effiziente Lösung mit quasilinearem Aufwand möglich ist. Die Verfahren basieren auf der geometrischen Betrachtung der Vereinigung und Überlappung von Kreisen um die bereits bestehenden Facilities.

6.3.3 Komplexität von Distanzproblemen

Die Komplexität des Farbproblems bzw. allgemein von Distanzproblemen (das Vorhandensein lokaler und globaler Lösungen, Kombinatorik und Symmetrie) lässt sich anhand einiger einfacher Betrachtungen verdeutlichen. Dazu wird im Folgenden aus Gründen der Anschaulichkeit das Problem MAXMIN 1 in der Ebene auf die Platzierung von Punkten in einem Quadrat angewandt. Die Visualisierung erfolgt in der Modellvorstellung der Packprobleme, d.h. durch die Platzierung von Kreisen.

Abbildung 6-11 verdeutlicht die Problematik lokaler und globaler Minima. Es sind jeweils vier Punkte vorgegeben, ein fünfter soll durch Anwendung eines lokalen Verfahrens (SQP-Verfahren, Abschnitt 6.5.3.2) zusätzlich platziert werden. Im ersten Fall (linker Teil der Ab-

[48] http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm (Zuletzt geprüft am: 01.10.2008)

bildung) sind die gegebenen Punkte in den Eckpunkten des Quadrats festgelegt, der neu zu bestimmende Punkt würde global optimal im Zentrum des Quadrats platziert.

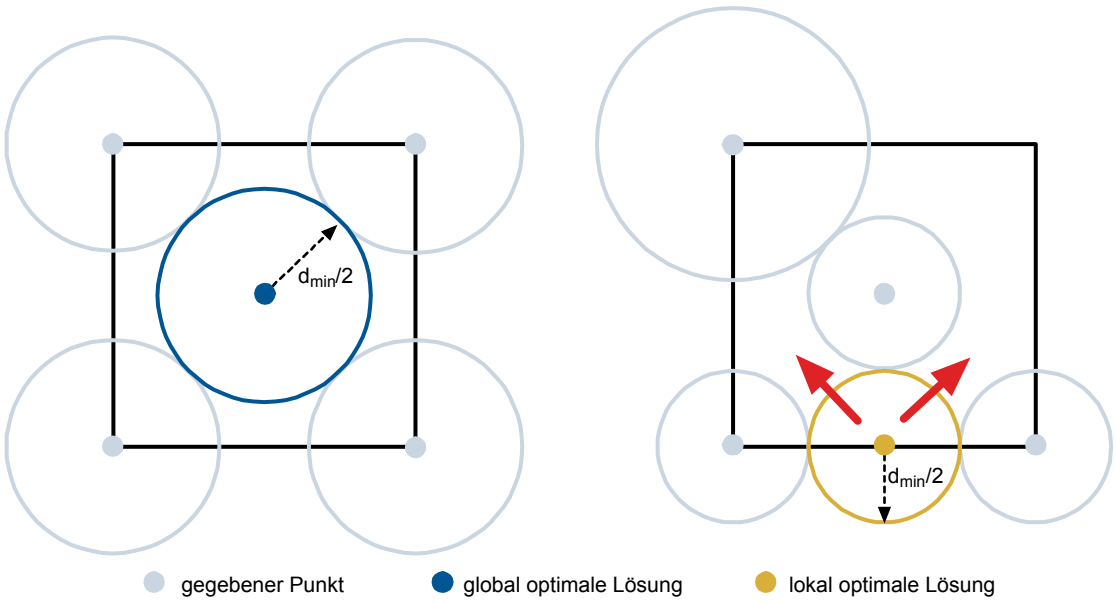


Abbildung 6-11: Platzierung eines Punktes bei vier gegebenen Punkten in der Ebene: Global optimale Lösung (links); lokal optimale Lösung durch Einkreisen des neuen Punktes durch die gegebenen Punkte (rechts)

Anders verhält es sich bei der in Abbildung 6-11 rechts dargestellten Konstellation der gegebenen Punkte. Würde der Startpunkt so gewählt, dass im Verlauf der Berechnung die gezeigte Situation auftritt, wäre der zu platzierende Punkt durch den im Verlauf der Berechnung größer werdenden minimalen Abstand zwischen den gegebenen Punkten „gefangen“. Er könnte das dargestellte lokale Minimum nur noch um den Preis der Verschlechterung der Zielfunktion verlassen und zum globalen Optimum „wandern“.

Eine detaillierte Betrachtung des Auffindens lokaler Minima, die nicht global optimal sind, ist anhand der Punktkonstellationen im Verlauf der Iterationen eines lokalen Verfahrens möglich. Dafür sollen nun fünf Punkte im Einheitsquadrat platziert werden. Die global optimale Lösung wurde bereits in Abbildung 6-11 links dargestellt. Bei der Wahl geeigneter Startpunkte wird diese Lösung auch erhalten. Anders verhält es sich, wenn beim Start alle Punkte im Ursprung liegen. Abbildung 6-12 verdeutlicht den Verlauf der Iterationen bei Anwendung des SQP-Verfahrens (Abschnitt 6.5.3.2).

Die Punkte wandern zunächst gemeinsam – aber unabhängig voneinander – vom Ursprung entlang der Diagonalen des Einheitsquadrats. Da die zu maximierende Distanz in den ersten Schritten noch sehr gering ist, liegen die Punkte sehr nahe beieinander. Nach der vierten Iteration haben sie sich dann explosionsartig voneinander entfernt. Allerdings bewegt sich dabei nur noch ein Teil der Punkte (3, 4, 5) in größerem Maße: Sie sorgen für eine Verbesserung des Wertes der Zielfunktion, indem sie sich in Richtung der Pfeile bewegen. Punkt 1 und 2 sind zu diesem Zeitpunkt schon weitgehend festgelegt, d.h. es ist ohne Verschlechterung der Zielfunktion keine größere Bewegung mehr möglich. Der Zustand nach Iteration 13 zeigt,

dass das erhaltene Ergebnis lediglich ein lokales Minimum darstellt (eine minimale Distanz von 0,6417 gegenüber 0,707 im optimalen Fall). Allerdings lässt dieser Zustand keine weitere Bewegung eines Punktes mehr zu: Jede Änderung würde den Abstand zu einem anderen Punkt und damit die minimale Distanz verringern.

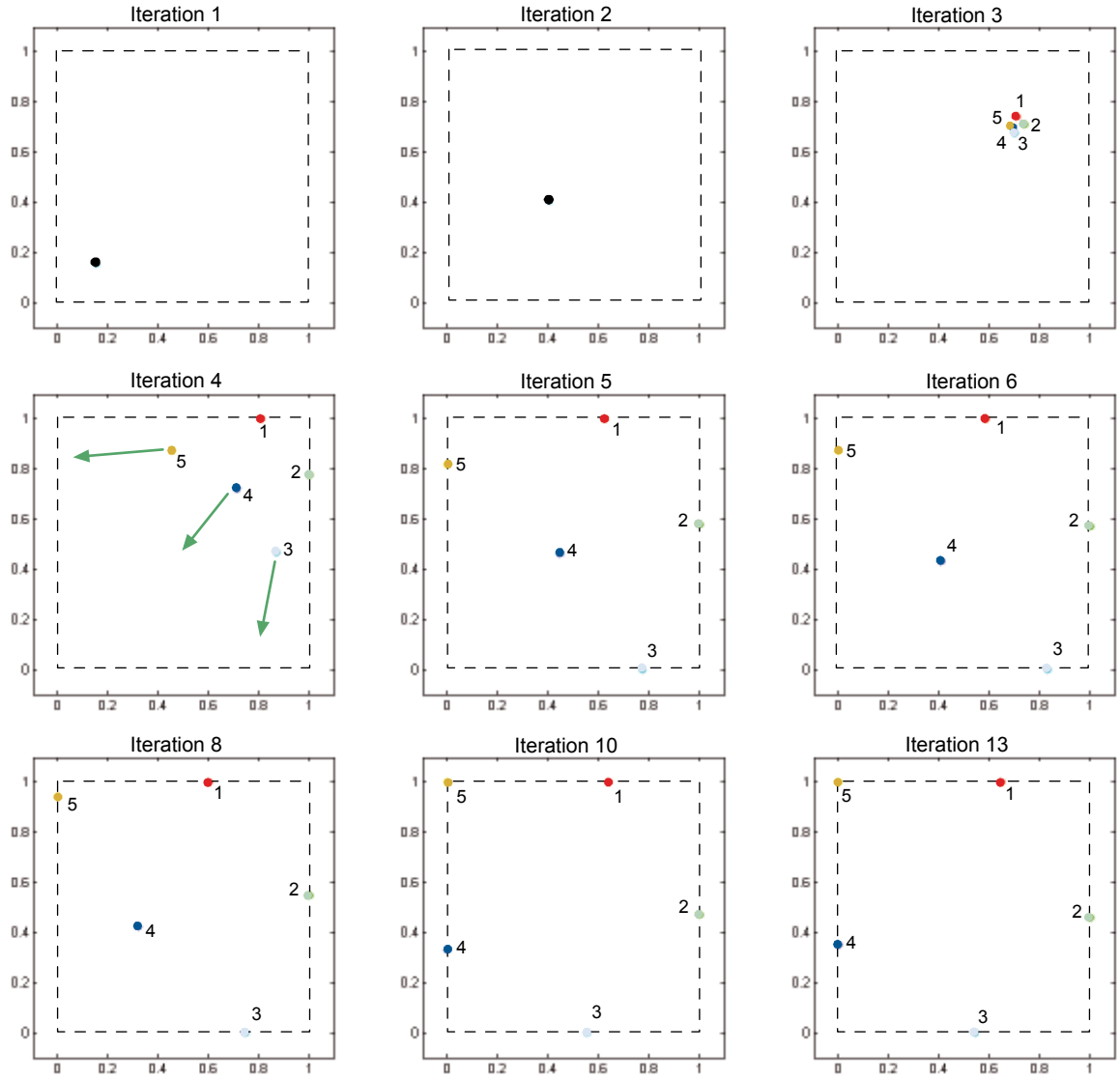


Abbildung 6-12: Iterationszustände bei der Platzierung von fünf Punkten im Einheitsquadrat

Die Kombinatorik des Farbproblems lässt sich ebenfalls gut an einem einfachen Beispiel in der Ebene verdeutlichen. Es werde wieder angenommen, dass bereits fünf Punkte in einem Quadrat platziert sind (Abbildung 6-13), zwei weitere Punkte sollen zusätzlich hinzugefügt werden. Nach Abbildung 6-13 existieren offensichtlich vier gleichwertige Punkte, die als Lösung in Frage kommen. Eine tatsächliche Lösung kann jede Auswahl von zwei Punkten aus diesen vier Punkten sein. Die Lösung wäre dann in diesem Beispiel global optimal aber nicht eindeutig. Ähnliche Konstellationen lassen sich selbstverständlich auch für lokal optimale Lösungen angeben.

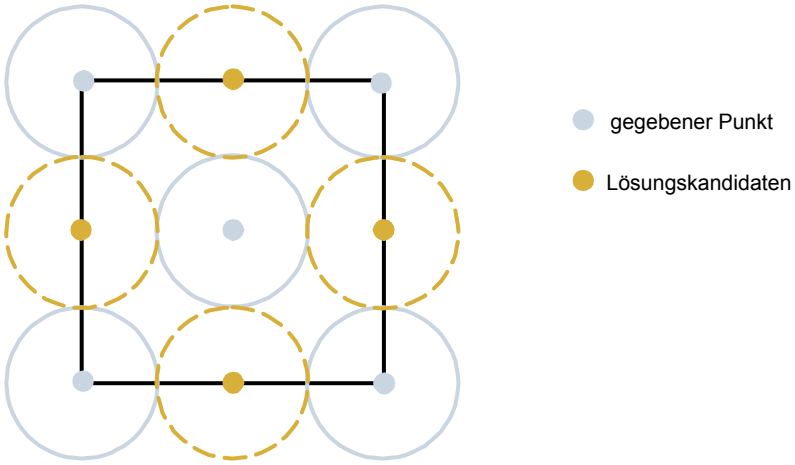


Abbildung 6-13: Kombinatorik und Symmetrie für die Platzierung eines Punktes bei fünf gegebenen Punkten in der Ebene

Die Symmetrie des Problems findet ihren Ausdruck in der Permutation von Lösungen. Unter der Annahme, dass im Beispiel der Abbildung 6-13 vier statt zwei neue Punkte gesucht sind, ist aus numerischer Sicht jede Permutation dieser Punkte eine mögliche Lösung. Für zwei gesuchte Punkte X_u und X_v der Menge X drückt sich die Symmetrie in der mathematischen Formulierung durch die Austauschbarkeit von Punkten bei gleicher Zielfunktion aus:

$$X^1 = (X_1, \ldots, X_u, X_v, \ldots, X_n)$$
$$X^2 = (X_1, \ldots, X_v, X_u, \ldots, X_n)$$

mit $d_{\min} \leq \left\| X_u^1, X_v^1 \right\|_2 = \left\| X_v^2, X_u^2 \right\|_2$.

6.3.4 Zwischenresümee und weiteres Vorgehen

Die Differenzierung der nichtlinearen Optimierung in lokale und globale Optimierung machte bereits deutlich, dass die Lösung nichtlinearer Programme im Spannungsfeld von globaler Optimalität und effizienter Lösbarkeit steht. Dies wurde durch die Betrachtung von Problemen der Distanzoptimierung, im Wesentlichen der Facility Location, bestätigt: Die Lösung dieser Probleme erfolgt zum Teil unter recht speziellen Rahmenbedingungen mit unterschiedlichen Verfahren. Allerdings ist zum jetzigen Zeitpunkt kein Standardverfahren verfügbar, dass jegliche Probleme dieser Art effizient und global optimal löst.

Ein Verfahren zur Lösung des Farbproblems wird im nächsten Kapitel detailliert beschrieben. Als Grundlage dieses Verfahrens dienen einige Standardverfahren und Lösungsparadigmen der Algorithmischen Geometrie und der Mathematischen Optimierung, die im weiteren Verlauf dieses Kapitels vorgestellt werden. Aus dem Bereich der Algorithmischen Geometrie wird ein Verfahren beschrieben, das eine Lösung des Farbproblems durch die Berechnung eines Voronoi-Diagramms ermöglicht. Verfahren zur numerischen Lösung des Problems MAXMIN 1 sind in der Mathematischen Optimierung und der Künstlichen Intelligenz verfügbar. Eine Übersicht über die in dieser Arbeit vorgestellten Methoden gibt Abbildung 6-14.

Gemäß der Differenzierung zwischen lokaler und globaler Optimierung erfolgt eine Unterscheidung in lokale und globale Verfahren. Für die lokalen Verfahren wird weiterhin zwischen unrestringierten und restringierten Problemen differenziert.

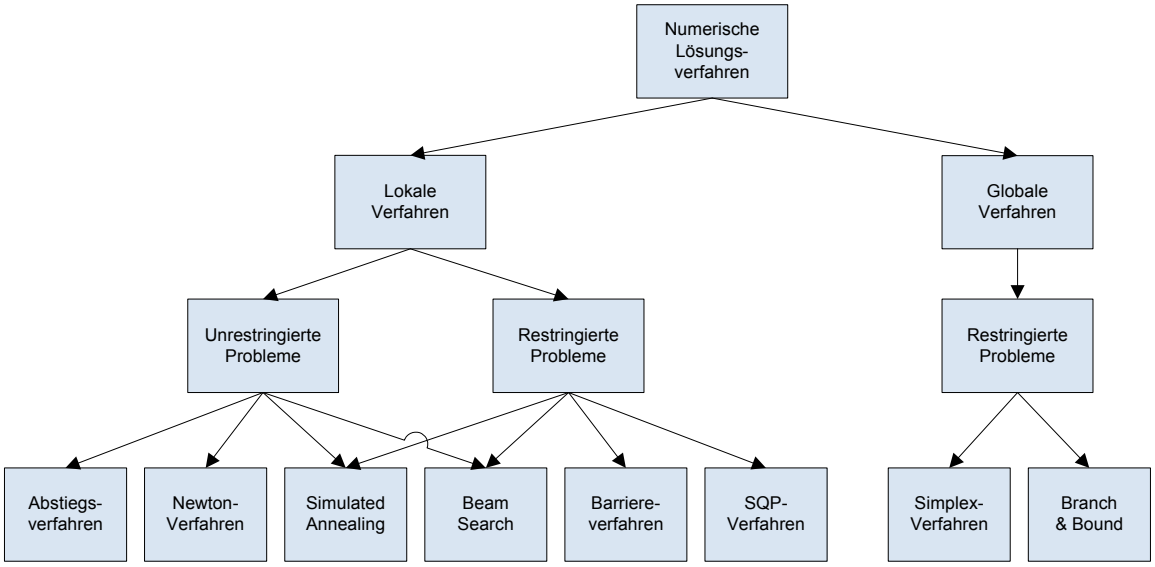


Abbildung 6-14: Übersicht über die in dieser Arbeit betrachteten Lösungsverfahren

Es sei klarstellend bemerkt, dass nicht alle der genannten Methoden direkt auf das Farbproblem anwendbar sind. Gradienten- und Newton-Verfahren sind in der unten beschriebenen Form zunächst nur für unrestringierte Probleme nutzbar, eignen sich aber gut zur Darstellung von Lösungsprinzipien und sind darüber hinaus Basis für Barriere- und SQP-Verfahren. Das Simplex-Verfahren, das ausschließlich auf lineare Programme anwendbar ist, verdeutlicht zusammen mit dem Barriere-Verfahren zwei wesentliche Lösungsparadigmen der Mathematischen Optimierung: Die Suche nach einer Lösung auf dem Rand der zulässigen Menge und die Annäherung an eine optimale Lösung aus dem Inneren der zulässigen Menge heraus.

Eine Bewertung der vorgestellten Verfahren in ihrer Anwendung auf das Farbproblem erfolgt im Zusammenhang mit dem eigentlichen Lösungsverfahren im nächsten Kapitel.

6.4 Geometrisches Verfahren zur Lösung von Distanzproblemen

Im Abschnitt 6.2.2.1 wurden bereits Optimierungsprobleme beschrieben, die durch Verwendung von Voronoi-Diagrammen lösbar sind. Das Problem der Platzierung eines Punktes, dessen minimaler Abstand zu *m* vorhandenen Punkten maximal wird, wurde meist im zweidimensionalen Raum behandelt und stellt aus Sicht geometrischer Betrachtungen mit dem Voronoi-Diagramm das Problem des *größten leeren Kreises* (*largest empty circle*) dar (Okabe & Suzuki 1997). Dieser ist der größte leere Kreis, dessen Mittelpunkt so in einer Region *S* mit *m* vorhandenen Objekten platzierbar ist, dass keines der bereits vorhandenen Objekte im Kreis enthalten ist. Der Mittelpunkt ist dann der gesuchte Punkt.

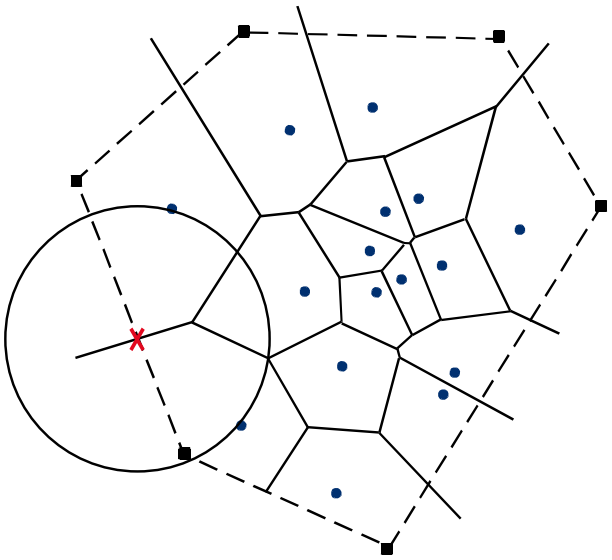


Abbildung 6-15: Platzierung des größten leeren Kreises anhand eines Voronoi-Diagramms (nach Toussaint 1983)

Das Problem des größten leeren Kreises wird ausführlich von Toussaint (1983) betrachtet (vgl. auch Preparata & Shamos 1985). Gegeben seien wiederum m Objekte q_i, $i=1,...,m$; die Region S sei ein beliebiges Polygon P. Wird für die Objekte das Voronoi-Diagramm berechnet, ist der Mittelpunkt eines größten leeren Kreises in der Menge der folgenden Punkte zu suchen (Toussaint 1983):

- Voronoi-Knoten,
- Schnittpunkte zwischen Voronoi-Kanten und Kanten des Polygons P,
- Knoten des Polygons P.

Abbildung 6-15 zeigt die betreffenden Punkte und den für diese Konstellation größten leeren Kreis. Anhand dieser Abbildung wird auch die Gültigkeit der obigen Aussage deutlich (für den detaillierten Beweis siehe Toussaint (1983)): Das Voronoi-Diagramm und dessen Schnitt mit dem Polygon P zerlegen P in m Subpolygone, jedes dieser Polygone enthält ein vorhandenes Objekt q_i. Ein beliebiger in P platzierter Punkt x sei nun Mittelpunkt eines leeren Kreises. Offensichtlich liegt x in einem Subpolygon V_j, der Radius des leeren Kreises ist $||x,q_j||_2$. Dieser Radius lässt sich durch Verschiebung von x so lange vergrößern, bis x ein Eckpunkt von V_j ist. Dementsprechend sind alle Eckpunkte der Subpolygone – und damit die oben aufgezählten Punkte – Kandidaten für den Mittelpunkt des größten leeren Kreises.

Ein neu hinzuzufügender Punkt würde im Mittelpunkt des größten leeren Kreises platziert. Mehrere Neupunkte könnten durch wiederholte Berechnung des größten leeren Kreises sukzessive eingefügt werden. Eine vertiefende Betrachtung dieses Vorgehens erfolgt im Abschnitt 7.1.

Ein Verfahren zur Bestimmung des größten leeren Kreises ist in Algorithmus 6-1 angegeben. Wie bei der konzeptionellen Darstellung von Algorithmen allgemein üblich, sind dieses und alle weiteren Verfahren in dieser Arbeit in Pseudocode formuliert.

Gegeben: *Eine Menge von Punkten* $Q = \{q_1,...,q_m\}$
Ein Polygon P, festgelegt durch eine Menge von Punkten $\{p_1, ..., p_n\}$
Gesucht: *Größter leerer Kreis mit Mittelpunkt in P, der keinen Punkt aus Q enthält*

1. *Berechne das Voronoi-Diagramm* $\mathsf{V(Q)}$*;*
2. *Prüfe, welche Voronoi-Knoten von* $\mathsf{V(Q)}$ *in* P *liegen und berechne die größten leeren Kreise der in* P *liegenden Punkte;*
3. *Berechne die Schnittpunkte* $\mathsf{T} = \{\mathsf{t}_1, ..., \mathsf{t}_k\}$ *der Voronoi-Kanten mit den Kanten von* P *und berechne für jedes* $\mathsf{t_i}$ *den größten leeren Kreis;*
4. *Bestimme für jedes* $\mathsf{p_i} \in \mathsf{P}$ *den nächsten Nachbarn* $\mathsf{q_j} \in \mathsf{Q}$ *und berechne seinen größten leeren Kreis;*
5. *Wähle den größten aller gefundenen leeren Kreise aus;*

Algorithmus 6-1: Verfahren zur Bestimmung des größten leeren Kreises in einer Menge von Punkten (nach Toussaint 1983)

Die Berechnung des größten leeren Kreises ist auf den dreidimensionalen Raum und die Berechnung der *größten leeren Kugel* übertragbar, was sich durch analoge Betrachtung des zweidimensionalen Falls für Polyeder im 3D-Raum bestätigen lässt. Weitere Ausführungen zum Vorgehen erfolgen im nächsten Kapitel im Rahmen der Lösung des Farbproblems.

6.5 Lokale Verfahren

Im Abschnitt 6.3.4 wurden bereits die Verfahren der (lokalen) Mathematischen Optimierung und der Künstlichen Intelligenz, die im Folgenden vorgestellt werden, genannt. Bevor eine vertiefende Beschreibung erfolgt, sind für das Verständnis der Verfahren der Mathematischen Optimierung noch einige Grundlagen erforderlich.

6.5.1 Allgemeine Grundlagen

Ziel der nachfolgend vorgestellten Verfahren der Mathematischen Optimierung – Abstiegs-, Newton-, Barriere- und SQP-Verfahren – ist es, Punkte zu bestimmen, die den *notwendigen Optimalitätsbedingungen* genügen, d.h. Bedingungen, die ein Punkt mindestens erfüllen muss, um Minimum eines Optimierungsproblems zu sein. Dies erfolgt meist durch numerische Methoden: Die Gleichungen der Optimalitätsbedingungen werden nicht exakt gelöst, sondern es werden Näherungslösungen bestimmt, indem iterativ Folgen von Punkten $x^{(0)}, x^{(1)}, \ldots, x^{(k)} = \{x^{(k)}\}$ berechnet werden, die gegen den gesuchten Punkt $\bar{x}$ konvergieren. Die Konvergenz eines Verfahrens ist dabei nicht selbstverständlich, sondern im Einzelnen nachzuweisen[49]. Die Definition der wichtigsten Konvergenzraten findet sich im Anhang A.5.

[49] Diese Arbeit beschränkt sich auf eine kompakte Beschreibung der Verfahren. Für tiefer gehende Ausführungen und insbesondere Nachweise zur Konvergenz wird auf die jeweilige Literatur verwiesen.

Im Folgenden werden neben notwendigen auch *hinreichende Optimalitätsbedingungen*, differenziert nach unrestringierten und restringierten Problemen, zusammengefasst. Während die Bedingungen unrestringierter Probleme hier als weitgehend bekannt vorausgesetzt und in Kürze angegeben werden, sind die der restringierten Probleme im GIS-Kontext weniger bekannt und werden ausführlicher dargestellt. Basis der Optimalitätsbedingungen restringierter Probleme ist die *Lagrange-Funktion*, die ebenfalls Teil dieses Abschnitts ist.

6.5.1.1 Optimalitätsbedingungen unrestringierter Probleme

In diesem Abschnitt wird von einem unrestringierten Problem der Form UP (Abschnitt 6.2.3.2) ausgegangen. Für die Zielfunktion gelte $f_0 : D \to \Re$, $D \subset \Re^n$ sei nichtleer und offen. Die Lösung des Problems soll durch notwendige Bedingungen erster und zweiter Ordnung sowie hinreichende Bedingungen zweiter Ordnung charakterisiert werden. Quelle der folgenden Ausführungen ist Alt (2002), dort finden sich auch ausführlichere Beschreibungen und Beweise.

Notwendige Bedingung erster Ordnung

In einem lokalen Minimum $\bar{x}$ muss für eine beliebige Richtung $d \in \Re^n$ und für ein hinreichend kleines $|t|$ offensichtlich $f_0(\bar{x} + td) \geq f_0(\bar{x})$ gelten.

Daraus folgt für eine in $\bar{x} \in D$ differenzierbare Zielfunktion f_0: Ist $\bar{x}$ ein lokales Minimum, dann gilt

$$\nabla f_0(\bar{x})^T d \geq 0 \qquad \forall d \in \Re^n .$$

Aus

$$\nabla f_0(\bar{x})^T (-d) = -\nabla f_0(\bar{x})^T d \geq 0 \qquad \forall d \in \Re^n$$

folgt dann die *notwendige Bedingung* für ein lokales Minimum:

$$\nabla f_0(\bar{x}) = 0 .$$

Ein Punkt, der diese Bedingung erfüllt, wird auch als *stationärer Punkt* von f_0 bezeichnet. Ein stationärer Punkt kann sowohl lokales Minimum als auch lokales Maximum von f_0 sein.

Notwendige Bedingung zweiter Ordnung

Die notwendige Bedingung zweiter Ordnung erlaubt die Differenzierung von lokalen Minima und Maxima:

f_0 sei in einer Umgebung von $\bar{x} \in D$ zweimal stetig differenzierbar. Ist $\bar{x}$ ein lokales Minimum, dann gilt die Bedingung erster Ordnung und

$$x^T \nabla^2 f_0(\bar{x}) x \geq 0 \qquad \forall x \in \Re^n ,$$

d.h. die Hesse-Matrix ist positiv semidefinit.

Hinreichende Bedingung zweiter Ordnung

Eine hinreichende Bedingung wird durch eine Verschärfung der notwendigen Bedingung zweiter Ordnung erhalten:

f_0 sei in einer Umgebung von $\bar{x} \in D$ zweimal stetig differenzierbar. Falls die notwendige Bedingung erster Ordnung erfüllt ist, und mit einem $r > 0$ für alle $z \in B(\bar{x}, r)$

$$x^T \nabla^2 f_0(z) x \geq 0 \qquad \forall x \in \Re^n$$

gilt, dann ist $\bar{x}$ lokales Minimum von f_0.

Falls die Hesse-Matrix positiv definit ist, d.h.

$$x^T \nabla^2 f_0(\bar{x}) x > 0 \qquad \forall x \in \Re^n,\ x \neq 0$$

gilt, gibt es $\alpha > 0$, $\beta > 0$ mit

$$f_0(x) \geq f_0(\bar{x}) + \beta \parallel x - \bar{x} \parallel_2^2 \qquad \forall x \in B(\bar{x}, \alpha)$$

und $\bar{x}$ ist striktes lokales Minimum.

6.5.1.2 Lagrange-Funktion und Lagrange-Dualität

Die Formulierung von Optimalitätsbedingungen für restringierte Probleme erfolgt durch die *Lagrange-Funktion*. Mit der *Lagrange-Dualität* ist unter Anwendung der Lagrange-Funktion ein weiteres wichtiges Konzept der Mathematischen Optimierung, das u.a. bei numerischen Methoden zum Einsatz kommt (vgl. Barriere-Verfahren im Abschnitt 6.5.3.1), definiert. Die Inhalte dieses Abschnitts sind entnommen aus Boyd & Vandenberghe (2004).

Die Lagrange-Funktion $L : \Re^n \times \Re^m \times \Re^p \rightarrow \Re$ erweitert die Zielfunktion des Problems SFP um die gewichtete Summe der Nebenbedingungen:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

mit $dom\ L = D \times \Re^m \times \Re^p$. Die λ_i, ν_i werden als *Lagrange-Multiplikatoren* bezeichnet, die Vektoren λ, ν als *Vektoren der Lagrange-Multiplikatoren* oder *duale Variablen* des Problems SFP.

Die *(Lagrangsche) duale Funktion* $g : \Re^m \times \Re^p \rightarrow \Re$ ist durch das Infimum der Lagrange-Funktion definiert:

$$g(\lambda, \upsilon) = \inf_{x \in D} L(x, \lambda, \nu) = \inf_{x \in D} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \text{ für } \lambda \in \Re^m, \nu \in \Re^p.$$

Die duale Funktion gibt für jedes Paar (λ, ν) und $\lambda \succeq 0$ eine untere Schranke der optimalen Lösung $\bar{p}$ des Problems SFP an, von Interesse ist dann die beste untere Schranke. Diese lässt sich über ein Maximierungsproblem, das *Lagrangsche duale Problem*, bestimmen:

DSFP	maximize	$g(\lambda, \upsilon)$
	subject to	$\lambda \succeq 0$.

Das Problem SFP wird in diesem Kontext auch als *primales Problem* bezeichnet. Die Ungleichung

$$\overline{q} \leq \overline{p},$$

mit $\overline{q}$ als Optimum der dualen Funktion, bezeichnet eine *schwache Dualität*. Falls

$$\overline{q} = \overline{p},$$

gilt eine *starke Dualität.* Dies ist in der Regel dann der Fall, wenn das primale Problem konvex ist.

Die Lücke zwischen primaler und dualer Zielfunktion, die *Dualitätslücke*, ist gegeben durch

$$f_0(x) - g(\lambda, \nu)$$

für einen primal zulässigen Punkt x und einen dual zulässigen Punkt (λ, ν). Im Fall einer starken Dualität gilt $f_0(x) = g(\lambda, \nu)$, d.h. die Dualitätslücke ist Null.

6.5.1.3 Optimalitätsbedingungen restringierter Probleme

Für die Optimalitätsbedingungen restringierter Probleme wird von der Form SFP ausgegangen:

$$\begin{array}{lll} \text{SFP} & \text{minimize} & f_0(x) \\ & \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & & h_i(x) = 0, \quad i = 1, \ldots, p\,. \end{array}$$

Die Optimalitätsbedingungen des unrestringierten Falles sind für dieses Problem offensichtlich zu eng gefasst, so ist beispielsweise nicht sichergestellt, dass ein stationärer Punkt der Zielfunktion in der Menge der zulässigen Punkte liegt. Im Folgenden wird deshalb die Lösung eines restringierten Problems durch gesonderte notwendige Bedingungen erster Ordnung und hinreichende Bedingungen zweiter Ordnung charakterisiert.

Notwendige Bedingungen erster Ordnung

Die Formulierung der notwendigen Bedingungen erster Ordnung erfolgt mit Hilfe der Lagrange-Funktion und der Lagrange-Multiplikatoren. Es gilt damit (vgl. Fletcher 1987, Boyd & Vandenberghe 2004):

Falls $\overline{x}$ ein lokales Minimum des Problems SFP ist und eine Regularitätsbedingung (s.u.) in $\overline{x}$ erfüllt ist, dann existieren Lagrange-Multiplikatoren $\overline{\lambda}, \overline{\nu}$, so dass $\overline{x}, \overline{\lambda}, \overline{\nu}$ die folgenden Gleichungen erfüllen:

$$f_i(\overline{x}) \leq 0, \quad i = 1, \ldots, m,$$

$$h_i(\overline{x}) = 0, \quad i = 1, \ldots, p,$$

$$\overline{\lambda}_i \geq 0, \quad i = 1, \ldots, m,$$

$$\overline{\lambda}_i \; f_i(\overline{x}) = 0, \quad i = 1, \ldots, m,$$

$$\nabla_x L(\bar{x},\bar{\lambda},\bar{\nu}) = \nabla f_0(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla f_i(\bar{x}) + \sum_{i=1}^{p} \bar{\nu}_i \nabla h_i(\bar{x}) = 0 .$$

Die letzte Bedingung enthält eine Linearisierung der Nebenbedingungen. Um sicherzustellen, dass eine Lösung des Problems NLP auch eine Lösung dieses linearisierten Problems darstellt, wird *Regularität* gefordert. Diese kann durch verschiedene Bedingungen, im Englischen als *constraint qualifications* bezeichnet, erreicht werden. Eine mögliche Regularitätsbedingung ist beispielsweise die Forderung der linearen Unabhängigkeit der Gradienten der in $\bar{x}$ aktiven Ungleichungsnebenbedingungen und der Gradienten der Gleichheitsnebenbedingungen (Geiger & Kanzow 2002).

Die Bedingung $\bar{\lambda}_i \; f_i(\bar{x}) = 0$ wird als *Komplementaritäts-Bedingung* bezeichnet (Fletcher 1987). Diese stellt sicher, dass in einem Punkt $\bar{x},\bar{\lambda},\bar{\nu}$ immer $\bar{\lambda}_i = 0$ oder $f_i(\bar{x}) = 0$ gilt und so nur die in $\bar{x}$ aktiven Ungleichungsnebenbedingungen betrachtet werden. Falls kein Index *i* mit $\bar{\lambda}_i = f_i(\bar{x}) = 0$ existiert, gilt *strikte Komplementarität* (ebd.).

Die obigen fünf Bedingungen werden auch als Karush-Kuhn-Tucker-Bedingungen (KKT-Bedingungen) bezeichnet (Geiger & Kanzow 2002).

Für konvexe Optimierungsprobleme werden die notwendigen Optimalitätsbedingungen häufig als Sattelpunktbedingungen angegeben (vgl. Geiger & Kanzow 2002).

Als Sattelpunkt der Lagrange-Funktion L wird ein Vektor $(\bar{x},\bar{\lambda},\bar{\nu}) \in \Re^n \times \Re^m \times \Re^p$ mit $\bar{\lambda} \geq 0$ bezeichnet, wenn

$$L(\bar{x},\lambda,\nu) \leq L(\bar{x},\bar{\lambda},\bar{\nu}) \leq L(x,\bar{\lambda},\bar{\nu})$$

für alle $(x,\lambda,\nu) \in \Re^n \times \Re^m \times \Re^p$ mit $\lambda \geq 0$ gilt (Geiger & Kanzow 2002). Als notwendige Optimalitätsbedingung ist dann formulierbar (ebd.):

Für ein konvexes Optimierungsproblem ist $\bar{x}$ ein globales Minimum, falls $(\bar{x},\bar{\lambda},\bar{\nu}) \in \Re^n \times \Re^m \times \Re^p$ ein Sattelpunkt der Lagrange-Funktion L ist.

Hinreichende Bedingungen zweiter Ordnung

Für die hinreichenden Bedingungen zweiter Ordnung gilt (Alt 2002):

f_i, h_i seien auf D zweimal stetig differenzierbar und es gebe Lagrange-Multiplikatoren $\bar{\lambda},\bar{\nu}$, so dass die Bedingungen erster Ordnung erfüllt sind. Es gelte darüber hinaus mit einer Konstanten $\alpha > 0$

$$d^T \nabla^2_{xx} L(\bar{x},\bar{\lambda},\bar{\nu}) d \geq \alpha \,\|\, d \,\|^2$$

für alle $d \in \Re^n$ und

$\nabla h_i(\bar{x})^T d = 0 , \qquad i = 1,\ldots,p ,$

$\nabla f_i(\bar{x})^T d = 0$ für die Indizes aus $i = 1,\ldots,m$, für die $\lambda_i > 0$, und

$\nabla f_i(\bar{x})^T d \leq 0$ für die Indizes aus $i = 1,\ldots,m$, für die $\lambda_i = 0$.

Dann ist $\bar{x}$ ein striktes lokales Minimum.

6.5.2 Abstiegsverfahren zur Lösung unrestringierter Probleme

Für die Beschreibung der Verfahren zur Lösung unrestringierter Probleme wird wieder von der Form UP ausgegangen:

UP minimize $f_0(x)$.

Es ist hier also lediglich die Zielfunktion zu minimieren.

Im Folgenden wird zunächst die allgemeine Form eines *Abstiegsverfahrens*, als spezielle Anwendung das *Gradientenverfahren* angegeben. Grundlage für die Konstruktion weiterer spezieller Abstiegsverfahren ist das *Newton-Verfahren*, das abschließend beschrieben wird.

6.5.2.1 Allgemeines Abstiegsverfahren

Durch Abstiegsverfahren wird für eine differenzierbare Zielfunktion f_0 eine Lösung der notwendigen Optimalitätsbedingung $\nabla f_0(x) = 0$ bestimmt, indem durch

$$x^{(k+1)} = x^{(k)} + \sigma^{(k)} d^{(k)}$$

iterativ eine absteigende Folge von Punkten $x^{(k)}$, $k = 0,1,\ldots$, berechnet wird (Boyd & Vandenberghe 2004, Alt 2002). Der Skalar $\sigma^{(k)} \geq 0$ wird als *Schrittweite*, der Vektor $d^{(k)}$ als *Suchrichtung* bezeichnet (Boyd & Vandenberghe 2004).

Für ein Abstiegsverfahren gilt offensichtlich $f_0\left(x^{(k+1)}\right) < f_0\left(x^{(k)}\right)$, solange $x^{(k)}$ nicht optimal ist. Die Suchrichtung muss in diesem Fall die Bedingung

$$\nabla f_0\left(x^{(k)}\right)^T d^{(k)} < 0$$

erfüllen und wird dann als Abstiegsrichtung bezeichnet (ebd.).

Der Ablauf eines allgemeinen Abstiegsverfahrens ist im Algorithmus 6-2 zusammengefasst.

Gegeben: *Startpunkt* $x^{(0)} \in \Re^n$
Gesucht: *Lösung der notwendigen Optimalitätsbedingung* $\nabla f_0(x) = 0$

Setze $\mathsf{k} = 0$*;*

do

Bestimme eine Abstiegsrichtung $\mathsf{d}^{(\mathsf{k})}$ *und eine Schrittweite* $\sigma^{(\mathsf{k})} > 0$,
so dass $\mathsf{f}_0(\mathsf{x}^{(\mathsf{k})} + \sigma^{(\mathsf{k})}\mathsf{d}^{(\mathsf{k})}) < \mathsf{f}_0(\mathsf{x}^{(\mathsf{k})})$*;*

Setze $\mathsf{x}^{(\mathsf{k}+1)} = \mathsf{x}^{(\mathsf{k})} + \sigma^{(\mathsf{k})}\mathsf{d}^{(\mathsf{k})}$*;*

Setze $\mathsf{k} = \mathsf{k} + 1$*;*

while *Abbruchkriterium nicht erfüllt;*

Algorithmus 6-2: Allgemeines Abstiegsverfahren (nach Boyd & Vandenberghe 2004, Alt 2002)

Das Abbruchkriterium ist erfüllt, wenn $\nabla f_0(x) = 0$ gilt. Für praktische Berechnungen eignet sich dieses Kriterium allerdings nicht. Stattdessen wird getestet, ob $\left\|\nabla f_0\left(x^{(k)}\right)\right\|_2 < \varepsilon_1$, $\left| f_0\left(x^{(k+1)}\right) - f_0\left(x^{(k)}\right)\right| < \varepsilon_2$ oder $\left\|x^{(k+1)} - x^{(k)}\right\|_2 < \varepsilon_3$ (Alt 2002).

Der dargestellte Ablauf beschreibt allgemein eine Klasse von Verfahren. Spezielle Abstiegsverfahren lassen sich durch die Art der Berechnung von Schrittweite und Suchrichtung konstruieren; davon abhängig ist die Konvergenz eines Verfahrens, die im Einzelnen betrachtet werden muss.

Die Forderung an die Schrittweite ist, dass sie einerseits so gewählt wird, dass der tatsächliche Abstieg dieselbe Größenordnung hat wie der Abstieg in erster Näherung, sie andererseits im Vergleich zu $\nabla f\left(x^{(k)}\right)^T d^{(k)}$ nicht zu schnell gegen 0 geht (Alt 2002). Eine Schrittweite, die diesen Bedingungen genügt, erfüllt das *Prinzip des hinreichenden Abstiegs* und wird als *effizient* bezeichnet (ebd.). Ein Beispiel für eine effiziente Schrittweite ist die *exakte Schrittweite*; dabei wird σ so gewählt, dass f_0 entlang des Strahls $\{x + \sigma d \mid \sigma \geq 0\}$ minimiert wird (Boyd & Vandenberghe 2004):

$$\sigma = \operatorname{argmin}_{s \geq 0} f(x + s\, d).$$

Für die Wahl einer Suchrichtung ist deren Bezug zum negativen Gradienten der Zielfunktion von Bedeutung, bei einer *gradientenbezogenen Suchrichtung* schließen die Suchrichtung und der negative Gradient einen spitzen Winkel ein (Alt 2002). Tiefergehende Ausführungen, insbesondere die mathematische Formulierung der Bedingungen für effiziente Schrittweiten und gradientenbezogene Suchrichtungen, findet sich ebenfalls in Alt (2002).

Die Konvergenz des allgemeinen Abstiegsverfahrens ist für gradientenbezogene Suchrichtungen und effiziente Schrittweiten gegeben: Stoppt das Verfahren nicht nach endlich vielen Schritten, dann gilt $\nabla f\left(x^{(k)}\right) \to 0$ für $k \to \infty$. Die Folge besitzt mindestens einen Häufungspunkt, für jeden dieser Punkte $\bar{x}$ gilt $\nabla f(\bar{x}) = 0$ (Alt 2002).

Ein spezielles Abstiegsverfahren ist das Verfahren des steilsten Abstiegs, das in jedem Iterationsschritt diejenige Richtung $\bar{d}$ unter allen auf 1 normierten Abstiegsrichtungen d nutzt, in der f_0 am stärksten abnimmt (Alt 2002). Damit ist $\bar{d}$ als Lösung des Optimierungsproblems

$$\begin{aligned} \underset{d \in \Re^n}{\text{minimize}} \quad & \nabla f(x)^T d \\ \text{subject to} \quad & \| d \| = 1 \end{aligned}$$

zu bestimmen, $\| \cdot \|$ kann jede Norm des $\Re^n$ sein (Boyd & Vandenberghe 2004, Alt 2002).

Die Lösung dieses Problems ist für in $\bar{x}$ differenzierbare f_0, $\nabla f_0 \neq 0$ und die Euklidische Norm gegeben durch

$$\bar{d} = -\frac{\nabla f_0(x)}{\| \nabla f_0(x) \|_2}.$$

Damit ist die Richtung des steilsten Abstiegs von f_0 in $\bar{x}$ durch den negativen Gradienten bestimmt. Verfahren mit dieser Suchrichtung werden auch als *Gradientenverfahren* bezeichnet.

6.5.2.2 Newton-Verfahren

Gegeben sei eine zweimal stetig differenzierbare Funktion f_0 auf $D \subset \Re^n$. D sei offen und f_0 besitze ein Minimum $\bar{x} \in D$; $\nabla^2 f_0\left(x^{(k)}\right)$ sei regulär. Gesucht ist wiederum eine Lösung $x \in \Re^n$ der notwendigen Optimalitätsbedingung $\nabla f_0(x) = 0$ (Alt 2002).

Ausgehend von einem Startpunkt $x^{(0)}$ berechnet das Newton-Verfahren in jeder Iteration eine Linearisierung der Ausgangsgleichung $\nabla f_0(x) = 0$ und findet den nächsten Iterationspunkt durch (Alt 2002)

$$x^{(k+1)} = x^{(k)} - \nabla^2 f_0\left(x^{(k)}\right)^{-1} \nabla f_0\left(x^{(k)}\right).$$

Praktische Berechnungen nutzen allerdings nicht die genannte Formel, sondern lösen in jeder Iteration das lineare Gleichungssystem (ebd.)

$$\nabla f_0\left(x^{(k)}\right) + \nabla^2 f_0\left(x^{(k)}\right)\left(x - x^{(k)}\right) = 0. \qquad (6.1)$$

Eine durch das Newton-Verfahren erzeugte Folge $\left\{x^{(k)}\right\}$ konvergiert dann für jeden Startpunkt $x^{(0)}$ in einer Umgebung von $\bar{x}$ superlinear gegen $\bar{x}$ (Alt 2002, Geiger & Kanzow 2002). Ist die Hesse-Matrix darüber hinaus in einer Umgebung von $\bar{x}$ Lipschitz-stetig[50], konvergiert das Verfahren quadratisch (Fletcher 1987, Geiger & Kanzow 2002).

Wird in (6.1) $d = x - x^{(k)}$ gesetzt, gilt zunächst

$$x^{(k+1)} = x^{(k)} + d^{(k)}$$

mit der *Newton-Richtung* $d^{(k)} = -\nabla^2 f_0\left(x^{(k)}\right)^{-1} \nabla f_0\left(x^{(k)}\right)$ (Alt 2002). Die Berechnung der Newton-Richtung erfolgt durch die Lösung des linearen Gleichungssystems

$$\nabla f_0\left(x^{(k)}\right) + \nabla^2 f_0\left(x^{(k)}\right) d = 0.$$

Dieses System ist als notwendige Optimalitätsbedingung des unrestringierten, quadratischen Optimierungsproblems

$$\underset{d \in \Re^n}{\text{minimize}} \qquad \nabla f\left(x^{(k)}\right) d + \frac{1}{2} d^T \nabla^2 \left(x^{(k)}\right) d$$

interpretierbar (zur Beschreibung quadratischer Probleme vgl. Abschnitt 6.2.3.2). Für zweimal stetig differenzierbare Zielfunktion f_0 und positiv definite Hesse-Matrix $\nabla^2 f_0(x)$ wird die Richtung $d^{(k)}$ als eindeutig bestimmte Lösung erhalten (ebd.).

Mit positiv definiter Hesse-Matrix ist die Newton-Richtung eine Abstiegsrichtung, aufgrund der stets gültigen Schrittweite von 1 aber nur dann ein Abstiegsverfahren, wenn der Startpunkt hinreichend nahe an der Lösung gewählt wird (Alt 2002). Durch Wahl einer geeigneten Schrittweite lässt sich allerdings ein Abstiegsverfahren mit größerem Konvergenzradius konstruieren (ebd.).

---

[50] Die Hesse-Matrix ist Lipschitz-stetig, falls es eine Konstante $M \geq 0$ gibt mit $\| \nabla^2 f(x) - \nabla^2 f(y) \| \leq M \| x - y \|$ $\forall x, y \in D$ (Alt 2002).

Damit ist das Newton-Verfahren Grundlage vieler Abstiegsverfahren zur Lösung unrestringierter Probleme, an späterer Stelle wird aber auch die Bedeutung für die Lösung restringierter Probleme ersichtlich werden.

6.5.3 Verfahren zur Lösung restringierter Probleme

Als Verfahren zur Lösung restringierter Probleme werden in diesem Abschnitt *Barriere-* und *SQP-Verfahren* vorgestellt. Erstere eignen sich sehr gut für die Lösung konvexer Programme und verdeutlichen das Lösungsparadigma der Innere-Punkte-Verfahren, die sich einer Optimallösung von einem Startpunkt aus dem Inneren der zulässigen Menge heraus nähern. SQP-Verfahren (*Sequentielle quadratische Programmierung*) sind sehr gut für die Lösung bzw. die Bestimmung einer lokalen Lösung allgemeiner nichtlinearer Probleme nutzbar.

6.5.3.1 Barriere-Verfahren

Gesucht sei eine Lösung des Problems COP aus Abschnitt 6.2.3.2:

$$\begin{array}{lll} \text{COP} & \text{minimize} & f_0(x) \\ & \text{subject to} & f_i(x) \le 0, \qquad i = 1,\ldots,m \\ & & Ax = b\,. \end{array}$$

Die Funktionen $f_0,\ldots,f_m : \Re^n \to \Re$ seien konvex und zweimal stetig differenzierbar, $A \in \Re^{p\times n}$ habe den Rang $p < n$ und es gebe eine optimale Lösung $\bar{x}$. Weiterhin sei das Problem strikt zulässig, d.h. es existieren Punkte $x \in D$, für die $Ax = b$ und $f_i(x) < 0, i = 1,\ldots,m,$ gilt. Damit existieren dual optimale Variablen $\bar{\lambda} \in \Re^m$ und $\bar{\nu} \in \Re^p$, die zusammen mit $\bar{x}$ die KKT-Bedingungen (Abschnitt 6.5.1.3) erfüllen.

Die Ausführungen in diesem Abschnitt basieren auf Boyd & Vandenberghe (2004).

Ziel von Barriere-Verfahren ist die Formulierung eines Problems, das die ursprüngliche Zielfunktion so um die Ungleichungsnebenbedingungen erweitert, dass die neue Zielfunktion innerhalb der zulässigen Menge der ursprünglichen Zielfunktion entspricht, außerhalb ∞ annimmt. Dieses Verhalten ist durch eine Indikatorfunktion $I_- : \Re \to \Re$ beschreibbar:

$$I_-(u) = \begin{cases} 0 & u \le 0 \\ \infty & u > 0 \end{cases}.$$

Da die Indikatorfunktion aufgrund ihrer Unstetigkeit und Nicht-Differenzierbarkeit für ein Optimierungsproblem ungeeignet ist, erfolgt eine Approximation durch eine Funktion, die diese Anforderungen erfüllt, beispielsweise:

$$\hat{I}_-(u) = -(1/t)\log(-u)\,.$$

Darin bestimmt der Barriere-Parameter $t > 0$ die Genauigkeit der Approximation von $\hat{I}_-(u)$ an die Indikatorfunktion. Das Barriere-Problem ist dann formulierbar als[51]:

$$\text{BP} \qquad \text{minimize} \qquad t f_0(x) - \sum_{i=1}^{m} \log(-f_i(x))$$

$$\text{subject to} \qquad Ax = b\,.$$

Die enthaltene Funktion

$$I_{\log}(x) = -\sum_{i=1}^{m} \log(-f_i(x))$$

wird als *logarithmische Barriere* bezeichnet. Offensichtlich sorgt diese Funktion dafür, dass bei einer Folge von Punkten innerhalb der zulässigen Menge für Punkte in der Nähe des Randes eine Barriere aufgebaut wird.

Das Problem BP ist durch ein Newton-Verfahren für Probleme mit Gleichheitsnebenbedingungen (auf die Beschreibung dieses Newton-Verfahrens wird in dieser Arbeit verzichtet) für jeden Parameter $t > 0$ eindeutig lösbar und damit $\bar{x}(t)$ als optimale Lösung in Abhängigkeit von t beschreibbar. Die Menge der Punkte $\bar{x}(t)$ für $t > 0$ wird als *zentraler Pfad* bezeichnet. Für wachsende t werden die Minima des Barriere-Problems immer stärker von $f_0(x)$ bestimmt und konvergieren entlang des zentralen Pfades gegen ein Minimum der ursprünglichen Zielfunktion.

Es lässt sich zeigen, dass für jeden Punkt $\bar{x}(t)$ des zentralen Pfades ein dual zulässiges Paar $\bar{\lambda}(t)$, $\bar{\nu}(t)$ existiert. Mit der Anzahl m der Ungleichungsnebenbedingungen gilt dann für die Dualitätslücke zwischen primaler und dualer Funktion

$$f_0(\bar{x}(t)) - g(\bar{\lambda}(t), \bar{\nu}(t)) = m/t\,.$$

Für größer werdende t verringert sich offensichtlich die Dualitätslücke, für $t \to \infty$ konvergiert $\bar{x}(t)$ gegen ein Optimum.

Das eigentliche Barriere-Verfahren löst eine Folge linear restringierter Probleme, als Abbruchkriterium dient die Dualitätslücke. Der gesamte Ablauf ist in Algorithmus 6-3 dargestellt.

Im Verlauf des Verfahrens wird in jeder Iteration, ausgehend von einem Punkt $\bar{x}(t)$ des zentralen Pfades, für ein $t > 0$ ein neuer Punkt auf dem zentralen Pfad berechnet und t erhöht. Das Verfahren generiert somit eine Folge von Punkten des zentralen Pfades. Es wird dabei zwischen einer äußeren und einer inneren Iteration unterschieden. Die äußere Iteration (alle Schritte innerhalb „wiederhole“) folgt dem zentralen Pfad, die innere Iteration erfolgt im

[51] Falls das ursprüngliche Problem keine Gleichheitsnebenbedingungen enthält, ist selbstverständlich auch das Problem BP unrestringiert.

Centering Step, indem durch das Newton-Verfahren der jeweilige Punkt des zentralen Pfades bestimmt wird.

Gegeben: *Strikt zulässiger Startpunkt* $x^{(0)}$, *Barriere-Parameter* $t^{(0)} > 0$, *Konstante* $\mu > 1$ *und Abbruchtoleranz* $\epsilon > 0$

Gesucht: *Punkt des zentralen Pfades, der* $tf_0(x) - \sum_{i=1}^{m} log(-f_i(x))$ *unter der Nebenbedingung* $Ax = b$ *minimiert.*

Setze $\mathsf{t} = \mathsf{t}^{(0)}$, $\mathsf{k} = 0$;

do

Centering Step: Berechne $\bar{\mathsf{x}}(\mathsf{t})$ *durch Minimierung von* $\mathsf{tf}_0(\mathsf{x}) - \sum_{\mathsf{i}=1}^{\mathsf{m}} \log(-\mathsf{f}_i(\mathsf{x}))$ *unter der Nebenbedingung* $\mathsf{Ax} = \mathsf{b}$, *Startpunkt ist* $\mathsf{x}^{(\mathsf{k})}$;

Setze $\mathsf{x}^{(\mathsf{k}+1)} = \bar{\mathsf{x}}(\mathsf{t})$;

Setze $\mathsf{k} = \mathsf{k} + 1$;

Erhöhe t: $\mathsf{t} = \mu\mathsf{t}$;

while $\mathsf{m}/\mathsf{t} > \epsilon$;

Algorithmus 6-3: Barriere-Verfahren (nach Boyd & Vandenberghe 2004)

Das Iterationsverhalten des Verfahrens wird maßgeblich durch die Wahl der Parameter μ und $t^{(0)}$ bestimmt. Der Parameter μ legt die Balance zwischen inneren und äußeren Iterationen fest: Kleine μ erhöhen t in jeder äußeren Iteration nur geringfügig, damit ist ein Punkt $x^{(k)}$ eine guter Startpunkt für das Newton-Verfahren im nachfolgenden Durchlauf. Insgesamt führt dies zu einer geringen Anzahl innerer aber zu einer großen Anzahl äußerer Iterationen. Für große μ gilt der umgekehrte Fall. Für ein zu groß gewähltes $t^{(0)}$ benötigt die erste Ausführung des Centering Steps eine größere Anzahl Iterationen, für ein zu klein gewähltes $t^{(0)}$ erhöht sich mindestens die Anzahl der äußeren Iterationen, unter Umständen auch die des ersten Centering Steps.

6.5.3.2 Sequentielle quadratische Programmierung

Verfahren der Sequentiellen quadratischen Programmierung (SQP-Verfahren) gelten als sehr leistungsfähig für die Lösung allgemeiner nichtlinearer Probleme (Hock & Schittkowski 1983, vgl. Geiger & Kanzow 2002), allerdings darf die Anzahl der Variablen und Nebenbedingungen nicht zu groß sein (vgl. Schittkowski 1985). Die Funktionen „*fmincon*“ und „*fminimax*“ der Matlab Optimization Toolbox nutzen beispielsweise SQP-Verfahren als Standard-Algorithmen für kleine und mittlere Probleme.

Die Lösung eines Problems der Form NLP mit SQP-Verfahren erfolgt durch sukzessive Lösung quadratischer Teilprobleme, die durch Approximation der Lagrange-Funktion und Linearisierung der Restriktionen entstehen. Basis der SQP-Verfahren ist das Lagrange-Newton-Verfahren, eine Anwendung des Newton-Verfahrens zur Bestimmung stationärer Punkte der Lagrange-Funktion $L(x, \lambda, \nu)$ (Fletcher 1987).

Für die folgenden Ausführungen wird von der Form NLP ausgegangen. Die Funktionen $f_i,\ i=1,\ldots,m,$ und $h_j,\ j=1,\ldots,p,$ seien zweimal differenzierbar, $(\bar{x},\bar{\lambda},\bar{\nu})$ sei ein KKT-Punkt des Problems NLP und die constraint qualification sei erfüllt. Weiterhin gelte strikte Komplementarität ($\bar{\lambda}_i + f_i(\bar{x}) \neq 0$ für $i=1,\ldots,m$) und die hinreichende Optimalitätsbedingung zweiter Ordnung sei erfüllt (Jarre & Stoer 2004, Geiger & Kanzow 2002).

Der KKT-Punkt $(\bar{x},\bar{\lambda},\bar{\nu})$ erfüllt dann die Optimalitätsbedingungen erster Ordnung (KKT-Bedingungen) (vgl. Jarre & Stoer 2004):

$$\Phi(\bar{x},\bar{\lambda},\bar{\nu}) = \begin{pmatrix} \nabla f_0(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla f_i(\bar{x}) + \sum_{i=1}^{p} \bar{\nu}_i \nabla h_i(\bar{x}) \\ \bar{\lambda}_1 f_1(\bar{x}) \\ \vdots \\ \bar{\lambda}_m f_m(\bar{x}) \\ h_1(\bar{x}) \\ \vdots \\ h_p(\bar{x}) \end{pmatrix} = 0\,,$$

$\bar{\lambda}_i \geq 0$ und $f_i(\bar{x}) \leq 0$ für $i=1,\ldots,m$.

Für die Jacobimatrix der Funktion Φ gilt (ebd.):

$$\Psi(x,\lambda,\nu,\nabla_{xx}L(x,\lambda,\nu)) :=$$

$$\nabla\Phi(x,\lambda,\nu)^T = \begin{pmatrix} \nabla_{xx}L(x,\lambda,\nu) & \nabla f_1(x) & \ldots & \nabla f_m(x) & \nabla h_1(x) & \ldots & \nabla h_p(x) \\ \lambda_1 \nabla f_1(x)^T & f_1(x) & & & & & \\ \vdots & & \ddots & & & 0 & \\ \lambda_m \nabla f_m(x)^T & & & f_m(x) & & & \\ \nabla h_1(x)^T & & & & & & \\ \vdots & & & 0 & & 0 & \\ \nabla h_p(x)^T & & & & & & \end{pmatrix}.$$

Eine Lösung der KKT-Bedingungen kann durch Anwendung des Newton-Verfahrens gemäß Abschnitt 6.5.2.2 (Formel 6.1) berechnet werden (vgl. auch Jarre & Stoer 2004):

$$\Psi\left(x^{(k)},\lambda^{(k)},\nu^{(k)},\nabla_{xx}L\left(x^{(k)},\lambda^{(k)},\nu^{(k)}\right)\right)\begin{pmatrix} \Delta x^{(k)} \\ \Delta\lambda^{(k)} \\ \Delta\nu^{(k)} \end{pmatrix} = -\Phi\left(x^{(k)},\lambda^{(k)},\nu^{(k)}\right)$$

mit

$$\Delta x^{(k)} = x^{(k+1)} - x^{(k)}, \qquad \Delta\lambda^{(k)} = \lambda^{(k+1)} - \lambda^{(k)}, \qquad \Delta\nu^{(k)} = \nu^{(k+1)} - \nu^{(k)}.$$

Für das SQP-Verfahren wird nun das Newton-Verfahren wie folgt formuliert (vgl. Jarre & Stoer 2004):

$$\Psi\left(x^{(k)},\lambda^{(k+1)},\nu^{(k+1)},B^{(k)}\right)\begin{pmatrix}\Delta x^{(k)}\\ \Delta\lambda^{(k)}\\ \Delta\nu^{(k)}\end{pmatrix}=-\Phi\left(x^{(k)},\lambda^{(k)},\nu^{(k)}\right) \tag{6.2}$$

mit den zusätzlichen Forderungen

$$\lambda_i^{(k+1)}\geq 0 \quad \text{für } i=1,\ldots,m \text{ und} \tag{6.3}$$

$$f_i\left(x^{(k)}\right)+\nabla f_i\left(x^{(k)}\right)^T\Delta x^{(k)}\leq 0 \text{ für } i=1,\ldots,m. \tag{6.4}$$

Im Ansatz (6.2) ist die relativ aufwändig zu berechnende Hesse-Matrix $\nabla_{xx}L\left(x^{(k)},\lambda^{(k)},\nu^{(k)}\right)$ durch eine symmetrische Matrix $B^{(k)}$ ersetzt. Das Ersetzen der Vektoren $\lambda^{(k)},\nu^{(k)}$ durch $\lambda^{(k+1)},\nu^{(k+1)}$ wird durch Ausmultiplizieren des Gleichungssystems 6.2 deutlich (vgl. Jarre & Stoer 2004):

$$\begin{aligned}B^{(k)}\Delta x^{(k)}+\left(\nabla f_1\left(x^{(k)}\right)\ \ldots\ \nabla f_m\left(x^{(k)}\right)\right)\Delta\lambda^{(k)}+\left(\nabla h_1\left(x^{(k)}\right)\ \ldots\ \nabla h_p\left(x^{(k)}\right)\right)\Delta\nu^{(k)}=\\ \nabla f_0\left(x^{(k)}\right)-\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i\left(x^{(k)}\right)-\sum_{i=1}^{p}\nu_i^{(k)}\nabla h_i\left(x^{(k)}\right)\\ \left(\lambda_i^{(k)}+\Delta\lambda_i^{(k)}\right)\nabla f_i\left(x^{(k)}\right)^T\Delta x^{(k)}+f_i\left(x^{(k)}\right)\Delta\lambda_i^{(k)}=-\lambda_i^{(k)}f_i\left(x^{(k)}\right),\quad i=1,\ldots,m\\ \nabla h_i\left(x^{(k)}\right)^T\Delta x^{(k)}=-h_i\left(x^{(k)}\right),\quad i=1,\ldots,p\end{aligned}$$

bzw.

$$\begin{aligned}\nabla f_0\left(x^{(k)}\right)+B^{(k)}\Delta x^{(k)}+\sum_{i=1}^{m}\lambda_i^{(k+1)}\nabla f_i\left(x^{(k)}\right)+\sum_{i=1}^{p}\nu_i^{(k+1)}\nabla h_i\left(x^{(k)}\right)=0\\ \lambda_i^{(k+1)}\left(f_i\left(x^{(k)}\right)+\nabla f_i\left(x^{(k)}\right)^T\Delta x^{(k)}\right)=0,\quad i=1,\ldots,m\\ h_i\left(x^{(k)}\right)+\nabla h_i\left(x^{(k)}\right)^T\Delta x^{(k)}=0,\quad i=1,\ldots,p.\end{aligned} \tag{6.5}$$

Die Gleichungen (6.3), (6.4) und (6.5) sind die KKT-Bedingungen eines quadratischen Minimierungsproblems (vgl. Newton-Verfahren im Abschnitt 6.5.2.2, Jarre & Stoer 2004):

$$\begin{aligned}&\text{minimize} && \nabla f\left(x^{(k)}\right)^T\Delta x+\frac{1}{2}\Delta x^T B^{(k)}\Delta x\\ &\text{subject to} && f_i\left(x^{(k)}\right)+\nabla f_i\left(x^{(k)}\right)^T\Delta x\leq 0 && i=1,\ldots,m\\ & && h_i\left(x^{(k)}\right)+\nabla h_i\left(x^{(k)}\right)^T\Delta x=0 && i=1,\ldots,p.\end{aligned}$$

Dieses Problem wird jeweils in den Iterationen des SQP-Verfahrens gelöst. Den Ablauf des Verfahrens skizziert Algorithmus 6-4.

Gemäß der Ausführungen zum Newton-Verfahren konvergiert darin die Folge $\left\{\left(x^{(k)},\lambda^{(k)},\nu^{(k)}\right)\right\}$ lokal superlinear gegen $\left(\bar{x},\bar{\lambda},\bar{\nu}\right)$, im Falle der lokalen Lipschitz-Stetigkeit

von $\nabla^2 f_0$, $\nabla^2 f_i$, $i = 1,\ldots,m$, und $\nabla^2 h_j$, $j = 1,\ldots,p$, sogar quadratisch (Geiger & Kanzow 2002).

Gegeben: *Startpunkt* $(x^{(0)}, \lambda^{(0)}, \nu^{(0)}) \in \Re^n \times \Re^m \times \Re^p$, *symmetrische Matrix* $B^{(0)} = (B^{(0)})^T \approx \nabla_{xx} L(x^{(0)}, \lambda^{(0)}, \nu^{(0)})$

Gesucht: *Punkt, der die KKT-Bedingungen des Problems NLP erfüllt*

Setze $\mathsf{k} = 0$*;*

do

Berechne eine Lösung $\Delta\mathsf{x}^{(\mathsf{k})} \in \Re^{\mathsf{n}}$ *des quadratischen Teilproblems*

$$\begin{aligned} &\textit{minimize} && \nabla \mathsf{f}(\mathsf{x}^{(\mathsf{k})})^{\mathsf{T}} \Delta\mathsf{x} + \tfrac{1}{2}\Delta\mathsf{x}^{\mathsf{T}} \mathsf{B}^{(\mathsf{k})} \Delta\mathsf{x} \\ &\textit{subject to} && \mathsf{f_i}(\mathsf{x}^{(\mathsf{k})}) + \nabla \mathsf{f_i}(\mathsf{x}^{(\mathsf{k})})^{\mathsf{T}} \Delta\mathsf{x} \leq 0, \quad \mathsf{i} = 1, \ldots, \mathsf{m}, \\ &&& \mathsf{h_i}(\mathsf{x}^{(\mathsf{k})}) + \nabla \mathsf{h_i}(\mathsf{x}^{(\mathsf{k})})^{\mathsf{T}} \Delta\mathsf{x} = 0, \quad \mathsf{i} = 1, \ldots, \mathsf{p} \end{aligned}$$

mit den Lagrange-Multiplikatoren $\lambda^{(\mathsf{k}+1)}$ *und* $\nu^{(\mathsf{k}+1)}$*;*

Setze $\mathsf{x}^{(\mathsf{k}+1)} = \mathsf{x}^{(\mathsf{k})} + \Delta\mathsf{x}^{(\mathsf{k})}$*;*

Bestimme $\mathsf{B}^{(\mathsf{k})} = (\mathsf{B}^{(\mathsf{k})})^{\mathsf{T}}$*;*

Setze $\mathsf{k} = \mathsf{k} + 1$*;*

while $(\mathsf{x}^{(\mathsf{k})}, \lambda^{(\mathsf{k})}, \nu^{(\mathsf{k})})$ *kein KKT-Punkt ist;*

Algorithmus 6-4: SQP-Verfahren (nach Geiger & Kanzow 2002)

Das SQP-Verfahren ist durch Einführung einer Schrittweitenstrategie globalisierbar, d.h. es wir ein global konvergentes Verfahren erhalten. Für das Konvergenzverhalten des Verfahrens ist auch die Wahl der Matrizen $B^{(k)}$ von Bedeutung. Vertiefende Ausführungen zur Wahl der Schrittweite und $B^{(k)}$ finden sich u.a. in Han (Han 1977), Geiger & Kanzow (Geiger & Kanzow 2002) oder Jarre & Stoer (Jarre & Stoer 2004).

6.5.4 Lokale Suchverfahren

Die Lösung von Optimierungsproblemen kann auch durch heuristische Verfahren erfolgen. Im Folgenden werden die Simulated Annealing Search und die Local Beam Search vorgestellt, die beide ihren Ursprung in der Künstlichen Intelligenz haben. Die Beschreibungen dieses Abschnitts sind entnommen aus Russell & Norvig (2003).

6.5.4.1 Simulated Annealing Search

Mit der Simulated Annealing Search wird der schrittweise Abkühlungsvorgang von glühendem Glas oder Metall simuliert. Dazu nutzt der Algorithmus – ähnlich wie die Abstiegsverfahren – von einem Startpunkt aus eine Suchrichtung, die allerdings in diesem Fall zufällig bestimmt wird. Verbessert eine Suchrichtung die Zielfunktion, wird in dieser Richtung vorgegangen, andernfalls wird diese Richtung lediglich mit einer Wahrscheinlichkeit, die kleiner als 1 ist, akzeptiert. Diese Wahrscheinlichkeit hängt sowohl von der Suchrichtung, als auch vom Fortschritt des Algorithmus (der Temperatur beim Abkühlungsvorgang) ab: Schlechtere

Suchrichtungen verringern die Wahrscheinlichkeit ebenso wie eine niedrige Temperatur. Die einzelnen Schritte des Verfahrens gibt Algorithmus 6-5 an.

Procedure SimulatedAnnealing(Problem, Zeitplan)
  Gegeben: *Problem := Ein Optimierungsproblem*
  Zeitplan := Mapping der Zeit auf die Temperatur
  Gesucht: *Ein Minimum von Problem*
  Initialize: *Weise* Aktuell *einen Zustand von* Problem *zu;*
  for t = 1...∞
    Temperatur = Zeitplan[t];
    if Temperatur = 0
      return Aktuell;
    end
    Naechster *sei ein zufällig gewählter Nachfolger von* Aktuell;
    ΔE = *Wert(*Aktuell*)* - *Wert(*Naechster*);*
    if ΔE > 0
      Aktuell = Naechster;
    else
      Aktuell = Naechster *mit Wahrscheinlichkeit* $e^{\Delta E/T}$*;*
    end
  end
end

Algorithmus 6-5: Simulated Annealing (nach Russell & Norvig 2003)

6.5.4.2 Local Beam Search

Die Local Beam Search bezeichnet ebenfalls ein randomisiertes lokales Verfahren. Der Algorithmus beginnt mit der zufälligen Initialisierung von *k* Zuständen und berechnet alle Nachfolger dieser Zustände. Im nächsten Schritt werden die *k* besten dieser Nachfolger als Startpunkte ausgewählt und wiederum deren Nachfolger berechnet. Diese Iterationen erfolgen solange, bis das gewünschte Ziel erreicht ist. Durch die Nutzung der parallelen Suchthreads und die Übernahme der jeweils besten Nachfolgezustände bewegt sich der Algorithmus sehr schnell in die Richtung, die die besten Ergebnisse verspricht (Best-first Search).

6.6 Globale Verfahren

Nach der Beschreibung ausgewählter lokaler Verfahren werden in diesem Abschnitt mit dem Simplex- und dem Branch-and-Bound-Verfahren noch zwei Methoden vorgestellt, die die Bestimmung einer global optimalen Lösung zum Ziel haben. Wie die Einbeziehung des Simplex deutlich macht, wird der Begriff des globalen Verfahrens hier nicht im Sinne der globalen Optimierung nichtlinearer Programme gebraucht, sondern bezeichnet allgemein alle Methoden, die eine global optimale Lösung berechnen.

6.6.1 Simplex-Verfahren

Der klassische Algorithmus zur Lösung linearer Programme ist das Simplex-Verfahren, das allerdings in jüngerer Zeit Konkurrenz durch Innere-Punkte-Verfahren bekommen hat. Welches dieser Verfahren effizienter ist, ist bislang nicht endgültig geklärt (Jarre & Stoer 2004). Das Simplex-Verfahren ist nicht direkt auf das Farbproblem anwendbar, wird aber in dieser Arbeit als Paradigma für Methoden benötigt, die eine Lösung durch systematisches Absuchen des Randes der zulässigen Menge finden.

Lineare Programme wurden bereits im Abschnitt 6.2.3.2 in der Notation des allgemeinen Problems SFP als Minimierungsprobleme eingeführt. Literatur mit Fokus auf der linearen Programmierung (z.B. Vanderbei 1996, Chvátal 1983) definiert die Standardform für lineare Programme durch eine zu maximierende Zielfunktion, lineare Ungleichungsnebenbedingungen und nicht-negative Entscheidungsvariablen:

$$\begin{array}{lll} \text{LP} & \text{maximize} & c^T x \\ & \text{subject to} & Ax \le b \\ & & x \ge 0 \end{array}$$

Falls die durch die Ungleichungsnebenbedingungen definierte zulässige Menge nicht leer ist, bildet sie ein konvexes Polyeder. Alle Optimalpunkte liegen auf dem Rand dieses Polyeders. Existiert ein eindeutiger Optimalpunkt, liegt dieser in einem Eckpunkt. Da jedes lineare Programm konvex ist (vgl. Abschnitt 6.2.3.2), ist die optimale Lösung immer global optimal.

Abbildung 6-16 zeigt links die geometrische Interpretation eines linearen Programms in der Ebene. Die zulässige Menge (blaues Polygon) wird durch Geraden(gleichungen) begrenzt, die durch die angenommene Gleichheit der Ungleichungsnebenbedingungen entstehen. Die Zielfunktion ist ebenfalls als Gerade gezeichnet und wird parallel verschoben, bis der „maximale“ Eckpunkt des Polyeders erreicht ist.

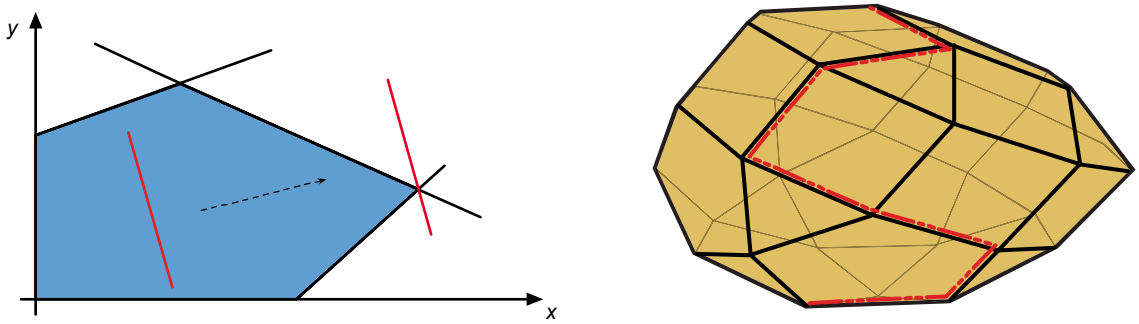


Abbildung 6-16: Graphische Darstellung eines linearen Programms in der Ebene (links); Fortschreiten des Simplex-Algorithmus auf einem konvexen Polyeder im dreidimensionalen Raum (rechts)

Für die Beschreibung des Simplex-Verfahrens sei ein lineares Optimierungsproblem in Standardform gegeben, die zulässige Menge sei nichtleer. Das Simplex-Verfahren findet die optimale Lösung, indem es sich von einer Startlösung (Eckpunkt des Polyeders) entlang der Kan-

ten des Polyeders so lange von Eckpunkt zu Eckpunkt bewegt (Abbildung 6-16, rechts), bis der Wert der Zielfunktion nicht mehr weiter verbessert wird (vgl. Winston 1991).

Das Vorgehen von Eckpunkt zu Eckpunkt des Polyeders wird durch die *Simplexmethode* umgesetzt, indem sukzessive lineare Gleichungssysteme gelöst werden. Dazu werden zunächst die Ungleichungen der oben dargestellten Standardform durch Einführung so genannter *Schlupfvariablen* $x_{n+1},\ldots,x_{n+m}$ (bei n Variablen und m Gleichungen) in Gleichungen umgewandelt (Vanderbei 1996):

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1,\ldots,m.$$

Es ergibt sich dann folgendes Problem:

$$\begin{aligned} &\text{maximize} && c^T x \\ &\text{subject to} && Ax = b, \\ & && x \geq 0 \end{aligned}$$

mit

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} & 1 & 0 & \ldots & 0 \\ a_{21} & a_{22} & \ldots & a_{2n} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \\ a_{m1} & a_{m2} & \ldots & a_{mn} & 0 & 0 & \ldots & 1 \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{pmatrix}.$$

Das lineare Gleichungssystem $Ax = b$ besteht aus m Gleichungen mit $n+m$ Variablen, es gelte $n > m$. Für die Simplexmethode werden m Variablen als *Basisvariablen* x_B (Basisvektor) gewählt, die übrigen als *Nichtbasisvariablen* x_N. Das Gleichungssystem ist dann durch Aufteilung der Matrix A darstellbar als (Vanderbei 1996):

$$Ax = \begin{bmatrix} B & N \end{bmatrix} \cdot \begin{bmatrix} x_B \\ x_N \end{bmatrix} = Bx_B + Nx_N = b. \tag{6.6}$$

Für die Zielfunktion gilt (ebd.):

$$\zeta := c^T x = \begin{bmatrix} c_B \\ c_N \end{bmatrix}^T \cdot \begin{bmatrix} x_B \\ x_N \end{bmatrix} = c_B^T x_B + c_N^T x_N. \tag{6.7}$$

Aus $Bx_B + Nx_N = b$ in Formel 6.6 folgt

$$x_B = B^{-1}b - B^{-1}Nx_N. \tag{6.8}$$

Eine *Basislösung* wird erhalten, indem die frei wählbaren Nichtbasisvariablen Null gesetzt werden:

$$x_N = 0 \qquad \Rightarrow \quad x_B = B^{-1}b.$$

Die Zielfunktion ζ besteht dann aus einem numerischen Wert, der sich durch $c_B^T x_B$ aus der Basislösung berechnet, und dem Ausdruck $c_N^T x_N$, der die Nichtbasisvariablen enthält (vgl. Formel 6.7).

Eine derartige Basislösung erfolgt in jeder Iteration des Simplex-Verfahrens (in jedem Eckpunkt des Polyeders). Die Bewegung entlang der Kanten (Simplexschritt) schreitet voran, indem eine Basisvariable (*Entering Variable*) gegen eine Nichtbasisvariable (*Leaving Variable*) derart ausgetauscht wird, dass sich der Wert der Zielfunktion vergrößert (Winston 1991). Ist der optimale Eckpunkt des Polyeders erreicht, ist das Optimum durch $c_B^T x_B$ gegeben.

Das Simplex-Verfahren fasst Algorithmus 6-6 zusammen.

Gegeben: *Zielfunktion ζ, Variablen des Basisvektors $x_B^{(0)}$, zugehörige Matrix $B^{(0)}$, Vektor $b^{(0)}$*

Gesucht: *Lösung eines linearen Maximierungsproblems*

Setze $\mathsf{k} = 0$*;*

do

Berechne die Basisvariablen: $\mathsf{x}_\mathsf{B}^{(\mathsf{k})} = (\mathsf{B}^{(\mathsf{k})})^{-1}\mathsf{b}^{(\mathsf{k})}$*;*

if *Alle Variablen in ζ haben nicht-positive Koeffizienten*

break; *//Das Optimum ist erreicht*

else

Wähle diejenige Variable in ζ, die den größten positiven Koeffizienten besitzt, als Entering Variable*;*

while *Alle aktuellen Basisvariablen sind nicht-negativ*

Erhöhe den Wert von Entering Variable*;*

end

Wähle diejenige Basisvaribale, die bei Erhöhung der Entering Variable *zuerst negativ wird, als* Leaving Variable*;*

end

while *Es gibt Variablen in ζ mit positiven Koeffizienten;*

Algorithmus 6-6: Simplex-Verfahren (nach Winston 1991, Vanderbei 1996)

An dieser Stelle sei erwähnt, dass bei Wahl einer anderen Abstandsfunktion das Farbproblem bzw. allgemein Distanzprobleme als lineare Programme lösbar sind. Offensichtlich gilt Linearität für die Manhattan-Norm ($L_1 - Norm$) für $x > 0$ und $x < 0$ (Anhang A.4.3). Eine mögliche Anwendung dieser Norm auf ein Distanzproblem beschreibt beispielsweise Melachrinoudis (1988) anhand der Platzierung einer „Obnoxious Facility“, indem er das Problem so formuliert, dass es durch eine Sequenz linearer Programme unter Verwendung des Simplex-

Verfahrens effizient lösbar ist. Allerdings liegt der Nutzung der Manhattan-Norm in diesem – und anderen Beispielen (vgl. auch Modelle der Facility Location, Abschnitt 6.3.2.2) – die Annahme zugrunde, dass Distanzen entlang eines annähernd rechtwinkligen Netzwerks gemessen werden. Diese Annahme ist für das Farbproblem offensichtlich nicht akzeptabel.

6.6.2 Branch-and-Bound-Verfahren

Das verbreitete Vorgehen der Mathematischen Programmierung zur Lösung diskreter und kombinatorischer Probleme ist die Anwendung einer systematischen Suchstrategie durch das Branch-and-Bound-Verfahren. Dieses stellt auch gewissermaßen eine Brücke von der Mathematischen Optimierung zum *Constraint Programming* (CP) dar. CP bezeichnet eine, im Vergleich zur Mathematischen Programmierung, recht junge Disziplin, die sich ebenfalls mit der Lösung von Optimierungsproblemen beschäftigt. Die Wurzeln liegen in der künstlichen Intelligenz und der logischen Programmierung (Apt 2003). Anders als in der Mathematischen Programmierung bezieht sich im Constraint Programming der Begriff Programming tatsächlich auf ein Computerprogramm zur Lösung eines bestimmten Problems (Lustig & Puget 2001, dort auch ausführlicheres zur Beziehung von Mathematischer Programmierung und Constraint Programming). Allerdings liegt dem Constraint Programming ein sehr viel allgemeineres Konzept zugrunde: Für eine Menge von Variablen, deren Domain nicht zwingend aus numerischen Elementen bestehen muss, sondern jegliche Art von Entitäten enthalten darf (vgl. Apt 2003), werden zunächst einschränkende (Neben)bedingungen (constraints) formuliert. Die Implementierung der Variablen und Bedingungen in ein Computerprogramm ermöglicht dann die Bestimmung von Werten für diese Variablen (Lustig & Puget 2001). In der Interpretation der Bedingungen liegt ein weiterer Unterschied zwischen Mathematischer Programmierung und Constraint Programming (Hooker 2006): Erstere fasst die Nebenbedingungen immer in der Gesamtschau als Festlegung der zulässigen Menge auf. Das Constraint Programming betrachtet dagegen auch die einzelnen Constraints und versteht diese als Verfahren, die auf den Lösungsraum angewandt werden und die Domains von Variablen reduzieren.

Das Constraint Programming geht zudem nicht zwingend von einem Optimierungsproblem aus, sondern zunächst von einem „*Constraint Satisfaction Problem*“. Für ein solches Problem genügt es, dass die Werte der Variablen die Constraints erfüllen (In der Terminologie der Mathematischen Optimierung: Es werden Punkte der zulässigen Menge gesucht). Liegt darüber hinaus eine zu maximierende oder minimierende Zielfunktion vor, handelt es sich um ein „C*onstraint Optimization Problem*“, für das eine optimale Lösung gesucht wird (Apt 2003). Die formale Terminologie des Constraint Programming ist ähnlich zu der der Mathematischen Programmierung, wird aber im Folgenden nicht weiter benötigt.

Aus Sicht des Constraint Programming lässt sich das Branch-and-Bound-Verfahren als eine Erweiterung der *Backtracking-Suche* interpretieren. Letztere ist neben lokalen Suchverfahren (z.B. Simulated Annealing, Abschnitt 6.5.4.1) und dem *Dynamic Programming*[52] eines der

[52] Dynamic Programming nutzt rekursive oder verschachtelte Strukturen zur Lösung eines Problems (Hooker 2006).

drei Hauptverfahren des Constraint Programming (Van Beek 2006). Das Backtracking löst ein Constraint Satisfaction Problem durch systematisches Durchsuchen des gesamten Lösungsraums, d.h. aller möglichen Kombinationen von Lösungen. Die Suche erfolgt, indem durch gezielte Initialisierung aller Variablen parallel ein Suchbaum aufgebaut und durchsucht wird (Apt 2003). Dafür wird in jedem Knoten des Baumes eine bis dahin nicht initialisierte Variable ausgewählt und dieser Variablen Werte ihrer Domain zugewiesen. Diese Zuweisungen stellen dann die Äste dar, die vom gerade betrachteten Knoten ausgehen. Die Äste repräsentieren damit alle Werte, die eine Variable annehmen kann. In jedem Knoten werden die Constraints überprüft, die keine uninstanziierte Variable mehr enthalten. Schlägt eine Überprüfung fehl, wird mit dem nächsten Wert der Domain fortgefahren. Wenn alle Werte der Domain geprüft wurden, geht der Algorithmus zum Elterknoten zurück (backtrack). Ein Knoten, der nicht zu einer Lösung führt, wird als Sackgasse (deadend) bezeichnet. Eine Lösung liegt dagegen vor, wenn nach der zuletzt instanziierten Variablen in einem Knoten alle Constraints erfüllt sind. Das Constraint Satisfaction Problem ist dann *konsistent* (Apt 2003). Dieses skizzierte Vorgehen wird von Van Beek (2006) als naives Backtracking beschrieben. Um die Effizienz der Suche zu erhöhen, kann dieser Algorithmus um verschiedene Techniken – allein oder in verschiedenen Kombinationen – erweitert werden (Van Beek 2006):

- *Branching Strategien* bezeichnen die Möglichkeiten, wie ein Knoten um Äste erweitert wird, d.h. wie eine Variable instanziiert wird. Drei häufig genutzte Vorgehensweisen sind:
  - *Aufzählung*: Eine Variable wird in der Reihenfolge der Werte ihrer Domain instanziiert.
  - *Binäre Wahl*: Diese Strategie baut einen Binärbaum auf, indem eine Variable x mit einem Wert d ihrer Domain durch $x = d$ und $x \neq d$ belegt wird.
  - *Domain Splitting*: Eine Variable wird nicht zwingend instanziiert, sondern es wird in jedem betrachteten Unterproblem die Domain reduziert. Bei einer geordneten Domain kann dies beispielsweise für eine Variable x und einen Wert d durch $x \leq d$ und $x > d$ erfolgen.
- *Constraint Propagation* verkleinert den Suchbaum vor und während des Aufbaus, indem das Ausgangsproblem durch ein äquivalentes aber einfacheres Problem ersetzt wird. Dies geschieht typischerweise dadurch, dass die Domains der Variablen um diejenigen Werte reduziert werden, die von vorneherein nicht Teil einer Lösung des Problems sein können (Apt 2003). Eine Möglichkeit der Constraint Propagation ist das Forward Checking, durch das „Arc Consistency“[53] für alle Constraints, die genau eine uninitialisierte Variable enthalten, erreicht wird (Van Beek 2006).

[53] Arc Consistency ist eine Form lokaler Konsistenz. Lokale Konsistenz bedeutet allgemein, dass ein Teilproblem eines Constraint Satisfaction Problems bestimmte Bedingungen erfüllt, beispielsweise konsistent ist (vgl. Apt 2003). Ein binärer Constraint (Constraint auf zwei Variablen) ist arc consistent, wenn jeder Wert der Domains beider Variablen Teil einer Lösung ist (ebd.).

- *Nogood recording* fügt einem Problem redundante (implizite) Constraints hinzu, d.h. Constraints, die keinen Einfluss auf die Lösungsmenge eines Problems haben. Als Nogood wird dabei eine Menge von Zuweisungen und Constraints verstanden, die keine Lösung ergeben. Im Verlauf einer Backtracking-Suche entspricht jede Sackgasse einem Nogood. Der dort gültige Zustand des Algorithmus kann dann als zusätzliches (vereinfachendes) Constraint eingeführt werden. Dieses Constraint hat allerdings keinen Einfluss mehr auf die aktuell erreichte Sackgasse, sondern *kann* die Suche im weiteren Verlauf in anderen Bereichen des Baums erleichtern, indem sich dadurch die Betrachtung von Teilbäumen mit gleichen Konstellationen, die zur Einführung des Nogoods geführt haben, erübrigt.
- *Nicht-chronologisches Backtracking*: Im oben beschriebenen naiven Backtracking ging der Algorithmus jeweils zum Elternknoten zurück. Dieses Verhalten wird auch als chronologisches Backtracking bezeichnet. Eine andere Strategie ist das Backjumping, das das Zurückweisen eines Astes des Suchbaums möglichst dort bezeichnet, wo der Ursprung der Sackgasse liegt, d.h. es wird nicht zum letzten Knoten zurückgegangen, sondern über mehrere Elternknoten zurückgesprungen. Das Zurückspringen kann jeweils bei Feststellung eines Nogoods erfolgen.
- *Heuristiken*: Aus der Beschreibung des naiven Backtracking ging bereits hervor, dass beim Aufbau des Suchbaums in jedem Knoten Entscheidungen zu treffen sind. Diese bestehen in der Festlegung, welche Variable als nächste initialisiert wird und mit welchem Wert diese Initialisierung erfolgt. Diese Entscheidungen können durch Heuristiken getroffen werden. Beispielsweise kann die nächste zu instanziierende Variable in Abhängigkeit von der Größe der Domain der verbleibenden Variablen gewählt werden.
- Durch *Randomisierung und Restart-Strategien* lassen sich verschiedene Abläufe einer Backtracking-Suche erreichen und so Fehler, die durch Heuristiken entstehen, vermeiden. Die Randomisierung wird auf Entscheidungen im Verlauf der Suche angewandt, beispielsweise auf die angesprochen Reihenfolge der Variableninitialisierung und die Frage, mit welchem Wert die Initialisierung erfolgt. Wird bei einer solchen zufallsbasierten Suche nicht nach t Schritten eine Lösung erreicht, erfolgt ein Neustart des Algorithmus und damit ein anderer – wiederum zufallsbasierter – Aufbau des Suchbaums.
- *Suchstrategien* beschreiben, welche Teile des Baumes der Algorithmus bevorzugt besucht:
  - Die *depth-first*-Strategie durchsucht immer zunächst den linken Teilbaum unterhalb eines Knotens und damit den Baum in die Tiefe.
  - Die b*est-first*-Strategie durchsucht zunächst die Teilbäume, die am ehesten eine Lösung erwarten lassen.

Das Branch-and-Bound-Verfahren zur Lösung von Constraint Optimization Problemen erweitert die Backtracking-Suche durch Einführung eines *Boundings*, das ein Unterproblem durch

eine untere und obere Schranke für die Lösung der Zielfunktion evaluiert. Die *untere Schranke* ist die während einer Suche aktuell beste gefundene Lösung; falls noch keine Lösung gefunden wurde, kann die untere Schranke durch eine Heuristik bestimmt werden. Die *obere Schranke* gibt für ein Unterproblem an, wie groß eine größtmögliche Lösung sein kann. Falls dann die aktuelle untere Schranke größer ist als die obere Schranke eines Unterproblems, braucht das gesamte Unterproblem nicht weiter betrachtet werden.

Die Effizienz des Verfahrens wird dann auch durch das Bounding beeinflusst: Die Qualität der jeweiligen Schranken bestimmt, welche Unterprobleme betrachtet werden, d.h. welche Teile des Baums durchsucht werden müssen.

Den Ablauf eines rekursiven Depth-first-Branch-and-Bound skizziert Algorithmus 6-7.

```
Procedure DFBB(t, ub)
    Gegeben: t := Menge von instanziierten Variablen
             ub := obere Schranke für ein Unterproblem
    Gesucht: Untere Schranke lb als Lösung eines Minimierungsproblems
    if Alle Variablen instanziiert
        return Untere Schranke lb(t);
    else
        Wähle eine nicht instanziierte Variable x_i;
        for each Element a der Domain von x_i;
            if lb(t ∪ {(x_i, a)}) ≺_v ub
                ub = min(ub, DFBB(t ∪ {(x_i, a)}, ub));
            end
        end
        return ub;
    end
end
```

Algorithmus 6-7: Rekursiver Depth-first-Branch-and-Bound (nach Meseguer et al. 2006)

Die Methoden des Constraint Programming – insbesondere auch das Branch-and-Bound-Verfahren – können durch Methoden der Mathematischen Programmierung ergänzt werden. Die häufigste Form ist die Formulierung einer sogenannten Lockerung (Relaxation) eines Constraint-Programming-Problems, beispielsweise in Form eines linearen Programms. Diese Lockerung kann auf verschiedene Weise für die Lösung eines Problems genutzt werden. Hooker (2006) nennt u.a.:

- Beschränkung der Domain einer Variablen durch Domain Filtering,
- Bestimmung eines zulässigen Punktes des ursprünglichen Problems durch Lösung des gelockerten Problems,

- Nutzung der Lösung eines gelockerten Problems als Schranken des Branch-and-Bound-Verfahrens.

Für die Formulierung einer Lockerung werden u.a. unterschieden (Hooker 2006):

- *Lagrangsche Relaxation*: Im Abschnitt 6.5.1.2 wurde die Lagrange-Funktion und das durch diese definierte duale Problem eingeführt. Die Bedeutung des dualen Problems wurde durch die Angabe einer unteren Schranke für das primale Problem beschrieben. Diese Schranke kann im Constraint Programming genutzt werden, indem das duale Problem in der Wurzel des Suchbaums gelöst wird und im weiteren Verlauf der Suche als untere Schranke dient. Eine weitere Möglichkeit einer Lagrangschen Relaxation ist das Domain Filtering: Falls eine obere Schranke O für den Wert der Zielfunktion bekannt ist, lassen sich Nebenbedingungen, die eine Variable x_i begrenzen, zusammen mit O und den Lagrange-Multiplikatoren λ_j zur Einschränkung des Wertebereichs von x_i nutzen.

- *LP Relaxation*: Eine LP Relaxation lässt sich ebenfalls in unterschiedlicher Weise zur Lösung eines Problems des Constraint Programming nutzen. Als Sonderfall der Lagrangschen Relaxation für lineare Programme kann darüber ein Domain Filtering formuliert werden. Weiterhin kann durch eine im ursprünglichen Problem nicht zulässige Lösung der LP Relaxation das Branching des Problems bestimmt werden: Für eine Variable x_i, die im ursprünglichen Problem als diskret definiert ist, lässt sich das gelockerte Problem durch Verzicht auf die Forderung der Ganzzahligkeit definieren und eine kontinuierliche Lösung $\tilde{x}_i$ finden. Das Branching im ursprünglichen Problem kann dann durch $x_i \leq \lfloor \tilde{x}_i \rfloor$ und $x_i \geq \lceil \tilde{x}_i \rceil$ erfolgen.
- *Mixed Integer/Linear Programming*: Die Modellierung eines Problems als Mixed Integer/Linear Programming Problem ermöglicht es, in jedem Knoten des Suchbaums die LP Relaxation des Problems zu lösen. Falls der Wert dieser Lösung größer ist als der beste bisher gefundene, geht die Suche im Baum zurück, falls alle Variablenwerte einer Lösung ganzzahlig sind, ist die Lösung Kandidat für eine Lösung des ursprünglichen Problems.
- *Cutting Planes*: Als Cutting Planes werden lineare Ungleichungen bezeichnet, die einem Integer oder Mixed Integer Problem hinzugefügt werden und die zulässige Menge der LP Relaxation eines Problems um nicht-ganzzahlige Werte beschneiden.

Trotz der skizzierten Möglichkeiten, die die Effizienz einer systematischen Suche erhöhen, sind Probleme von der Art und Größe des Farbproblems (zur Lösung kleinerer Problem vgl. Abschnitt 6.3.2) nicht annähernd on demand lösbar.

6.7 Vergleichbare Arbeiten zur Bestimmung von Farben

Abschließend werden in diesem Abschnitt einige Arbeiten betrachtet, die sich ebenfalls mit der Bestimmung kontrastreicher Farben beschäftigt haben. Dabei wurden nicht nur Optimierungsmodelle, sondern auch wissensbasierte Systeme angewandt.

6.7.1 Wissensbasierte Systeme

Chesneau et al. (2005) beschreiben ein wissensbasiertes System aus dem engeren Kontext der Kartographie. Ziel ist es, kontrastreiche Farben zur Darstellung von Risikokarten zu bestimmen. Dafür sollen die Kontraste der Farben von beliebigen Karten eines Nutzers zunächst analysiert, und bei nicht ausreichenden Kontrasten verbessert werden. Die Auswahl der kontrastreichen Farben erfolgt aus einem wissensbasierten System, das feste, vorab bestimmte, Farben eines Farbmodells mit 163 Mustern enthält.

6.7.2 Optimierungsansätze

Ziel der Lösung durch Optimierungsverfahren war ebenfalls immer die Bestimmung gut unterscheidbarer Farben für eine Visualisierung. Besondere Restriktionen, wie beispielsweise bereits im Farbraum platzierte Farben, waren allerdings nicht Teil der Probleme.

Carter & Carter (1982) beschreiben die Bestimmung von Mengen von Farben mit einem möglichst hohen Kontrast zur Darstellung qualitativer Daten. Die Darstellung der Farben soll durch Medien mit additiver Farbmischung erfolgen, die Berechnungen wurden im CIELUV-Farbraum durchgeführt. Carter & Carter nutzen ein MAXMIN-Modell, formulieren dies allerdings nicht mathematisch und stellen auch keinen Zusammenhang zur Mathematischen Optimierung her. Die Lösung erfolgt durch eine iterative Heuristik: In einem ersten Schritt werden n (Anzahl der gesuchten Farben) Punkte des Farbraums zufällig ausgewählt. Anschließend werden iterativ einzelne Punkte so verschoben, dass sich ein verbessertes Ergebnis ergibt.

Campadelli et al. (1999) gehen von der gleichen Fragestellung aus wie Carter & Carter, beschränken den Farbraum allerdings auf Farbpaletten (ISCC-NBS Color Naming System mit 267 Farben, X Windows mit 502 Farben und Munsell Atlas mit 2745 Farben). Die Formulierung erfolgt als Problem der Kombinatorischen Optimierung mit der Modellierung durch einen kompletten, ungerichteten gewichteten Graphen. Die Lösung erfolgt durch eine iterative Heuristik.

Die Fragestellung von Glasbey et al. (2007) geht ebenfalls von der Darstellung von Qualitäten durch Farben aus; Ziel ist wieder die Bestimmung einer Menge von Farben mit hohem Kontrast. Die Autoren betonen, den Ansatz von Carter & Carter im Hinblick auf Rechenleistung und eine breitere Berücksichtigung der Farbmetrik zu betrachten, während mathematische Details nach Campadelli et al. vernachlässigt werden sollen. Die Lösung erfolgt im CIELAB-Farbraum durch die Anwendung eines Simulated Annealing Algorithmus auf eine Menge von n zufällig gewählten Startpunkten.

6.8 Zwischenresümee

In diesem Kapitel wurden die theoretischen Grundlagen für die Lösung des Farbproblems gelegt und mehrere Standardverfahren vorgestellt. Die Verfahren der lokalen Optimierung, die auf das Farbproblem anwendbar sind (Barriere-Verfahren, SQP-Verfahren), ermöglichen eine effiziente Berechnung des Problems. Die Lösung hängt allerdings vom gewählten Start-

punkt ab und ist lokal optimal; von globaler Optimalität kann im Allgemeinen nicht ausgegangen werden. Für die Bestimmung einer global optimalen Lösung durch das Branch-and-Bound-Verfahren ist das Farbproblem NP-vollständig; eine Berechnung on demand ist nicht möglich.

Ein effizientes Verfahren zur Lösung des Farbproblems wird auf Basis von Standardverfahren bzw. deren Lösungsideen im nächsten Kapitel entwickelt.

7 Verfahren zur Identifikation optimaler Farben

Auf Basis der Standardverfahren des letzten Kapitels lässt sich ein Verfahren entwickeln, das für den Fall des Farbproblems eine effiziente Lösung ermöglicht. Kern ist die Nutzung eines lokalen Verfahrens und die Bestimmung einer lokalen Lösung, die der global optimalen Lösung nahe kommt. Um dies zu erreichen wird das lokale Verfahren durch eine Methode zur Bestimmung geeigneter Startpunkte und eine Bewertung des Ergebnisses anhand eines geometrischen Verfahrens durch Berechnung des Voronoi-Diagramms flankiert.

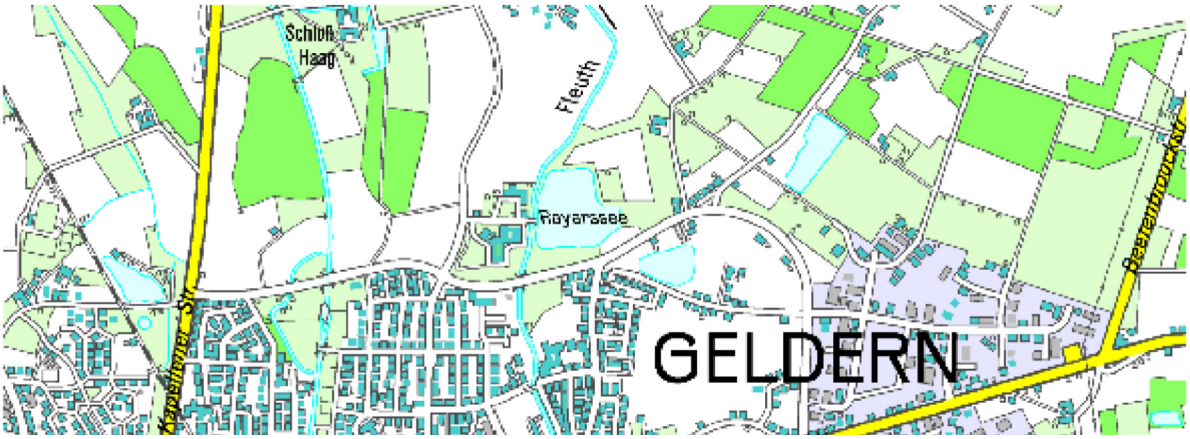
Das in den folgenden Abschnitten vorgestellte Verfahren besteht demnach im Wesentlichen aus drei Schritten:

- Für die Bestimmung von geeigneten Startpunkten für das lokale Verfahren wird eine besondere Eigenschaft des Farbproblems genutzt und angenommen, dass sich Punkte auf dem Rand des Farbraumpolyeders besonders gut als Startpunkte eignen. Diese Punkte werden durch eine Methode erhalten, die die Modellierung des Optimierungsfarbraums als konvexes Polyeder ausnutzt und analog zur Idee des Simplex-Algorithmus Punkte durch systematische Bewegung auf dem Rand des Polyeders sukzessive verbessert. Die erste Ausgangsmenge der Punkte wird durch die Integration randomisierter Elemente nach Art der Beam Search erhalten.
- Durch Einsatz eines lokalen Verfahrens (SQP-Verfahren) werden lokale Lösungen gefunden, die bei geeigneten Startpunkten nahe am globalen Optimum liegen.
- Die Berechnung des Voronoi-Diagramms und die Betrachtung der Dichte von Teilräumen durch Bestimmung der größten leeren Kugel ermöglichen die Bewertung einer lokalen Lösung und eine globale Verbesserung durch sprunghafte Bewegung von Punkten.

Die Eignung dieses Lösungsverfahrens wird anhand zweier konkreter Berechnungen gezeigt. Die erste Berechnung platziert 20 Farben im „leeren“ Farbraum, die zweite geht von einem Kartenbeispiel aus, das die Platzierung von zehn linienhaften Objekten (Themenrouten für Radfahrer) vor einem farbigen Kartenhintergrund zum Ziel hat. Als Kartengrundlage für dieses Beispiel dient der generalisierte Stadtplan des Regionalverbands Ruhr (RVR), abrufbar über einen frei zugänglichen Web Map Service[54]. Einen Ausschnitt des Stadtplans samt der darin enthaltenen Objektklassen und ihrer Repräsentation im RGB-Farbraum zeigt Abbildung 7-1. Die Themenrouten sind Teil eines Radwegenetzes und über einen SLD-fähigen Web Map Service[55] allgemein verfügbar.

[54] http://217.78.131.130:8080/wmsconnector/gdi/wms_spw? (Zuletzt geprüft am: 19.09.08)

[55] http://www.grrad.grenzeloos-fietsen.com/radwege/wms? (Zuletzt geprüft am: 19.09.08)



Flächenhaftes Objekt	Farbe	R	G	B	Linienhaftes Objekt	Farbe	R	G	B
Wald		139	255	98	Hauptverkehrsstraße		255	255	0
Grünfläche		222	255	205	Gewässerlinie		0	255	255
Gewerbefläche		230	230	255	Linie		74	74	74
Gewerbliches Gebäude		189	189	189	(Schrift)		0	0	0
Wohngebäude		74	205	205					
Wasserfläche		222	255	255					

Abbildung 7-1: Beispielkarte; Ausschnitt des generalisierten Stadtplans des Regionalverbands Ruhr (oben); RGB-Werte der darin genutzten Farben (unten)

Abschnitt 7.1 greift zunächst die Verfahren des letzten Kapitels auf und verdeutlicht Möglichkeiten und Grenzen für die Lösung des Farbproblems. Abschnitt 7.2 beschreibt den in dieser Arbeit genutzten Algorithmus, Abschnitt 7.3 vertieft mit der Bestimmung von Startpunkten für das lokale Optimierungsverfahren einen wesentlichen Aspekt dieses Algorithmus. Die Ergebnisse der angesprochenen Farbberechnungen werden im Abschnitt 7.4 angegeben, bevor im Abschnitt 7.5 die Einbeziehung weiterer Rahmenbedingungen in die Farbberechnung skizziert wird. Abschnitt 7.6 bewertet die Ergebnisse dieses Kapitels.

7.1 Lösung des Farbproblems durch Standardverfahren

Im vorangegangenen Kapitel wurden verschiedene etablierte Lösungsverfahren beschrieben, die sich für die Anwendung auf das Farbproblem eignen. Als Methoden der lokalen Optimierung waren dies das Simulated Annealing und das SQP-Verfahren, als globale Methode das Branch-and-Bound-Verfahren. Aus dem Bereich der Algorithmischen Geometrie wurde eine Vorgehensweise skizziert, die auf der Berechnung eines Voronoi-Diagramms basiert. Diese genannten Verfahren sind nun im Hinblick auf eine Anwendung auf das Farbproblem zu analysieren.

Mögliche Lösungen des Farbproblems wurden anhand einfacher Beispiele im Abschnitt 6.3.3 aufgezeigt. Allgemein lassen sich danach drei Fälle unterscheiden:

- Das Problem wird global optimal gelöst.
- Es wird ein lokales Optimum erreicht, das einem globalen Optimum nahe kommt. Dieses Ergebnis wird im Folgenden als „gutes“ lokales Optimum bezeichnet.

- Es wird ein lokales Optimum erreicht, dessen Wert nicht in der Nähe eines globalen Optimums liegt. Ein solches Ergebnis wird im Folgenden als „ungünstiges“ lokales Optimum bezeichnet.

Abbildung 7-2 fasst diese drei Fälle, die auch schon im Abschnitt 6.3.3 illustriert wurden, noch einmal für fünf Punkte in der Ebene zusammen.

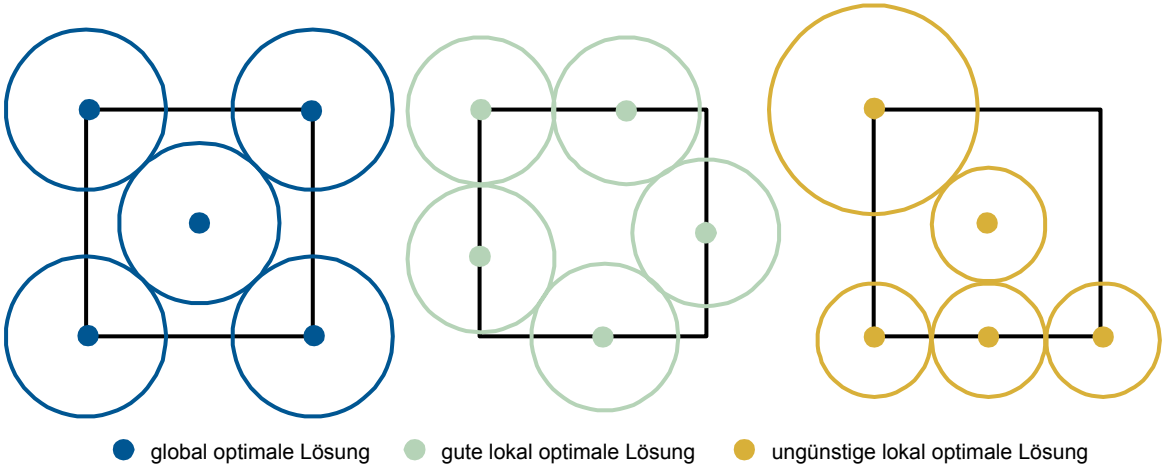


Abbildung 7-2: Drei mögliche Fälle für die Lösung eines Platzierungsproblems; globale Optimalität (links), gute lokale Optimalität (Mitte), ungünstige lokale Optimalität (rechts)

Die Anwendung eines lokalen Verfahrens, beispielsweise des SQP-Verfahrens, kann in Abhängigkeit von den genutzten Startpunkten jede dieser Lösungen hervorbringen (vgl. auch Ausführungen zur Facility Location im Abschnitt 6.3.2.2). Typischerweise wird bei praktischen Berechnungen und in (kommerziellen) Optimierungsprogrammen (z.B. Solver[56]) diesem Umstand dadurch Rechnung getragen, dass, ausgehend von verschiedenen (randomisierten) Startpunktkonstellationen, mehrere Lösungen berechnet werden (vgl. auch Abschnitt 6.3.2.2). Die beste lokale Lösung wird dann als nahe am globalen Optimum angenommen. Offensichtlich muss aber ein solches Vorgehen auf Kosten der Effizienz gehen.

Bei Einsatz eines globalen Verfahrens, d.h. des Branch-and-Bound-Algorithmus, würde in jedem Fall eine global optimale Lösung gefunden. Allerdings wurde bereits anhand der Arbeiten zur Facility Location beschrieben, dass auch eine auf Distanzprobleme zugeschnittene Suche nicht sehr effizient ist, d.h. keine Lösung on demand ermöglicht. Eine Anwendung auf das Farbproblem wäre unter Nutzung eines globalen Suchverfahrens (z.B. Backtracking, Abschnitt 6.6.2) für die Bestimmung einer lokal optimalen Lösung möglich, indem aus Sicht des Constraint Programming eine „Lockerung“ des Problems MAXMIN 1 formuliert wird. Diese Lockerung wird erreicht, indem auf die Zielfunktion verzichtet und ein Constraint Satisfaction Problem formuliert wird:

[56] http://www.solver.com (Zuletzt geprüft am 23.10.08)

$$d_{\min} \le \left\| X_i, Y_j \right\|_2, \qquad i = 1,\ldots,n,\ j = 1,\ldots,m+n,\ j > i$$

$$a_k^T X_i \le b_k, \qquad i = 1,\ldots,n,\ k = 1,\ldots,9\,.$$

Allerdings hängt die Lösbarkeit und die Lösung dieses Problems (ähnlich wie die Lösung des Problems MAXSUM im Abschnitt 6.1) maßgeblich von der Wahl des Mindestabstands $d_{\min}$ ab: Ein niedriger Wert verfehlt den Zweck der Bestimmung möglichst gut unterscheidbarer Farben, ein zu groß gewählter Wert erschwert die Berechnung oder macht das Problem sogar unlösbar. Für die Festlegung von $d_{\min}$ wäre demnach die Nutzung von Vorwissen erforderlich. Dieses Wissen müsste mindestens eine Information darüber enthalten, wie groß der Mindestabstand bei einer Anzahl von $m+n$ Farben in der Menge $Y = (X, F)$ der gesuchten und gegebenen Farben sein darf. Allerdings würde solch ein allgemeiner Wert davon ausgehen, dass die bereits vorhandenen Farben der Menge F untereinander ebenfalls diesen Abstand besitzen, was aber im Allgemeinen nicht angenommen werden kann. Somit müsste für jede Menge F gegebener Farben zunächst eine Abschätzung vorgenommen werden. Aus diesem Grunde wird die Formulierung des Farbproblems als Constraint Satisfaction Problem verworfen.

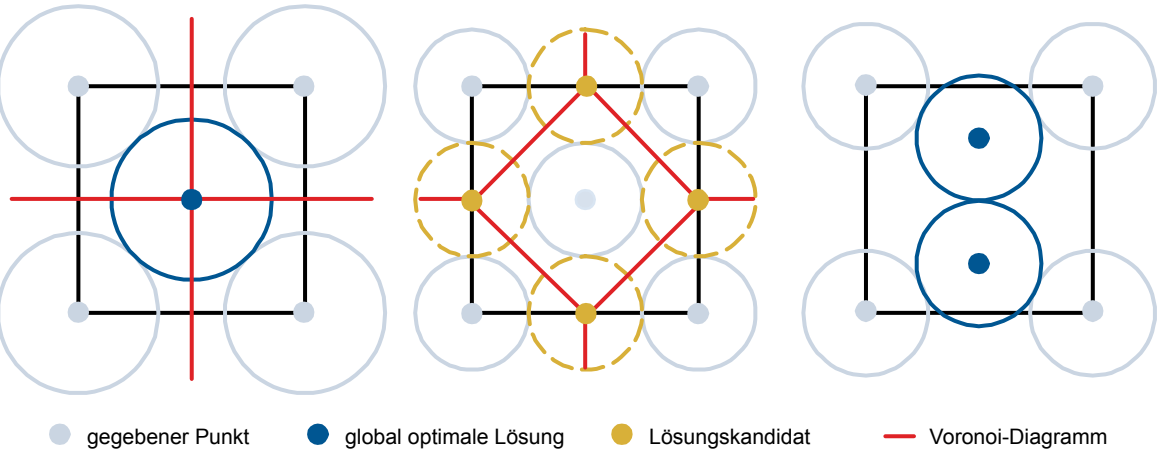


Abbildung 7-3: Platzierung von zwei Punkten in einem Quadrat durch sukzessives Einfügen in den Mittelpunkt des größten leeren Kreises; links die Konstellation des ersten, global optimal eingefügten Punktes; in der Mitte die Lösungskandidaten für den zweiten Punkt; rechts eine mögliche global optimale Lösung

Anders verhält es sich mit dem in Abschnitt 6.4 beschriebenen Verfahren zur Berechnung des größten leeren Kreises bzw. der größten leeren Kugel. Dieses Verfahren weist sowohl Eigenschaften eines lokalen als auch eines globalen Verfahrens auf. Ersteres wird wieder anhand eines einfachen Beispiels in der Ebene deutlich. Es seien wieder vier Punkte in einem Quadrat gegeben, zwei weitere sollen zusätzlich innerhalb des Quadrats platziert werden. Diese Platzierung soll durch sukzessives Hinzufügen des Mittelpunktes des jeweils größten leeren Kreises erfolgen. Abbildung 7-3 illustriert die Lage der gegebenen und der neu bestimmten Punkte. Das Einfügen des ersten Punktes ergibt ein global optimales Ergebnis (Abbildung 7-3 links). Für die Platzierung des zweiten Punktes stehen vier gleichwertige Lösungskandidaten zur Verfügung (Abbildung 7-3 Mitte). Der Vergleich dieser beiden Schritte mit einer mögli-

chen global optimalen Lösung in Abbildung 7-3 rechts zeigt, dass zwar in diesem Fall jedes Einfügen eines einzelnen Punktes in eine jeweils vorhandene Punktkonstellation eine global optimale Lösung ergibt, dies für das sukzessive Einfügen mehrerer Punkte aber nicht gilt.

Andererseits erlaubt die Berechnung des größten leeren Kreises in einem gewissen Rahmen eine globale Bewertung einer Punktkonstellation. Beispielsweise würde sich die in Abbildung 7-2 rechts dargestellte ungünstige lokale Lösung durch die Berechnung des größten leeren Kreises feststellen lassen (Abbildung 7-4 links): Es handelt sich um eine ungünstige Lösung, wenn ein Kreis gefunden wird, dessen Radius größer ist als der kleinste Abstand zwischen zwei Punkten der aktuellen Punktkonstellation. In Abbildung 7-4 (links) ist der Mittelpunkt dieses Kreises aus der Menge der Voronoi-Knoten, der Schnittpunkte von Voronoi-Kanten mit Kanten des Quadrats und Eckpunkten des Quadrats durch den vierten, bisher nicht besetzten, Eckpunkt gegeben (vgl. Abschnitt 6.4). Der Radius des Kreises ist größer als der kleinste Abstand der fünf Punkte der ungünstigen lokalen Lösung.

Über die Bewertung hinaus erlaubt die Berechnung des größten leeren Kreises offensichtlich auch eine Verbesserung durch sprunghafte Bewegungen von Punkten: In der Abbildung 7-4 (links) lässt sich der am schlechtesten platzierte Punkt in Richtung des Pfeils in den Mittelpunkt des größten leeren Kreises verschieben. Ergebnis ist in diesem Fall eine global optimale Lösung, die in Abbildung 7-4 rechts dargestellt ist. Das Beispiel der Abbildung 7-2 (Mitte) zeigt aber, dass dieses Vorgehen offensichtlich nicht die Unterscheidung einer guten lokalen von einer global optimalen Lösung ermöglicht.

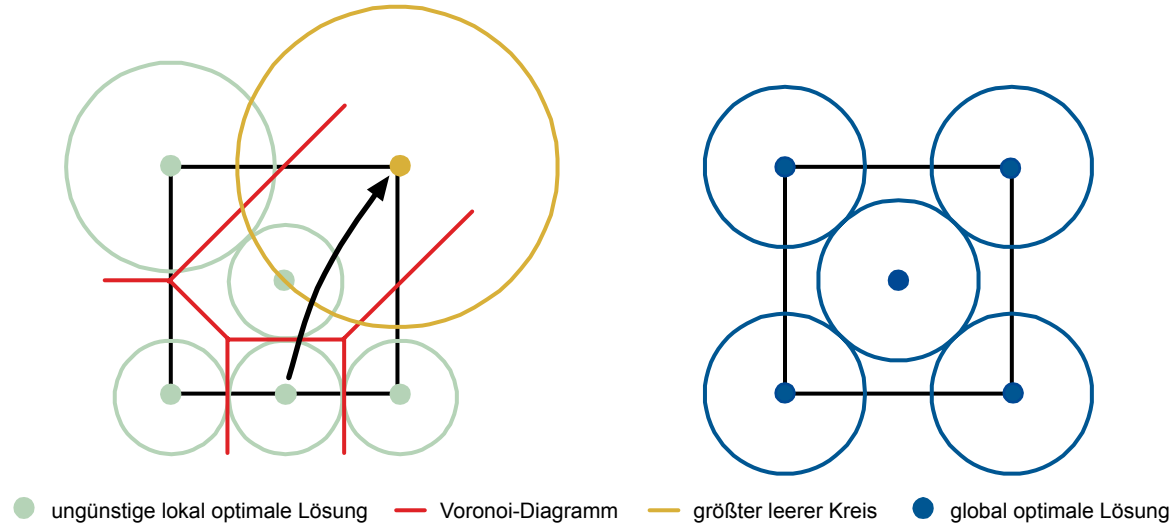


Abbildung 7-4: Sprunghafte Verbesserung einer ungünstigen lokalen Lösung durch die Berechnung des größten leeren Kreises (links); nach Verbesserung erhaltene global optimale Lösung (rechts)

Die Lösung des Farbproblems bedarf damit einer Abwägung zwischen einer effizienten Berechnung und der Bestimmung einer bestmöglichen – global optimalen – Lösung. Im Anwendungskontext dieser Arbeit ist offensichtlich die Effizienz von primärer Bedeutung: Eine Lösung soll „on demand“ gefunden werden, d.h. die Dauer der Berechnung sollte mit wenigen Sekunden in der Größenordnung liegen, die eine typische Kartenabfrage im WWW benö-

tigt. Die Schwierigkeit dieser Anforderung wird noch einmal im Rückblick auf die vergleichbaren Arbeiten der Facility Location (Abschnitt 6.3.2.2) deutlich. Diese zeigen, dass auch für kleinere Probleme sehr schnell Rechenzeiten von mehreren Minuten auftreten (z.B. Welch et al. 2006).

Um sowohl Effizienz als auch Optimalität Rechnung zu tragen, wird in dieser Arbeit von einer „Lockerung“ ausgegangen, die auch bereits im Zusammenhang mit einer Arbeit der Facility Location beschrieben wurde (vgl. Abschnitt 6.3.2.2): Es wird darauf verzichtet, eine Lösung zu finden, deren globale Optimalität gesichert ist. Stattdessen wird eine gute lokal optimale Lösung, die mit einem lokalen Verfahren effizient erhalten werden kann, als ausreichend angenommen. Die Bestimmung dieser Lösung bzw. das genutzte Verfahren wird durch die Bestimmung möglichst geeigneter Startpunkte unterstützt.

Detaillierte Ausführungen zum weiteren Vorgehen werden in den beiden folgenden Abschnitten gegeben.

7.2 Verfahren zur Lösung des Farbproblems

Im letzten Abschnitt wurde postuliert, dass eine lokale Lösung, die einem globalen Optimum nahe kommt, als Ergebnis des Farbproblems ausreichend ist. Das dafür erforderliche Vorgehen, das eine Berechnung on demand ermöglicht, lässt sich grob in drei Schritten zusammenfassen:

- Bestimmung geeigneter Startpunkte,
- Optimierung der Startpunktkonstellation durch Berechnung einer „guten“ lokalen Lösung mit Hilfe eines lokalen Verfahrens,
- Bewertung der Lösung durch Berechnung der größten leeren Kugel und – falls nötig – eine globale Verbesserung durch sprunghafte Bewegung von Punkten. Letzteres sollte nach Optimierung durch das lokale Verfahren allerdings die Ausnahme sein.

Die Bestimmung der Startpunkte wird detailliert im nächsten Abschnitt beschrieben, ein Verfahren zur Bestimmung der größten leeren Kugel (bzw. des größten leeren Kreises) wurde bereits im Abschnitt 6.4 skizziert, eine ausführlichere Vorgehensweise wird in diesem Abschnitt geschildert. Für die effiziente Bestimmung eines lokalen Minimums wird das SQP-Verfahren genutzt.

Die genannten Schritte bzw. Verfahren werden durch die Funktion „berechneFarben(…)“ – die Hauptmethode der Farberechnung – aufgerufen (Algorithmus 7-1).

Diese Funktion bekommt als Parameter die Anzahl der gesuchten Farben, die bereits im Farbraum vorhandenen Farben mit ihren RGB-Werten und die Ebenenparameter des Farbraumpolyeders übergeben. Wesentliche Schritte des Algorithmus erfolgen dann in folgenden Zeilen:

- *Zeile 4:* Die für das lokale Verfahren benötigten Startpunkte werden bestimmt. Eine detaillierte Beschreibung der Funktion „bestimmeStartpunkte(…)“ erfolgt im nächsten Abschnitt.

- *Zeile 5:* Der Aufruf des SQP-Verfahrens ergibt eine lokale Verbesserung der Lage der Startpunkte.
- *Zeilen 6-13:* Durch die Berechnung der größten leeren Kugel wird geprüft, ob es sich bei dem Ergebnis der Optimierung um eine ungünstige lokale Lösung handelt, die durch sprunghafte Punktbewegungen verbessert werden kann. Ist dies der Fall, erfolgt die Überprüfung solange, bis ein Mittelpunkt einer größten leeren Kugel keine Verbesserung der aktuellen Punktkonstellation mehr ergibt. Die Funktion „groessteLeereKugel(…)“ wird im Folgenden detaillierter beschrieben.

Procedure berechneFarben (n, GegebenRGB, Polyeder)

Gegeben: *n := Anzahl der zu bestimmenden Farben*
GegebenRGB := gegebene, feste Farbwerte im RGB-Farbraum (Menge F)
Polyeder := Ebenenparameter des Polyeders des Optimierungsfarbraums

Gesucht: *RGB-Werte von n Farben, die untereinander und von den Farben in GegebenRGB gut unterscheidbar sind (Menge X)*

```
1   for each Farbe in GegebenRGB
2       GegebenLUV[ i ] := Transformiere Farbe nach CIELUV;
3   end
4   Startpunkte = bestimmeStartpunkte(n, GegebenLUV, Polyeder);
5   NeuLUV := Verbessere die Lage der Startpunkte durch ein lokales
    Optimierungsverfahren (hier: SQP-Verfahren);
6   do
7       Berechne die Minimumdistanz eines Punktes aus NeuLUV zu
        NeuLUV ∪ GegebenLUV;
8       Berechne den am SchlechtestenPlatziertenPunkt aus NeuLUV zu
        NeuLUV ∪ GegebenLUV;
9       [Radius, Mittelpunkt] = groessteLeereKugel(NeuLUV, GegebenLUV, Polyeder);
10      if Radius > Minimumdistanz
11          Ersetze SchlechtestenPlatziertenPunkt in NeuLUV durch Mittelpunkt;
12      end
13  while Radius > Minimumdistanz;
14  for each Farbe in NeuLUV
15      NeuRGB[ i ] := Transformiere Farbe nach RGB;
16  end
17  return NeuRGB;
end
```

Algorithmus 7-1: Funktion „berechneFarben(…)“ zur Bestimmung von n gut unterscheidbaren Farben

Ein Algorithmus zur Bestimmung des größten leeren *Kreises* wurde bereits im Abschnitt 6.4 skizziert. Die potentiellen Mittelpunkte des Kreises waren nach der Berechnung des Voronoi-Diagramms einer Punktmenge in der Menge der Knoten des umschließenden Polygons, den Voronoi-Knoten und den Schnittpunkten zwischen Voronoi-Kanten und Kanten des Polygons zu suchen. Angewandt auf das Farbproblem und die Berechnung der größten leeren Kugel besteht diese Punktmenge aus den Voronoi-Knoten, den Extrempunkten der konvexen Hülle des Optimierungsfarbraums und den Durchstoßpunkten der Voronoi-Kanten durch die Flächen des Farbraumpolyeders.

Die Extrempunkte sind offenbar mit der konvexen Hülle des Optimierungsfarbraums bekannt, die Voronoi-Knoten werden mit Hilfe der im Abschnitt 6.2.2.2 beschriebenen Vorgehensweise zur Berechnung eines Voronoi-Diagramms erhalten. Die Durchstoßpunkte der Voronoi-Kanten durch die Flächen des Farbraumpolyeders lassen sich ermitteln, indem die Schnitte der Voronoi-Regionen mit dem Rand des Polyeders bestimmt werden. Unbeschränkte Voronoi-Regionen werden durch einen solchen Schnitt geschlossen, beschränkte Voronoi-Regionen verkleinert.

Die Schnitte können durch den Schnitt von Halbräumen, d.h. durch Verschneidung der Begrenzungsflächen der Voronoi-Regionen und des Farbraumpolyeders zu geschlossenen Polyedern erfolgen. Durchstoßpunkte sind dann diejenigen Eckpunkte dieser Polyeder, die auf dem Rand des Farbraums liegen.

Die Berechnung der größten leeren Kugel erfolgt durch die Funktion „groessteLeereKugel(…)“ im Algorithmus 7-2. Die Funktion bekommt als Parameter die oben bestimmten Farben im CIELUV-Farbraum, die vorab gegebenen Farben (ebenfalls im CIELUV-Farbraum) und die Ebenenparameter des Farbraumpolyeders übergeben. In den einzelnen Zeilen des Algorithmus werden dann folgende Schritte ausgeführt:

- *Zeile 1*: Das Voronoi-Diagramm der aktuellen Punktkonstellation von neu bestimmten und gegebenen Farben wird berechnet. Die zurückgegebene Struktur enthält die Knoten des Voronoi-Diagramms, die Ebenengleichungen der Voronoi-Flächen und die Zuordnung der Knoten und Ebenen zu Voronoi-Regionen.
- *Zeile 2-10*: Die größte leere Kugel, deren Mittelpunkt ein Voronoi-Knoten ist, wird bestimmt.
- *Zeile 11*: Aus den Parametern der Ebenen, die den Farbraumpolyeder begrenzen, werden durch Halbraumschnitt die Extrempunkte dieses Polyeders errechnet.
- *Zeilen 12-18*: Es wird geprüft, ob eine größte leere Kugel mit Mittelpunkt in der Menge der Extrempunkte des Farbraumpolyeders als insgesamt größte leere Kugel in Frage kommt.
- *Zeile 19*: Die einzelnen Voronoi-Regionen werden mit den Flächen des Farbraumpolyeders verschnitten. Ergebnis sind für jede Region die Koordinaten der Eckpunkte des entstandenen Polyeders.

```
Procedure groessteLeereKugel(NeuLUV, GegebenLUV, Polyeder)
    Gegeben: NeuLUV := n neu bestimmte Farben im CIELUV-Farbraum
             GegebenLUV := gegebene, feste Farbwerte im CIELUV-Raum (Menge F)
             Polyeder := Ebenenparameter des Polyeders des Optimierungsfarbraums
    Gesucht: Mittelpunkt und Radius der größten leeren Kugel der Gesamtmenge von
             Punkten in NeuLUV und GegebenLUV
1   Berechne das VoronoiDiagramm von NeuLUV ∪ GegebenLUV;
2   for each Knoten in VoronoiDiagramm
3       if Knoten in Polyeder
4           Bestimme den Radius der größten leeren Kugel um Knoten;
5           if Radius > bisher gefundener GroessterRadius einer leeren Kugel
6               Übernehme Radius als GroessterRadius;
7               Übernehme Knoten als Mittelpunkt;
8           end
9       end
10  end
11  Berechne durch Halbraumschnitt die Extrempunkte von Polyeder;
12  for each Extrempunkt in Extrempunkte
13      Bestimme für Extrempunkt den nächsten Nachbarn aus NeuLUV ∪ GegebenLUV
        und daraus den Radius der größten leeren Kugel;
14      if Radius > bisher gefundener GroessterRadius einer leeren Kugel
15          Übernehme Radius als GroessterRadius;
16          Übernehme Extrempunkt als Mittelpunkt;
17      end
18  end
19  Berechne durch Halbraumschnitt aus VoronoiDiagramm und Polyeder
    geschlossene VoronoiPolyeder;
20  for each VoronoiPolyeder_1 in VoronoiPolyeder
21      for each Flaeche von Polyeder
22          Schnittpunkte := Bestimme Extrempunkte von VoronoiPolyeder_1, die
            in Flaeche liegen;
23          for each Schnittpunkt in Schnittpunkte
24              Bestimme für Schnittpunkt den Radius der größten leeren Kugel;
25              if Radius > bisher gefundener GroessterRadius einer leeren Kugel
26                  Übernehme Radius als GroessterRadius;
27                  Übernehme Schnittpunkt als Mittelpunkt;
28              end
29          end
30      end
31  end
32  return GroessterRadius, Mittelpunkt;
end
```

Algorithmus 7-2: Funktion „groessteLeereKugel(..)“ zur Berechnung der größten leeren Kugel

- *Zeilen 20-31*: Aus der Menge der Eckpunkte der im letzten Schritt erhaltenen Polyeder werden die Punkte bestimmt, die auf den Flächen des Farbraums liegen (Durchstoßpunkte). Für diese Menge der Durchstoßpunkte wird dann ebenfalls die jeweils größte leere Kugel errechnet und geprüft, ob eine dieser Kugeln größer ist, als die größte bisher gefundene.

Die Effizienz des gesamten Verfahrens hängt primär von der Zahl der Iterationen des SQP-Verfahrens ab. Weitere Betrachtungen hierzu erfolgen im Zusammenhang mit der konkreten Berechnung von Farben im Abschnitt 7.4. Zuvor ist jedoch noch die Beschreibung eines geeigneten Verfahrens zur Bestimmung von Startpunkten notwendig.

7.3 Bestimmung von Startpunkten

In der Beschreibung des Lösungsverfahrens im letzten Abschnitt ist noch die Methode zur Bestimmung geeigneter Startpunkte offen geblieben. Vor der Beschreibung eines Verfahrens muss jedoch noch erörtert werden, welche Punkte als geeignet bezeichnet werden können.

7.3.1 Besonderheit des Farbproblems

Für alle folgenden Beschreibungen ist wieder die Modellvorstellung des Knapsack-Problems bzw. des Circle Packings vorteilhaft. Dieses Modell verdeutlicht eine besondere Eigenschaft des Farbproblems, die in den bisherigen Ausführungen zur Charakterisierung des Problems bereits verwendet, aber nicht explizit gemacht wurde. Packprobleme machen es erforderlich, dass alle betrachteten Flächen oder Volumina komplett innerhalb des zu packenden Containers liegen. Diese Forderung ist für das Farbproblem dagegen völlig ohne Belang. Hier genügt es, wenn sich der Mittelpunkt einer Kugel, d.h. der eigentliche Farbort, im Inneren des Farbraums befindet. Damit unterscheidet diese Eigenschaft das Farbproblem von allen anderen Packproblemen. Abbildung 7-5 zeigt die Situation im direkten Vergleich: Links ist das Packen von Kreisen in einen polygonalen Querschnitt des Optimierungsfarbraums dargestellt, die Kreise sind komplett enthalten. Der rechte Teil der Abbildung illustriert dagegen eine Situation, bei der lediglich die Kreismittelpunkte innerhalb des Polygons liegen müssen. Offensichtlich erhöht sich durch diesen Unterschied die Zahl der platzierbaren Kreise signifikant. Weiterhin ist die Zahl der Kreise, die den Rand des Polygons schneiden, größer als die Zahl derer, die komplett innerhalb des Polygons liegen.

Unter Berücksichtigung, dass bei gegebener Anzahl n von Kreisen oder Kugeln der Radius maximiert werden soll, lässt sich eine weitere Betrachtung vornehmen (im Folgenden in der Terminologie des dreidimensionalen Raumes). Dafür wird einerseits von dem verfügbaren Volumen V_R des Farbraums ausgegangen, andererseits von der Summe der Volumina der zu platzierenden Kugeln:

$$\sum_{i=1}^{n} V_i \,.$$

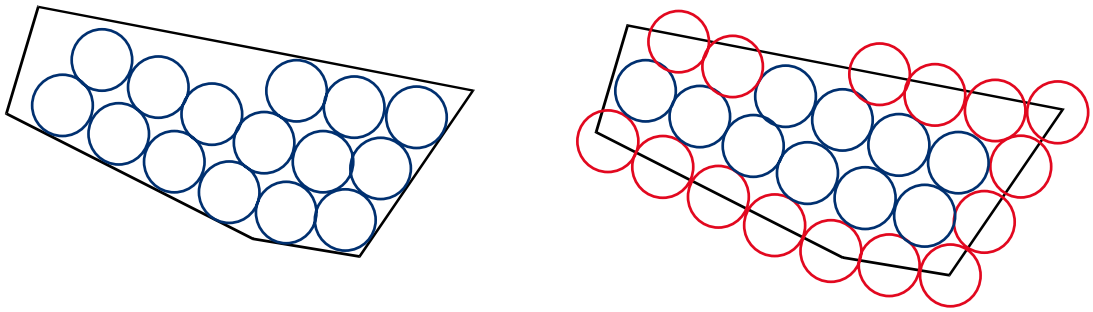


Abbildung 7-5: Packen von Kreisen in einen Querschnitt des Optimierungsfarbraums; links ein typisches Packproblem, bei dem die Kreisflächen komplett im Polygon liegen; rechts das Farbproblem, bei dem lediglich die Kreismittelpunkte innerhalb des Polygons liegen müssen.

Für eine möglichst gute Ausnutzung des verfügbaren Raumes muss offensichtlich

$$\sum_{i=1}^{n} V_i \rightarrow V_R \text{ gelten,}$$

d.h. die Summe der Kugelvolumina soll möglichst nahe an das Volumen V_R heranreichen (da Kugeln selbstverständlich nicht ohne Zwischenräume packbar sind, wird sich natürlich immer eine gewisse Differenz ergeben). Um dies zu erreichen, sind zwei Beobachtungen von Bedeutung:

- Es müssen Kugelmittelpunkte – Farben – so platziert sein, dass das durch diese Punkte aufgespannte Polyeder bzw. die konvexe Hülle dieser Punkte möglichst mit dem Polyeder des Optimierungsfarbraums identisch ist. Nur so besteht überhaupt die Voraussetzung, dass das Volumen V_R in Gänze verfügbar ist. Dies ist offenbar der Fall, wenn Punkte auf den Eckpunkten des Optimierungsfarbraums platziert werden (sofern dies nicht durch die minimale Distanz zu gegebenen Farben verhindert wird).
- Je weniger Volumen eine Kugel zur Gesamtsumme beiträgt, desto größer kann der gesuchte Radius der Kugeln werden.

Für den letztgenannten Punkt ist die Lage der Kugeln entscheidend. Gemäß der obigen Beschreibung zur Besonderheit des Farbproblems lassen sich für diese Lage drei Fälle differenzieren:

- Eine Kugel liegt komplett innerhalb des Farbraums und geht mit ihrem gesamten Volumen in die Summe der V_i ein.
- Der Kugelmittelpunkt ist in einer Fläche des Polyederrandes positioniert. Damit wird durch die Kugel lediglich $V_i / 2$ des Raumvolumens V_R beansprucht.
- Der Kugelmittelpunkt ist auf einem der Eckpunkte positioniert, die Kugel beansprucht so weniger als $V_i / 2$ des Gesamtvolumens. Der genaue Anteil ist abhängig von den Winkeln, die die im Eckpunkt zusammentreffenden Flächen einschließen.

Damit kommt dem Rand des Optimierungsfarbraums offensichtlich eine besondere Bedeutung für die Platzierung von Farben zu. Diese Bedeutung lässt sich durch den Schnitt von regelmäßigen Kugelpackungen mit dem Farbraumpolyeder bestätigen.

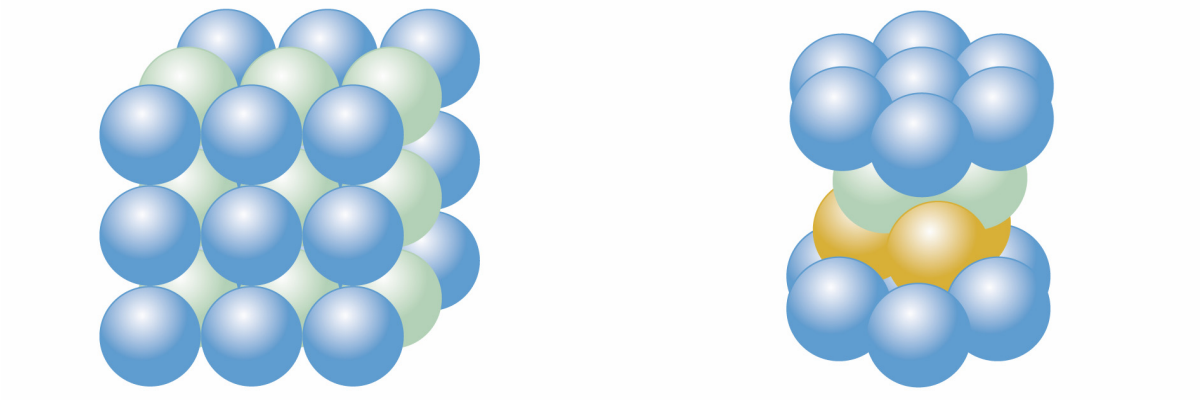


Abbildung 7-6: Regelmäßige (links) und kubische Kugelpackung (rechts)

Für regelmäßig gepackte Kugeln, deren Mittelpunkte auf einem Würfelgitter platziert werden (Abbildung 7-6, links), ergibt sich, dass bei einem Durchmesser von 45 Längeneinheiten (45 Längeneinheiten werden als Mindestabstand für die Unterscheidbarkeit von zwei Farben angenommen, vgl. Abschnitt 6.5.1) je nach Verschiebung der Packung gegenüber dem Farbraumpolyeder zwar 13-16 Kugelmittelpunkte innerhalb des Polyeders liegen, aber höchstens 25% der Kugeln mit ihrem gesamten Volumen. Für eine kubische Kugelpackung (Abbildung 7-6, rechts), deren Kugeln sehr viel dichter angeordnet sind, liegen bei gleichem Durchmesser 22-26 Mittelpunkte im Polyeder, allerdings nur 2-4 Kugeln in ihrer Gesamtheit.

7.3.2 Verfahren zur Bestimmung von Punkten auf dem Polyederrand

Wie bereits im Abschnitt 7.2, wird im Folgenden weiter angenommen, dass m Farben in der Menge F gegeben seien, weitere n der Menge X seien gesucht. Die Bestimmung der n Farben wurde im Abschnitt 7.2 durch die Funktion „berechneFarben(…)“ beschrieben; die für das lokale Optimierungsverfahren benötigten Startpunkte wurden durch den Funktionsaufruf „bestimmeStartpunkte(…)“ erhalten. Innerhalb dieser Funktion sollen nun n Startpunkte derart gefunden werden, dass der minimale Abstand der Punkte sowohl untereinander als auch gegenüber den m gegebenen Punkten möglichst groß wird.

Offensichtlich besitzt das Problem in dieser Formulierung eine ähnliche Komplexität wie das eigentliche Farbproblem. Aus diesem Grund wird auf die im Abschnitt 7.3.1 beschriebene Besonderheit des Farbproblems zurückgegriffen und angenommen, dass sich die Punkte auf dem Rand des Farbraumpolyeders sehr gut als Startpunkte eignen:

- Randpunkte sorgen für eine möglichst gute Ausnutzung des Optimierungsfarbraums.
- Die Komplexität der Punktbestimmung wird reduziert, indem nicht Punktbewegungen in einem dreidimensionalen Körper betrachtet werden, sondern lediglich Bewegungen auf den Begrenzungsflächen des Polyeders.

Im Folgenden wird ein allgemeines Verfahren beschrieben, das diese Besonderheit nutzt und Startpunkte möglichst gleichmäßig auf den Rand des Farbraums platziert. Vorraussetzung für dieses Verfahren ist eine Eigenschaft des Optimierungsfarbraums, die für den in dieser Arbeit genutzten Raum gegeben ist: Die Modellierung als konvexes Polyeder (vgl. Abschnitt 6.1), das sich als Schnitt von *k* Halbräumen bzw. Hyperebenen durch ebenso viele Ungleichungen repräsentieren lässt.

Diese Repräsentation des Polyeders und die gewünschte Bestimmung von Punkten auf dessen Rand sind zunächst vergleichbar mit den im Kapitel 6 beschriebenen linearen Programmen und deren Lösung durch das Simplexverfahren. Die zulässige Menge stellt dort in gleicher Weise ein Polyeder dar. Die Bestimmung einer Lösung durch das Simplex-Verfahren erfolgt, indem sich das Verfahren entlang der Kanten des Polyeders zu Extrempunkten mit größer werdendem Wert der Zielfunktion bewegt. Die Systematik der Bewegung resultiert aus der beschriebenen Wahl einer Basis bzw. eines Basiswechsels und entspricht einer Tiefensuche hin zum optimalen Ergebnis.

Das Verfahren zur Startpunktbestimmung greift diese Idee auf, bewegt sich in ähnlicher Weise über den Rand des Farbraumpolyeders und platziert Punkte auf diesem Rand. Allerdings erfolgt die Bewegung dabei nach Art einer Breitensuche: Für die Platzierung sind nicht nur Eckpunkte und Kanten von Bedeutung, sondern auch die Flächen der Polyederoberfläche, d.h. das Verfahren bewegt sich entlang benachbarter Elemente frei über die gesamte Oberfläche.

Ähnlich wie die Wahl der Basis, die das Simplex-Verfahren auf einen bestimmten Extrempunkt des Polyeders (zulässige Menge) festlegt, lassen sich beim Startpunktverfahren alle Elemente der Polyederoberfläche bzw. die Platzierung eines Punktes auf einem dieser Elemente betrachten, indem gefordert wird, dass bestimmte Ungleichungen aktiv (bindend), d.h. mit Gleichheit erfüllt, sind:

- Für eine bestimmte Fläche des Polyeders und die Platzierung eines Punktes in dieser Fläche muss die Nebenbedingung, die diese Fläche repräsentiert, aktiv sein, die übrigen *k-1* Nebenbedingungen inaktiv. Innerhalb einer Fläche ist ein Punkt durch ein Optimierungsverfahren frei verschiebbar.
- Für eine Kante des Polyeders gilt, dass zwei Nebenbedingungen aktiv und *k-2* Bedingungen inaktiv sein müssen. Ein Punkt, für den dies gültig ist, kann sich über die Kante von einer Fläche in benachbarte bewegen.
- Analog gilt für einen Eckpunkt des Polyeders, dass drei Nebenbedingungen aktiv und *k-3* Bedingungen inaktiv sind. Die Bewegung eines Punktes kann hier offensichtlich von einer Fläche in jeweils zwei andere erfolgen.

Das eigentliche Startpunktverfahren besteht im Wesentlichen aus drei Schritten. Zunächst wird nach Art der Beam Search eine Ausgangsmenge von Punkten auf dem Polyederrand bestimmt und aus dieser Menge anschließend ein Sample gezogen. Einzelne Punkte dieses Samples werden dann jeweils iterativ so über den Rand bewegt, dass sich durch Maximierung der minimalen Distanz zu allen anderen Punkten eine möglichst gute Verteilung ergibt.

Die Festlegung der Ausgangsmenge erfolgt, indem sowohl deterministisch als auch randomisiert *2n* Punkte so bestimmt werden, dass sie über die gesamte Polyederoberfläche verteilt sind. Dazu werden als erste die Eckpunkte des Polyeders in die Menge übernommen; die übrigen Punkte werden dadurch erhalten, dass zufällig Punkte in den Dreiecken der Delaunay-Triangulation der Oberfläche platziert werden. Das Vorgehen zur Bestimmung der Ausgangsmenge ist in der Funktion „bestimme2nPunkte(…)“ zusammengefasst (Algorithmus 7-3).

Procedure bestimme2nPunkte(n, Polyeder)

Gegeben: *n := Anzahl der zu bestimmenden Punkte*
Polyeder := Ebenenparameter des Polyeders des Optimierungsfarbraums

Gesucht: *Menge von 2*n Punkten auf dem Rand des Polyeders*

1 *Berechne durch Halbraumschnitt die* Extrempunkte *von* Polyeder;
2 *Kopiere* Extrempunkte *in* 2nPunkte*;*
3 *Berechne aus* Extrempunkte *die* DelaunayTriangulation *der konvexen Hülle des Polyeders;*
4 **while** *Anzahl Punkte in* 2nPunkte $< 2n$
5 *Füge* 2nPunkte *einen in* DelaunayTriangulation *zufällig gewählten Punkt hinzu;*
6 **end**
7 **return** 2nPunkte;

end

Algorithmus 7-3: Funktion „bestimme2nPunkte(…) zur Festlegung von 2n Punkten auf dem Rand des Farbraumpolyeders

Die Funktion „bestimme2nPunkte(…)“ bekommt die Anzahl der zu bestimmenden Punkte und die Ebenenparameter des Farbraumpolyeders übergeben. Die wichtigsten Schritte sind dann:

- *Zeilen 1 und 2*: Durch das bereits im Abschnitt 7.2 angesprochene Vorgehen des Schnitts von Halbräumen werden die Extrempunkte des Farbraumpolyeders explizit bestimmt und als erste in die Menge der *2n* Punkte übernommen.
- *Zeile 3*: Aus den Extrempunkten wird die Delaunay-Triangulation (vgl. Abschnitt 6.2.2.2) der konvexen Hülle berechnet.
- *Zeile 5*: Die Punkte, die nach Bestimmung der Extrempunkte noch benötigt werden, werden in zufällig gewählten Dreiecken der Triangulation der konvexen Hülle platziert. Die eigentliche Platzierung erfolgt ebenfalls randomisiert.

Anschließend wird aus den insgesamt *2n* Punkten dann das Sample von *n* Punkten derart gezogen, dass diese Punkte sowohl untereinander als auch gegenüber den gegebenen Punkten einen möglichst großen Abstand besitzen. Damit wird eine erste Menge X der gesuchten Punkte erhalten.

```
Procedure zieheSample(n, 2nPunkte, GegebenLUV)
    Gegeben: n := Anzahl der zu bestimmenden Punkte
             2nPunkte := Ausgangsmenge der 2*n Punkte
             GegebenLUV := gegebene, feste Farbwerte im CIELUV-Raum
                           (Menge F)
    Gesucht: Menge von n Punkten, die auf der Oberfläche des Farbraum-
             polyeders verteilt sind
1   Teile 2nPunkte randomisiert in n Startpunkte und n Restpunkte;
2   for k Versuche
3       Berechne die Minimumdistanz_1 eines Punktes aus Startpunkte zu
        Startpunkte ∪ GegebenLUV;
4       Berechne den am SchlechtestenPlatziertenPunkt_1 aus Startpunkte zu
        Startpunkte ∪ GegebenLUV;
5       for each Punkt aus Restpunkte
6           Ersetze SchlechtestenPlatziertenPunkt_1 aus Startpunkte durch Punkt;
7           Berechne die Minimumdistanz_2 eines Punktes aus Startpunkte zu
            Startpunkte ∪ GegebenLUV;
8           Berechne den am SchlechtestenPlatziertenPunkt_2 aus Startpunkte zu
            Startpunkte ∪ GegebenLUV;
9           if Minimumdistanz_2 > Minimumdistanz_1
10              break;
11          else
12              Lösche Punkt aus Startpunkte und stelle ursprüngliche
                Konstellation wieder her;
13          end
14      end
15  end
16  return Startpunkte;
end
```

Algorithmus 7-4: Funktion „zieheSample(…) zum Ziehen eines Samples von n Punkten mit maximalem Abstand aus einer Menge von 2n Punkten

Das Ziehen des Samples ist durch die Funktion „zieheSample(…)“ umgesetzt (Algorithmus 7-4). Diese Funktion bekommt als Parameter die Anzahl der gesuchten Punkte, die Menge der *2n* Ausgangspunkte und die bereits im Farbraum vorhandenen Farben übergeben. Wesentliche Schritte des weiteren Vorgehens sind dann:

- *Zeile 1*: Die Menge der *2n* Ausgangspunkte wird zufallsbasiert in eine Menge von *n* Startpunkten und *n* „Restpunkten“ geteilt.
- *Zeilen 2-15*: In *k* Iterationen, wobei *k* frei wählbar ist, wird jeweils versucht, den am schlechtesten platzierten Startpunkt gegen ein bisher nicht gewähltes Element der

Restpunkte derart auszutauschen, dass sich eine Verbesserung der minimalen Distanz zwischen einem Punkt aus der Menge der Startpunkte zu den Punkten aus der Gesamtmenge der Startpunkte und gegebenen Punkte ergibt.

Die Lage dieser nach Ziehen des Samples auf dem Rand des Farbraumpolyeders erhaltenen Punkte soll nun weiter so verbessert werden, dass, im Sinne eines Distanzproblems, eine möglichst optimale Verteilung auf dem Polyeder erhalten wird. Dies wird erreicht, indem iterativ jeweils ein einzelner Punkt innerhalb der Polyederfläche, in der er liegt, durch ein lokales Optimierungsverfahren verschoben und optimal platziert wird. Kandidat für diese Verschiebung ist immer der Punkt, der in der aktuellen Konstellation nach dem Kriterium des Maximums der minimalen Distanz $d_{\min}$ am schlechtesten platziert ist. Wurde ein solcher Punkt X_K gefunden, muss zunächst über die Gleichsetzung der Ungleichungen $a_i^T X_K \le b_i$ festgestellt werden, in welcher Fläche *k* des Polyeders der Punkt liegt. Wurde eine solche Fläche gefunden, wird innerhalb der Fläche für den betrachteten Punkt das Optimierungsproblem

$$\text{MAXMINRAND} \quad \text{maximize} \quad \min\left(d\left(X_K, Y_i\right)\right) \quad i = \{1,\ldots,m+n\} \setminus K$$

$$\text{subject to} \quad \begin{array}{l} a_1^T X_K \le b_1 \\ \vdots \\ a_j^T X_K \le b_j \\ a_k^T X_K = b_k \\ a_l^T X_K \le b_l \\ \vdots \\ a_p^T X_K \le b_p \end{array}$$

gelöst. Dieses Problem maximiert die kleinste Distanz des am schlechtesten platzierten Punktes X_K zu allen anderen Punkten der Menge $Y = (X, F)$. Durch die Nebenbedingungen, insbesondere die Bindung der Ebenengleichung, auf der sich der Punkt befindet, wird die zulässige Menge auf ein begrenzendes Polygon des Farbraumpolyeders eingeschränkt.

Falls nach Lösung dieses Optimierungsproblems für $d_{\min}$ ein höherer Wert erreicht wird, kann durch die Übernahme der neuen Lage von X_K eine verbesserte Startpunktkonstellation erhalten werden. Unabhängig von dieser Übernahme kann X_K weiterhin der Punkt sein, der am schlechtesten platziert ist. Die Lage des Punktes muss dann vertiefend betrachtet werden. Falls weiterhin die Nebenbedingung der Ebene *k* mit Gleichheit erfüllt ist, d.h.

$$\begin{array}{l} a_1^T X_K \le b_1 \\ \vdots \\ a_j^T X_K \le b_j \\ a_k^T X_K = b_k \\ a_l^T X_K \le b_l \\ \vdots \\ a_p^T X_K \le b_p \end{array}$$

gilt, liegt der Punkt im Inneren des entsprechenden Polygons. In diesem Fall ist keine weitere Verbesserung der Punktlage – und der gesamten Punktmenge – durch eine kontinuierliche Punktbewegung innerhalb des Polygons möglich und das Verfahren kann beendet werden.

Erfüllt X_K zwei der Nebenbedingungen mit Gleichheit, liegt er offensichtlich auf einer Kante des Polyeders. Diese Lage erlaubt eine Bewegung des Punktes in eine zur Fläche *k* benachbarte Fläche *j*, indem in der nächsten Iteration die Bindung zur Fläche *k* aufgehoben wird und lediglich noch die Nebenbedingung *j* bindend ist:

$$\begin{array}{ll} a_1^T X_K \le b_1 & a_1^T X_K \le b_1 \\ \vdots & \vdots \\ a_j^T X_K = b_j & a_j^T X_K = b_j \\ a_k^T X_K = b_k \quad \rightarrow & a_k^T X_K \le b_k \\ a_l^T X_K \le b_l & a_l^T X_K \le b_l \\ \vdots & \vdots \\ a_p^T X_K \le b_p & a_p^T X_K \le b_p \end{array}$$

Das Problem MAXMINRAND wird dann mit diesen Nebenbedingungen erneut gelöst.

Falls ein Punkt sogar drei Nebenbedingungen mit Gleichheit erfüllt, handelt es sich um einen Extrempunkt des Polyeders. In diesem Fall ist eine Bewegung des Punktes aus der Fläche *k* in eine der benachbarten Flächen *j* und *l* möglich, d.h. für die nächste Iteration des Verfahrens wird nur noch eine der Flächen *k*, *j* und *l* als bindend festgelegt:

$$\begin{array}{ll} a_1^T X_K \le b_1 & a_1^T X_K \le b_1 \\ \vdots & \vdots \\ a_j^T X_K = b_j & a_j^T X_K \le b_j \\ a_k^T X_K = b_k \quad \rightarrow & a_k^T X_K \le b_k \\ a_l^T X_K = b_l & a_l^T X_K = b_l \\ \vdots & \vdots \\ a_p^T X_K \le b_p & a_p^T X_K \le b_p \end{array}$$

Für diese Lage in der Fläche *l* wird dann ebenfalls das Problem MAXMINRAND erneut gelöst.

Auf diese Weise kann sich ein Punkt so lange über den gesamten Rand des Polyeders bewegen, bis entweder eine verbesserte Position gefunden ist, oder alle Ebenen besucht wurden, ohne dass eine Verbesserung möglich ist. Damit das Startpunktverfahren terminiert, muss allerdings sichergestellt sein, dass ein Punkt jeweils nur in noch nicht von ihm besuchte Flächen wandert, d.h. besuchte Flächen müssen gemerkt werden.

Durch diese Betrachtung der Lage und – falls möglich – die sukzessive Bewegung eines Punktes in jeweils benachbarte Flächen wird in Analogie zur Wanderung des Simplex-Verfahrens entlang der Kanten eines Polyeders eine Wanderung auf der gesamten Oberfläche

des Polyeders beschrieben. Die fortgesetzte Gleichsetzung der Ungleichungen entspricht dabei einem Basiswechsel beim Simplex.

Das geschilderte Vorgehen wird durch die Funktion „optimiereAufRand(…)“ in Algorithmus 7-5 umgesetzt.

```
Procedure optimiereAufRand(nPunkte, GegebenLUV, Polyeder)
    Gegeben: nPunkte := Menge von n Punkten, die auf der Oberfläche des
                        Farbraumpolyeders verteilt sind
             GegebenLUV := gegebene, feste Farbwerte im CIELUV-Raum
                           (Menge F)
             Polyeder := Ebenenparameter des Polyeders des Optimierungs-
                         farbraums
    Gesucht: Verbesserte Verteilung der nPunkte auf der Oberfläche des
             Farbraumpolyeders
1   for k Versuche
2       Bestimme die Minimumdistanz_1 eines Punktes aus nPunkte zu
        nPunkte ∪ GegebenLUV;
3       Bestimme den am SchlechtestenPlatziertenPunkt aus nPunkte zu
        nPunkte ∪ GegebenLUV;
4       Bestimme Flaeche von Polyeder, auf der SchlechtestenPlatziertenPunkt liegt,
        berücksichtige dabei nur Flächen, die für diesen Punkt noch nicht
        betrachtet wurden;
5       if Flaeche existiert
6           Berechne VerbessertenPunkt des SchlechtestenPlatziertenPunkt durch
            lokales Optimierungsverfahren innerhalb Flaeche so, dass die
            Minimumdistanz_2 zu allen anderen Punkten aus nPunkteund
            GegebenLUV maximal wird;
7           Merke Flaeche als besucht;
8       end
9       else break;
10      if Minimumdistanz_2 > Minimumdistanz_1
11          Übernehme VerbessertenPunkt in nPunkte;
12      end
13  end
14  return nPunkte;
end
```

Algorithmus 7-5: Funktion „optimiereAufRand(…)“ zur Optimierung der Lage von n Punkten auf dem Rand des Farbraumpolyeders

Diese Funktion bekommt als Parameter das zuvor bestimmte Sample von *n* Punkten, die bereits im Farbraum vorhandenen Farben und die Ebenenparameter des Farbraumpolyeders

übergeben. In *k* Versuchen wird dann die oben beschriebene lokale Verbesserung und Bewegung von Punkten durchgeführt. Die Anzahl der Versuche ist zwar frei wählbar, allerdings nach oben durch das Produkt aus *n* und der Anzahl der Polyederflächen begrenzt: Jeder Punkt kann sich höchstens einmal über den gesamten Rand bewegen.

Die Integration der in diesem Abschnitt bisher beschriebenen Methoden in einer Funktion „bestimmeStartpunkte(…)“ ergibt dann den durch Algorithmus 7-6 beschriebenen Ablauf.

Procedure bestimmeStartpunkte(n, GegebenLUV, Polyeder)

Gegeben: *n := Anzahl der zu bestimmenden Punkte*
GegebenLUV := gegebene, feste Farbwerte im CIELUV-Raum (Menge F)
Polyeder := Ebenenparameter des Polyeders des Optimierungsfarbraums

Gesucht: *n gut auf der Oberfläche des Farbraumpolyeders verteilte Punkte*

```
1  2nPunkte = bestimme2nPunkte(n, Polyeder);
2  nPunkte = zieheSample(n, 2nPunkte, GegebenLUV);
3  Startpunkte = optimiereAufRand(nPunkte, GegebenLUV, Polyeder);
4  return Startpunkte;
end
```

Algorithmus 7-6: Funktion „bestimmeStartpunkte(…)“ zur Bestimmung von n Startpunkten auf dem Rand des Farbraumpolyeders

Die Funktion „bestimmeStartpunkte(…)“ bekommt als Parameter die benötigte Anzahl der Startpunkte, die Ebenenparameter der begrenzenden Flächen des Farbraumpolyeders und die bereits im Farbraum vorhandenen Farben übergeben. Anschließend wird dann zunächst die Ausgangsmenge der Punkte bestimmt, aus dieser Menge das Sample gezogen. Nach sukzessiver Verbesserung des Samples erfolgt die Rückgabe der Startpunkte an die aufrufende Funktion „berechneFarben(…)“ im Abschnitt 7.2.

Die bisher in diesem Kapitel skizzierten Verfahren werden im nächsten Abschnitt zur Berechnung von Farben genutzt.

7.4 Berechnung und Ergebnisse

In diesem Abschnitt werden der Algorithmus aus Kapitel 7.2 und die Methode zur Bestimmung von Startpunkten aus Kapitel 7.3 genutzt, um anhand von zwei Beispielen konkrete Berechnungen von Farben durchzuführen:

- Im Abschnitt 7.4.1 werden 20 Farben im „leeren“ Farbraum platziert, d.h. es sind keine Farben der Menge *F* vorab gegeben ($F = \varnothing$). Zu maximieren sind lediglich die Abstände zwischen den neu zu bestimmenden Farborten.
- Im Abschnitt 7.4.2 erfolgt eine Lösung des einleitend beschriebenen Beispiels der Stadtplankarte, d.h. die Farben der Menge *F* sind durch die Farben dieser Karte gege-

ben. Die neu zu berechnenden Farben sollen sowohl untereinander als auch gegenüber den Farben aus F einen maximalen Abstand besitzen.

Anhand der Bestimmung von Farben im leeren Farbraum wird darüber hinaus die Zweckmäßigkeit der Bestimmung von Startpunkten auf dem Rand des Farbraumpolyeders deutlich.

7.4.1 Optimale Farben im „leeren" Farbraum

Es werden nun 20 gut unterscheidbare Farben im leeren Farbraum ($F = \varnothing$, $n = 20$) bestimmt und durch die Berechnung mit verschiedenen Startpunktkonstellationen die Präferenz für die Platzierung der Startpunkte auf dem Rand des Farbraumpolyeders bestätigt.

Da die Startpunkte direkt vom lokalen Optimierungsverfahren, dem SQP-Verfahren, genutzt werden, findet die Eignung von Punkten ihren Ausdruck im Verhalten und den Ergebnissen dieses Verfahrens: Von zwei Mengen von Punkten eignet sich offensichtlich diejenige besser als Menge von Startpunkten, die in kürzerer Zeit, d.h. nach weniger Iterationen des Verfahrens, die Berechnung eines besseren Ergebnisses für den Wert der Zielfunktion – einer größeren minimalen Distanz – ermöglicht.

Die verschiedenen Mengen der Startpunkte setzen sich jeweils aus unterschiedlichen Verhältnissen von Punkten auf dem Rand und im Inneren des Farbraumpolyeders zusammen: Jede Menge enthält r Randpunkte mit $r = 20,19,...,1,0$ und n-r Punkte im Inneren des Polyeders. Insgesamt werden also 21 Mengen verschiedener Punktverhältnisse erhalten. Zur Durchführung der Berechnung wird das in Abschnitt 7.2 beschriebene Vorgehen der Farbberechnung leicht modifiziert:

- Durch das Verfahren „bestimmeStartpunkte(…)" werden statt n Startpunkte lediglich r bestimmt. Die n-r Punkte im Inneren werden über eine Zufallsfunktion platziert.
- Das Verfahren wird nach Ausführung des SQP-Verfahrens abgebrochen, d.h. auf Berechnung der größten leeren Kugel wird zunächst verzichtet.

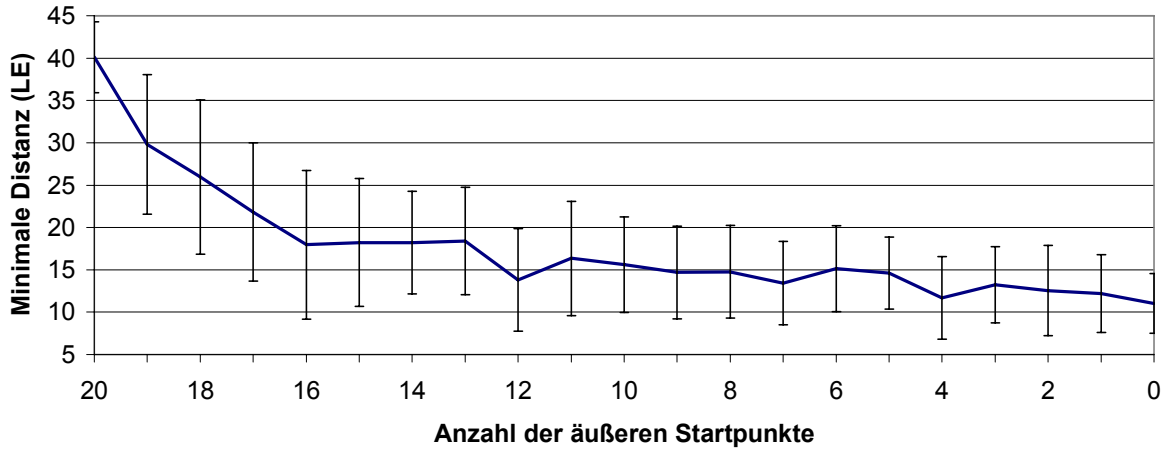


Abbildung 7-7: Minimale Distanzen der Startpunktmengen für den leeren Farbraum in Abhängigkeit von der Anzahl der äußeren Startpunkte (Mittelwerte und Standardabweichungen von jeweils 25 Startpunktmengen, LE bezeichnet Längeneinheiten im CIELUV-Farbraum)

Weiterhin werden für jedes Verhältnis *r/(n-r)* von Punkten auf dem Rand und im Inneren nicht nur eine, sondern 25 verschiedene Mengen betrachtet und die Ergebnisse jeweils gemittelt. Abbildung 7-7 zeigt für die Startmengen die Mittelwerte und Standardabweichungen der jeweils minimalen Distanz.

Abbildung 7-8 zeigt die erreichten minimalen Distanzen für eine Auswahl von Startpunktmengen verschiedener Punktverhältnisse in Abhängigkeit von der Anzahl der durchgeführten Iterationen des SQP-Verfahrens. Offensichtlich sind für Startvektoren mit einer großen Anzahl von Randpunkten bereits nach einer geringen Anzahl von Iterationen wesentliche Verbesserungen erreicht. Für die folgenden Berechnungen werden deshalb 100 Iterationen als ausreichend erachtet, um ein gutes Ergebnis der lokalen Optimierung zu erhalten.

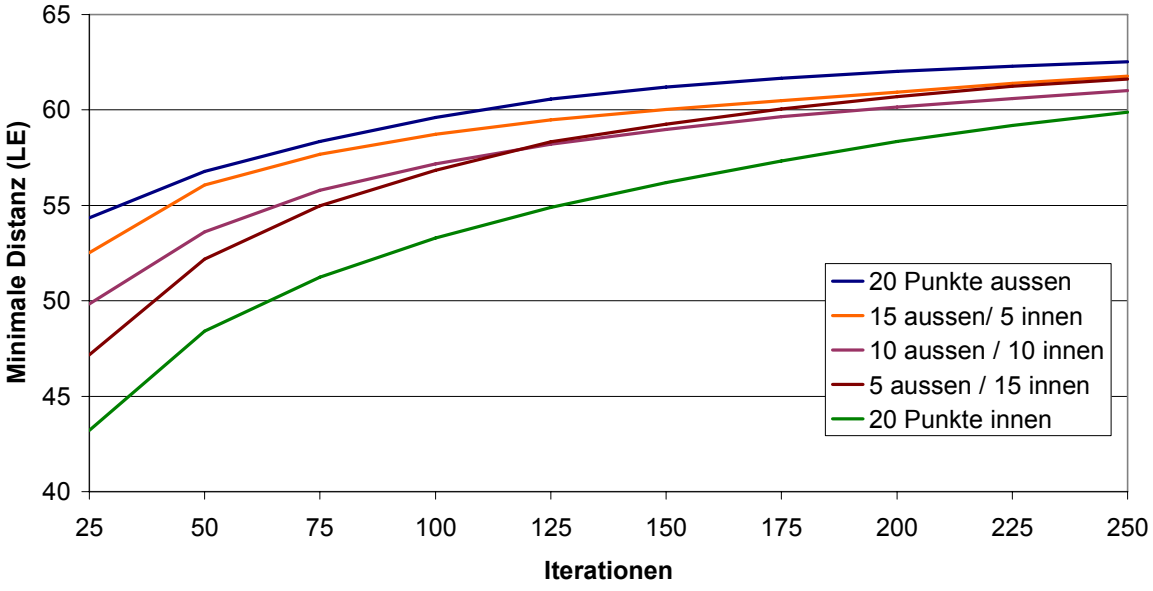


Abbildung 7-8: Minimale Distanzen für eine Auswahl von Startpunktmengen im leeren Farbraum in Abhängigkeit von der Anzahl der durchgeführten Iterationen (Mittelwerte und Standardabweichungen von jeweils 25 Startpunktmengen)

Das ausführliche Ergebnis für alle Punktverhältnisse nach 100 Iterationen zeigt Abbildung 7-9, dargestellt sind wiederum Mittelwerte und Standardabweichungen. Offensichtlich sinkt die Güte des Ergebnisses – die minimale Distanz – mit der Anzahl der im Inneren platzierten Punkte.

Die Farben, die sich bei einer Berechnung mit 20 Startpunkten auf dem Rand ergeben, sind in Abbildung 7-10 dargestellt. Gewählt wurden dabei solche Startpunkte, die nach 100 Iterationen des SQP-Verfahrens einen durchschnittlichen Wert der minimalen Distanz ergeben (vgl. Abbildung 7-9). Der linke Teil der Abbildung 7-10 illustriert beispielhaft die Farben nach 100 Iterationen, die erreichte minimale Distanz liegt dann bei 59,6 Längeneinheiten. Für die Konvergenz (mit einer Größenordnung des Abbruchkriteriums bei 10^{-6}) benötigt das SQP-Verfahren bei gleicher Startpunktmenge 862 Iterationen. Die Ergebnisse sind in Abbildung 7-10 rechts dargestellt.

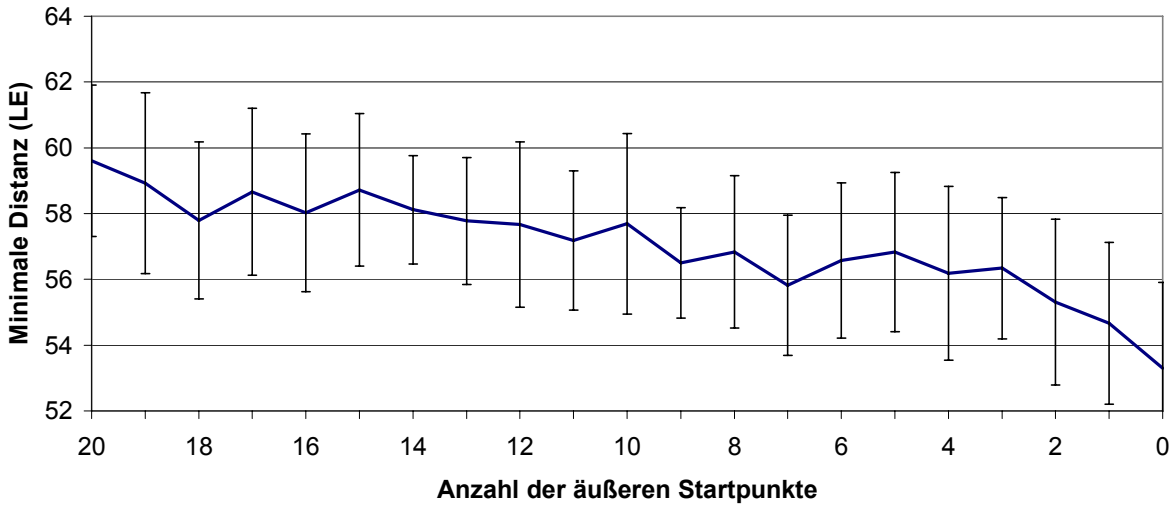


Abbildung 7-9: Minimale Distanzen für 25 Startpunktmengen des leeren Farbraums nach 100 Iterationen des SQP-Verfahrens (jeweils Mittelwert und Standardabweichung)

Die Unterschiede zwischen den Farben nach 100 Iterationen und nach Konvergenz des Verfahrens sind trotz dieses Unterschieds in der Iterationsanzahl als gering zu bezeichnen. Offensichtlich sind die Farben vom Farbton her ähnlich und unterscheiden sich primär in der Helligkeit. Dieses Ergebnis war nach der Betrachtung der Punktwanderung im Abschnitt 6.3.3 (Abbildung 6-12) zu erwarten: Nach einer gewissen Anzahl von Iterationen sind die Punkte auf dem Rand des verfügbaren Raumes – hier des Polyeders – bereits weitgehend festgelegt. Punktbewegungen sind dann nur noch in geringem Umfang auf diesem Rand möglich.

100 Iterationen								862 Iterationen (Konvergenz)							
Farbe	R	G	B	Farbe	R	G	B	Farbe	R	G	B	Farbe	R	G	B
	35	0	0		255	127	199		0	54	66		0	149	229
	106	0	0		0	154	230		107	0	61		255	149	0
	68	0	144		255	144	152		97	0	143		255	151	170
	0	0	255		255	166	0		56	0	250		93	207	214
	0	105	0		237	171	254		14	107	0		255	179	249
	0	108	122		0	255	7		136	100	0		0	255	7
	153	105	0		187	242	95		208	0	245		255	218	0
	255	1	0		255	228	0		255	1	0		0	255	182
	255	0	131		144	255	238		255	0	138		208	255	124
	239	0	251		254	255	209		255	0	210		255	252	227

Abbildung 7-10: Ergebnis der Platzierung von 20 Farben im leeren Farbraum nach 100 Iterationen (links) und nach Konvergenz des Verfahrens (rechts), jeweils geordnet nach der Helligkeit L im CIELUV-Farbraum

Wird auf die Ergebnisse der Abbildung 7-10, die ja Berechnungen nach Ausführung des SQP-Verfahrens (vgl. Beginn dieses Abschnitts) darstellen, die Berechnung der größten leeren Kugel angewandt, ergibt sich keine weitere Verbesserung. Es handelt sich also um eine gute lokale Lösung.

7.4.2 Optimale Farben am Beispiel des Stadtplanwerkes

Das Verfahren der Farbberechnung wird nun auf das Beispiel des Stadtplanwerks angewandt. Wie einleitend zu diesem Kapitel beschrieben, sei dafür die Anzahl *m* der gegebenen Farben 10, ebenso die Anzahl *n* der gesuchten Farben.

Es werden zunächst wieder Berechnungen mit verschiedenen Mengen von Startpunkten durchgeführt. Diese Mengen setzen sich – wie für den leeren Farbraum beschrieben – aus unterschiedlichen Verhältnissen von Punkten auf dem Rand und im Inneren des Farbraumpolyeders zusammen, d.h. jede Menge enthält wieder *r* Randpunkte mit *r = 10,9,...,1,0* und *n-r* Punkte im Inneren des Raumes. Für jedes Verhältnis von Punkten werden wiederum jeweils 25 Mengen betrachtet. Das Ergebnis nach 100 Iterationen des SQP-Verfahrens zeigt Abbildung 7-11 (Zur Berechnung vgl. Ausführungen zum leeren Farbraum).

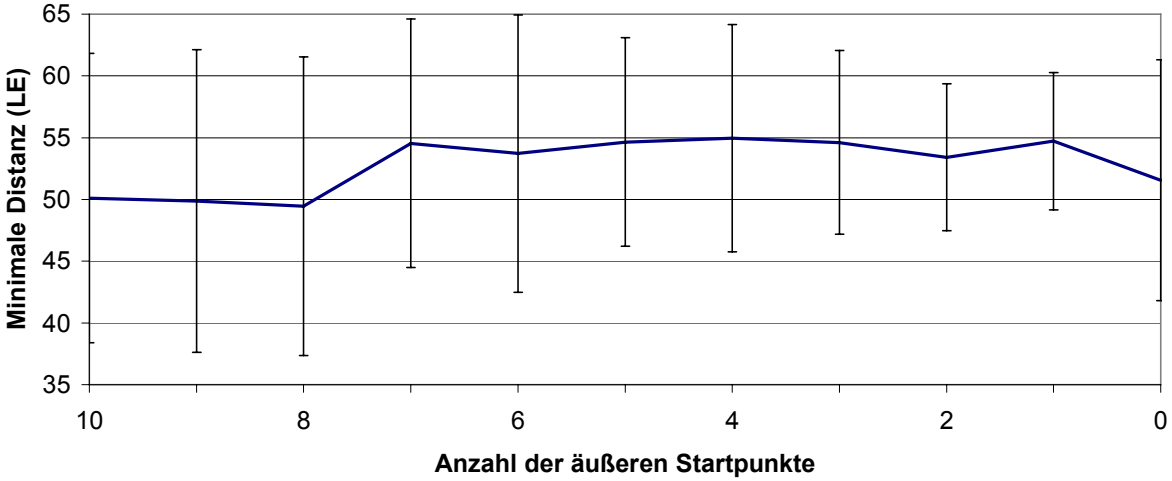


Abbildung 7-11: Minimale Distanzen für 25 Startpunktmengen der Beispielkarte nach 100 Iterationen des SQP-Verfahrens (jeweils Mittelwert und Standardabweichung)

An den Ergebnissen in Abbildung 7-11 fällt vor allem zweierlei auf:

- Die Präferenz für die Randpunkte lässt sich nicht bestätigen.
- Es ergeben sich, beispielsweise im Vergleich mit den Ergebnissen der Berechnung für den leeren Farbraum, sehr große Standardabweichungen.

Die Ergebnisse werden nun für den Fall der 10 auf dem Rand liegenden Startpunkte detailliert betrachtet. Die Werte für Mittelwert und Standardabweichung der minimalen Distanz liegen bei

$$d_{min} = 50{,}1 \text{ LE}$$

$$s = 11{,}7 \text{ LE}.$$

Die der Mittelwertberechnung zugrunde liegenden Zahlen zeigen, dass die geringste der maximierten minimalen Distanzen bei *33,0 LE,* die höchste bei *68,2 LE* liegt. Die jeweiligen Startpunkte, die zu diesen Ergebnissen geführt haben, sind in der Abbildung 7-12 links für *33 LE* und rechts für *68,2 LE* als rote Kreuze dargestellt und werden im Folgenden als „gute“ und „ungünstige“ Startpunktkonstellationen bezeichnet.

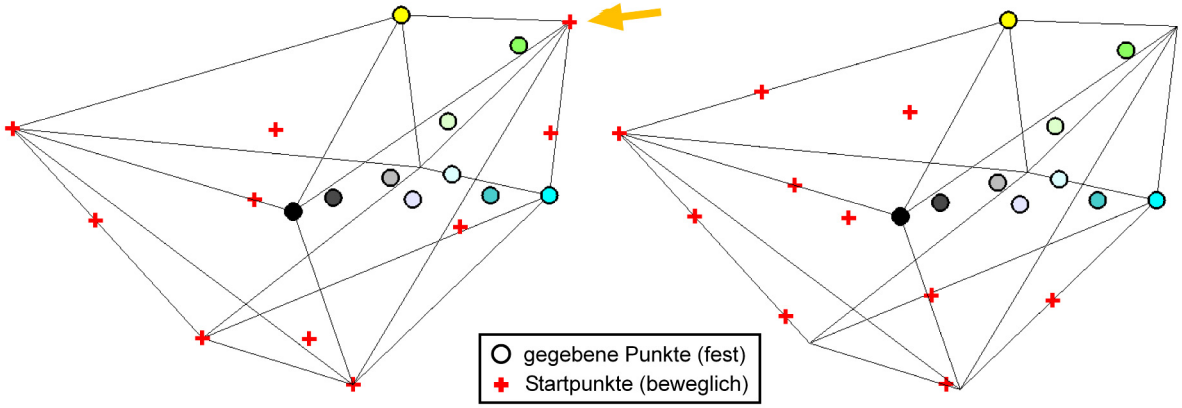


Abbildung 7-12: Farborte von Beispielkarte und Startpunkten im Farbraumpolyeder: Ungünstige (links) und gute (rechts) Startpunktkonstellation

Die ungünstige Startpunktkonstellation wird durch eine Situation verursacht, die bereits im Abschnitt 6.3.3 beschrieben wurde. Der im linken Bild der Abbildung 7-12 durch den Pfeil gekennzeichnete Startpunkt liegt in einem Eckpunkt des Farbraumpolyeders, in geringer Entfernung von einem gegebenen, festen Punkt. Die Abstandsrestriktion zwischen diesem festen Punkt und dem Startpunkt hindert letzteren daran, seinen Platz zu verlassen und in andere Bereiche des Farbraumpolyeders zu „wandern“. Dies führt dazu, dass das Optimierungsverfahren die Startpunkte lediglich geringfügig verbessert und – sobald keine Punktverschiebung mehr möglich ist – nach wenigen Iterationen abbricht. Die Punktplatzierungen des Ergebnisses entsprechen dann weitgehend denen der Startpunkte.

Es ist nun zu prüfen, ob diese ungünstige lokale Lösung durch Anwendung des Verfahrens der größten leeren Kugel detektiert und verbessert werden kann. Die Berechnung des ersten Kugelradius ergibt einen Wert von *78,2 LE*. Offensichtlich würde also eine sprunghafte Bewegung des bisher am schlechtesten platzierten Punktes in den Mittelpunkt der größten leeren Kugel eine Verbesserung ergeben. Wird diese Verschiebung durchgeführt, lassen sich nachfolgend – wie im Abschnitt 7.2 beschrieben – durch fortgesetzte Berechnung von jeweils minimaler Distanz und größter leerer Kugel weitere Verbesserungen vornehmen. Abbildung 7-13 (links) zeigt alle Punktbewegungen in der Reihenfolge ihres Auftretens. Das abschließend erhaltene Ergebnis ist aus Gründen der Übersichtlichkeit noch einmal in der Abbildung 7-13 rechts dargestellt. Die minimale Distanz eines Punktes der lokalen Lösung zu den Punkten der Gesamtmenge aus lokaler Lösung und gegebenen Punkten beträgt dann *57,9 LE*.

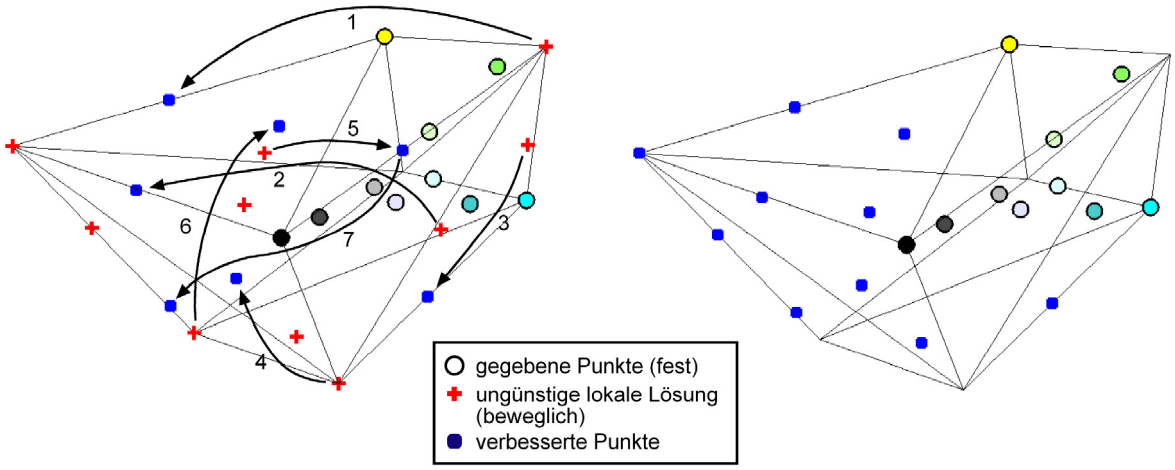


Abbildung 7-13: Verbesserung der ungünstigen lokalen Lösung für die Beispielkarte durch sprunghafte Punktbewegungen (links), die Zahlen geben die Reihenfolge der Verbesserung an; rechts ist die verbesserte Konstellation der Punkte dargestellt.

Eine Berechnung der leeren Kugel für die lokale Lösung, die nach Anwendung des SQP-Verfahrens auf eine gute Startpunktkonstellation erhalten wird, ergibt keine weitere Verbesserungsmöglichkeit. Es handelt sich bei dem Ergebnis damit um eine gute lokale Lösung.

Gegebene Farben				Neu bestimmte Farben aus ungünstigen Startpunkten				guten Startpunkten			
Farbe	R	G	B	Farbe	R	G	B	Farbe	R	G	B
	0	0	0		77	0	131		105	0	0
	74	74	74		144	0	13		84	11	244
	74	205	205		170	83	238		255	0	1
	189	189	189		255	2	1		255	0	141
	139	255	98		255	0	150		255	3	231
	0	255	255		170	126	0		0	152	229
	230	230	255		254	0	225		255	131	0
	222	255	205		0	153	230		215	143	252
	255	255	0		255	142	0		255	144	205
	222	255	255		255	156	202		255	202	148

Abbildung 7-14: Ergebnis der Platzierung von 10 Farben für die Ergänzung der Beispielkarte nach 100 Iterationen des Optimierungsverfahrens; Farben jeweils geordnet nach der Helligkeit L im CIELUV-Farbraum

Abbildung 7-14 zeigt die Optimierungsergebnisse der beiden beschriebenen Startpunktkonstellationen (verbesserte ungünstige Startpunkte und gute Startpunkte) als Farben mit zugehörigen RGB-Werten. Die Farbmengen stimmen von den Farbtönen her weitgehend überein;

sechs der Farben sind sogar soweit identisch, dass eine Unterscheidung durch das menschliche Auge nicht oder nur schwer möglich ist.

7.5 Einbeziehung weiterer Restriktionen

Im Abschnitt 6.1 wurden verschiedene Forderungen genannt, um die sich die Problemformulierung erweitert lässt.

Ein gewisser Grad an Buntheit, d.h. ein Abstand von der Unbuntgeraden, ist durch die Forderung

$$\Delta C^*_{uv} = \sqrt{(u^*_i)^2 + (v^*_i)^2} \geq r_{\min}$$

erreichbar (vgl. psychometrische Buntheit, Abschnitt 5.3.4.3). Farben, die innerhalb des Zylinders mit Radius $r_{\min}$ um die Unbuntachse liegen, werden so nicht in die Berechnung einbezogen.

Die Forderung nach der Variation der Farben in nur einer Dimension des Farbraums ist durch die Gleichsetzung in den jeweils anderen beiden Dimensionen umsetzbar.

Eine gleiche Sättigung ist mit der psychometrischen Sättigung C^*_{uv} / L^* (vgl. Abschnitt 5.3.4.3) erreichbar:

$$\frac{\sqrt{(u^*_i)^2 + (v^*_i)^2}}{L^*_i} = \frac{\sqrt{(u^*_j)^2 + (v^*_j)^2}}{L^*_j}, \quad i, j = 1, \ldots, n; i \neq j .$$

Ein konstanter Farbton wird durch die Restriktion

$$\arctan(v_i^* / u_i^*) = \arctan(v_j^* / u_j^*), \qquad i, j = 1, \ldots, n; i \neq j$$

erreicht, eine gleiche Helligkeit durch

$$L^*_i = L^*_j, \qquad i, j = 1, \ldots, n; i \neq j .$$

Falls nach der im Abschnitt 5.6.3 beschriebenen Methode Informationen über die Farbdarstellung des genutzten Wiedergabegeräts verfügbar sind, lässt sich die beschriebene Anpassung auf einen spezifischen Schwarzpunkt bzw. spezifische Gammawerte vornehmen. Diese Umrechnung kann entweder im Anschluss an die eigentliche Farbberechnung erfolgen oder teilweise in die eigentliche Optimierung integriert werden. Im letzten Fall bedeutet dies die Verkleinerung des Farbraumpolyeders.

Sehschwächen lassen sich berücksichtigen, indem die in Abschnitt 5.6.2 skizzierte Transformation in den SML-Raum vorgenommen wird. Für Protanope erfolgt die Berechnung dann beispielsweise in der Ebene, die durch Transformation der in Abbildung 5-21 dargestellten protanopen Ebene in den CIELUV-Farbraum entsteht. Aufgrund des sehr eingeschränkten Farbumfangs sind allerdings nur wenige Farben unterscheidbar. Für die Visualisierung muss

damit verstärkt auf andere Darstellungsmöglichkeiten zurückgegriffen werden (vgl. Abschnitt 5.6.3).

7.6 Bewertung der Ergebnisse

In den beiden letzten Kapiteln wurde zunächst das Farbproblem als Problem der Mathematischen Optimierung formuliert und als allgemeines nichtlineares Programm charakterisiert. Die Schwierigkeiten der Berechnung durch gängige Standardverfahren wurden verdeutlicht.

Für eine Lösung wurde nachfolgend ein Verfahren beschrieben, das mehrere Vorgehensweisen bzw. Standardverfahren integriert. Durch einen randomisierten Algorithmus wird zunächst eine Menge von Punkten auf dem Rand des Optimierungsfarbraums bestimmt. Für diese Randpunkte wurde gezeigt, dass sie sich sehr gut als Startpunkte für lokale Optimierungsverfahren eignen. Wird anschließend ein solches Verfahren (SQP-Verfahren) auf diese Startpunkte angewandt, wird eine verbesserte Platzierung im Farbraum erreicht. Da bei diesem Vorgehen sowohl gute lokal optimale Lösungen – lokale Lösungen, die einem globalen Optimum nahe kommen – als auch schlechte lokal optimale Lösungen – lokale Lösungen, die weit von einem globalen Optimum entfernt sind – auftreten können, werden letztere mit Hilfe einer geometrischen Betrachtung, die den am wenigsten dicht besetzten Ort des Farbraums findet, aufgedeckt und verbessert.

Dieser Ablauf ermöglicht für die Größenordnungen, in der sich Farben im Farbraum unterscheidbar platzieren lassen (vgl. Abschnitt 6.1) eine Berechnung des Problems on demand, d.h. in wenigen Sekunden. Dabei wird eine Lösung erhalten, die mindestens ein gutes lokales Optimum darstellt; auf die Berechnung einer Lösung, deren globale Optimalität gesichert ist, wird zugunsten der Effizienz verzichtet.

Das Verfahren wurde erfolgreich auf zwei exemplarische Farbberechnungen angewandt: Zum einen wurden 20 Farben im „leeren“ Farbraum bestimmt, zum anderen 10 Farben, die sich von bereits im Farbraum vorhandenen Farben einer Beispielkarte unterscheiden sollten. Als Ergebnis, dargestellt in den Abbildungen 7-10 und 7-14, wurden Farben erhalten, die für das normalsichtige menschliche Auge gut unterscheidbar sind. Rechnerisch beträgt der minimale Abstand im ersten Fall 59,6 LE (vgl. Ausführungen im Abschnitt 7.4.1), im zweiten Fall 57,9 LE (vgl. Ausführungen im Abschnitt 7.4.2). Damit erfüllen diese Werte den Mindestabstand von 45 Längeneinheiten, der im Abschnitt 5.5.1 als untere Grenze für eine gute Unterscheidbarkeit angegeben wurde, deutlich. Diese rechnerische Einschätzung wird durch einen visuellen Vergleich der Farben bestätigt[57].

Die erhaltenen Farben eignen sich sehr gut zur Lösung der im Abschnitt 4.4.1 beschriebenen Aufgabenstellung, der Visualisierung bzw. Integration räumlicher Objekte in einer topographischen Karte. Eine Einsatzmöglichkeit wäre beispielsweise die Differenzierung verschiede-

[57] Die visuell gute Unterscheidbarkeit ist natürlich nur unter der Voraussetzung von Darstellungsmedien gegeben, die Farben weitgehend korrekt reproduzieren, bspw. also ein kalibrierter Monitor. Für einen Leser dieser Arbeit muss die gute Unterscheidbarkeit in den Abbildungen 7-10 und 7-14 also nicht immer wahrnehmbar sein.

ner Radwege für die im Abschnitt 3.2 beschriebenen Freizeitportale (z.B. besonders ausgezeichnete Radwege).

In der Systematik der visuellen Variablen kommen die Farben durch Variation der drei Dimensionen der Farbwahrnehmung (Farbton, Helligkeit, Sättigung) zustande, nur so lässt sich die in den Beispielen geforderte hohe Anzahl an Farben bestimmen (zur Anzahl von Farben in der Visualisierung vgl. Abschnitt 5.5). Spezifischere Farbpaletten bzw. Farbskalen, die z.B. eine Ordnung repräsentieren, lassen sich durch Einbringen weiterer Gestaltungsregeln in das Optimierungsmodell berechnen. Diese Regeln gehen in Form der im Abschnitt 7.5 beschriebenen Restriktionen ein. Beispielsweise werden geordnete Farbskalen durch Variation von lediglich einer Dimension der Farbwahrnehmung erhalten. Dabei ist natürlich zu beachten, dass durch Einführung weiterer Restriktionen die verfügbare Größe des Optimierungsfarbraums eingeschränkt wird, d.h. die Anzahl der Farben, die gut unterscheidbar platziert werden können, wird geringer.

Ebenso können durch Einführung weiterer Restriktionen die Sehschwächen eines Nutzers und Anzeigeeigenschaften seines Mediums berücksichtigt werden. In ihrer Auswirkung stellen diese Restriktionen eine Anpassung des Optimierungsfarbraums durch Ausschluss nicht darstellbarer Farben dar.

8 Architektur und prototypische Umsetzung

In den Kapiteln 2 bis 7 wurden alle technischen und konzeptionellen Aspekte beschrieben, die für die ad-hoc-Erstellung personalisierter Karten benötigt werden. In einem letzten Schritt sollen diese Technologien und Konzepte nun in einer Architektur und einem automatisierten Verfahren zur Farbauswahl und Kartengestaltung gebündelt werden. Dazu wird die im Abschnitt 3.2 vorgestellte Architektur der Webportale mit Raumbezug um die erforderlichen Komponenten erweitert und eine prototypische Umsetzung aufgezeigt.

Vorab seien noch einmal explizit zwei wesentliche Rahmenbedingungen genannt, die dafür gefordert werden:

- Es sollen Infrastruktur und Technologien von Internet und WWW genutzt werden.
- Es sollen möglichst offene Standards genutzt werden. Dies gilt insbesondere für die Kommunikation und die Client-Seite.

In Abschnitt 8.1 wird zunächst die Architektur beschrieben, Abschnitt 8.2 skizziert die wesentlichen Technologien und Werkzeuge der prototypischen Umsetzung.

8.1 Architektur

Im Abschnitt 3.2 wurden Architekturen von Webportalen mit Raumbezug vorgestellt. Diese umfassten bereits einen wesentlichen Teil der im Folgenden benötigten Komponenten, insbesondere

- eine Benutzerschnittstelle zur Informationsabfrage auf Seiten des Clients (Web Map Client),
- eine Datenintegration auf Seiten des Servers,
- die Abfrage von Informationen (Geodaten und Contents) aus verschiedenen Quellen,
- Elemente einer Personalisierung.

Die Erweiterung zu einer Architektur, die das Erstellen beliebiger Karten on demand ermöglicht, muss vor allem

- die Anpassung der Personalisierungskomponente auf die Berücksichtigung von Sehschwächen und Geräteeigenschaften,
- die Erweiterung der Datenintegration um Funktionen zur Analyse und Gestaltung von Daten umfassen.

Es wird nun zunächst die Gesamtarchitektur vorgestellt, bevor die zuletzt genannten Komponenten der Personalisierung und Datenintegration einschließlich der Analyse- und Gestaltungsfunktion beschrieben werden.

8.1.1 Übersicht und Zusammenwirken

Abbildung 8-1 zeigt die Architektur zur Erstellung personalisierter Karten. Im Vergleich zum Webportal mit Raumbezug sind wesentliche Änderungen durch die Erweiterung der Datenintegration und die Einbindung der Personalisierung in den Server-Rechner innerhalb der Domain gegeben. Letzteres dient allerdings lediglich einer vereinfachten Darstellung, eine externe Personalisierung, wie im Abschnitt 3.2.3 beschrieben, wäre ebenso möglich.

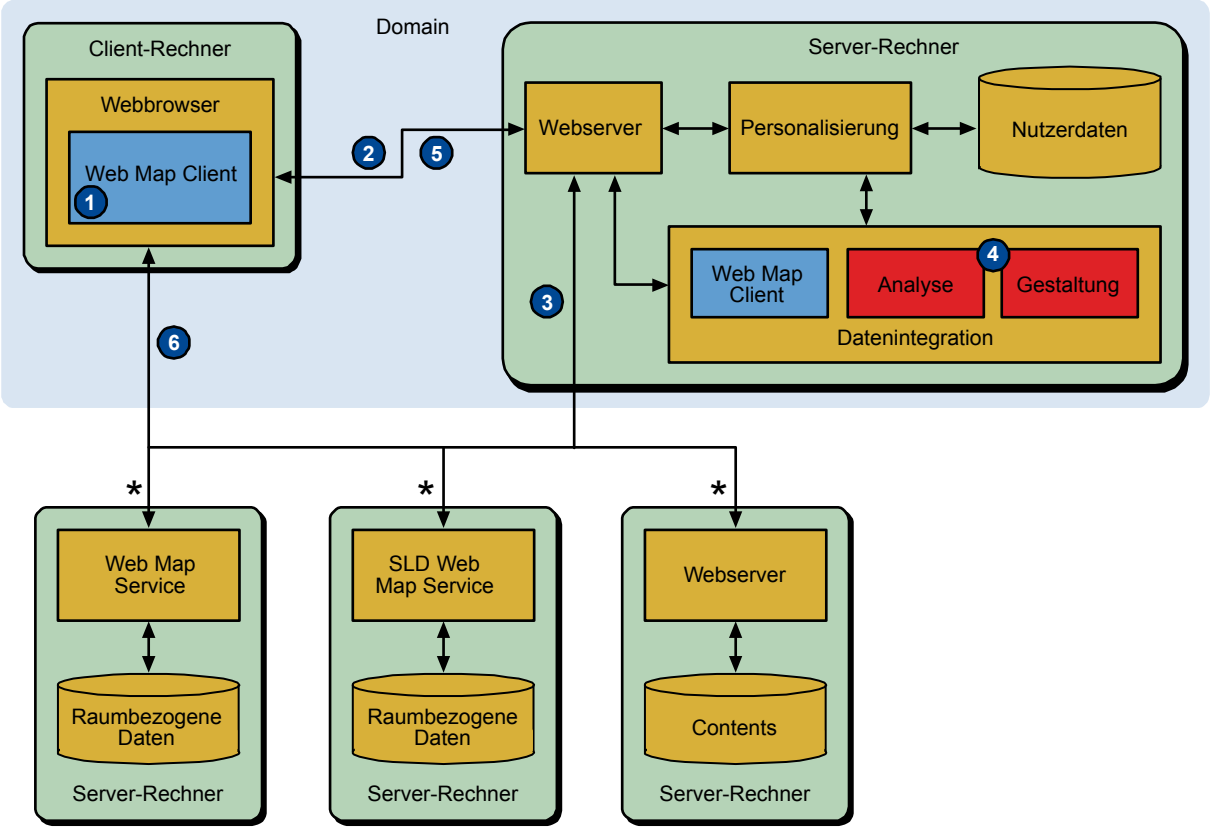


Abbildung 8-1: Architektur der automatisierten Kartenerstellung im WWW (* bezeichnet die Multiplizität „beliebig viele“, Erläuterung der Zahlen im Text)

Die für die Kartenerstellung benötigten Daten können von beliebig vielen Servern, die beispielhaft im unteren Teil der Abbildung 8-1 dargestellt sind (zur Art der Schnittstellen vgl. auch Abschnitt 2.3.2) bezogen werden:

- Daten von Web Map Services (WMS, Abschnitt 2.3.3.2), die keine Erweiterung um Styled Layer Descriptoren (SLD-WMS, Abschnitt 2.3.3.3) nutzen, sind in ihrer graphischen Darstellung von Seiten des WMS bereits festgelegt. In diesem Fall ist keine Beeinflussung der Darstellung durch SLD möglich – entweder weil dies nicht gewollt ist, oder, im Fall vorgefertigter Karten, die lediglich im Rasterformat vorliegen, technisch nicht ohne weiteres möglich ist.
- Bei einem SLD-WMS liegen die Daten als raumbezogene Objekte vor und erhalten ihre graphische Darstellung gemäß den Wünschen des Abfragenden nach dem in den Abschnitten 2.3.3.1 bzw. 4.1.1 beschriebenen Ablauf der Visualisierungspipeline.

- Contents bezeichnen die primär informierenden Daten (Texte, Bilder) und sind durch Koordinaten räumlich einzuordnen (Abschnitt 2.3.2). Die graphische Repräsentation richtet sich nach der Art der Koordinaten (punkt-, linien- oder flächenhaft) und ist unabhängig von der abgefragten Schnittstelle.

Die einzelnen Schritte einer Kartenerstellung lassen sich gemäß Abbildung 8-1 wie folgt zusammenfassen (eine vertiefende Beschreibung der Personalisierung, Analyse und Gestaltung erfolgt im Anschluss): Ein Nutzer gibt in seinen Web Map Client die URLs ein, von denen er Daten beziehen möchte (1). Diese Daten werden an den Web Server gesandt (2), der wiederum die betreffenden Services anfragt und die zur Visualisierung benötigten Daten abruft (3). Im Fall eines WMS sind dies beispielsweise GetCapabilities-Dokumente oder auch Beispieldaten. Diese Informationen werden zunächst auf die graphischen Darstellungen der angebotenen Daten hin analysiert und aus deren Gesamtschau mögliche Gestaltungsvorschriften generiert (4). Diese Vorschriften können dann entweder auf den Client-Rechner übertragen (5) und auf die von dort abgefragten Daten angewandt werden (6) oder auf dem Server-Rechner verbleiben. In diesem Fall müssen die Datenabfragen des Clients über diesen Rechner erfolgen. Detaillierte Ausführungen zur Gestaltung werden im Abschnitt 8.1.2.2 gegeben.

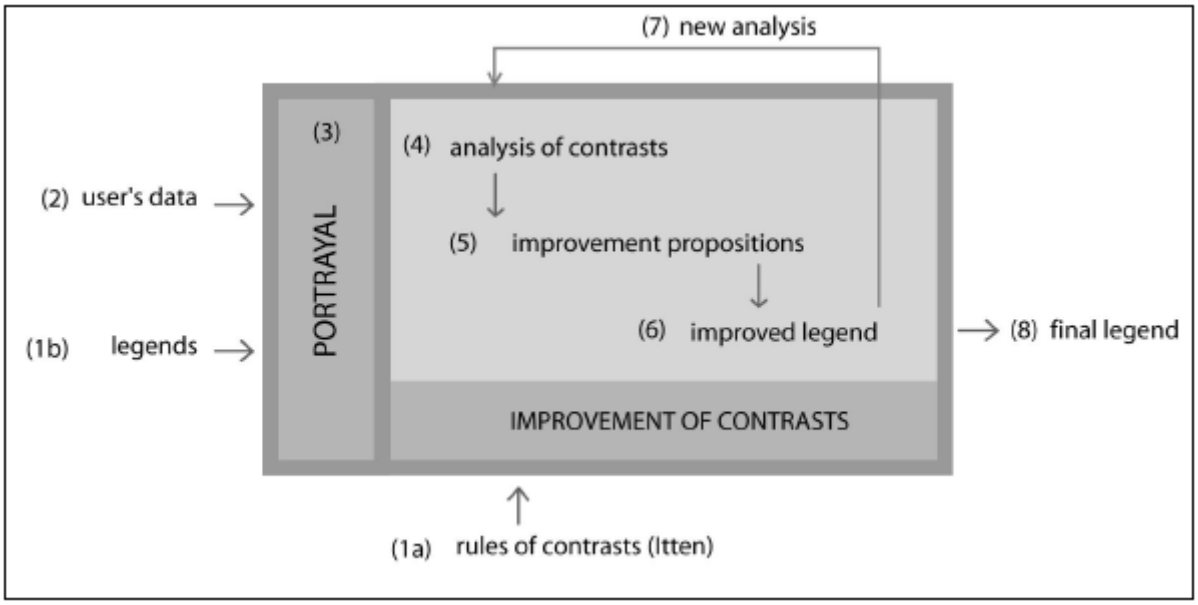


Abbildung 8-2: Ablauf bei der Bestimmung kontrastreicher Karten nach Chesneau et al. (Quelle: Chesneau et al. 2005)

Einen ähnlichen Ablauf beschreiben auch Chesneau et al. (2005), deren Arbeit bereits im Abschnitt 6.7.1 für die Gestaltung von Risikokarten genannt wurde (Abbildung 8-2). Allerdings wird dort keinerlei technische Einbettung angegeben.

8.1.2 Komponenten

Nachdem die Architektur und der Ablauf einer Kartenanfrage vorgestellt wurden, sind nun noch die Funktionen von Personalisierung, Analyse und Gestaltung zu vertiefen.

8.1.2.1 Personalisierung

Die Personalisierungskomponente stellt wesentliche Funktionen zur Erstellung, Verwaltung und Vorhaltung von Nutzerprofilen zur Verfügung. Ein solches Profil muss dabei neben obligatorischen Informationen wie Benutzername, E-Mail-Adresse und Passwort auch folgende Daten umfassen:

- Eine eventuelle *Sehschwäche*, die zum Benutzer in einer 1:1 –Beziehung steht.
- *Geräteprofile* sind einem Nutzer in einer 1 : n – Beziehung zugeordnet, d.h. ein Nutzer hat die Möglichkeit, Darstellungseigenschaften verschiedener genutzter Geräte abzuspeichern.

Weiterhin muss die Personalisierung bieten:

- *Anmeldefunktion*: Ein Nutzer gibt auf einer Webseite seine Daten (mindestens einen Nutzernamen) ein. Das System empfängt diese Daten, sendet dem Nutzer einen Aktivierungslink und ein Passwort. Nach Anklicken des Links und Eingabe des Passworts wird ein Nutzerprofil eingerichtet.
- *Nutzeranalyse*: Ein Nutzer muss in seinem Profil Angaben über seine Sehschwächen machen oder – falls diese nicht bekannt sind – Tests nach der in Abschnitt 5.6.3 skizzierten Methode durchführen.
- *Geräteanalyse*: Fügt ein Nutzer seinem Profil ein neues Gerät hinzu, muss er die in Abschnitt 5.6.3 beschriebenen Handlungen zur annähernden Erfassung der Geräteeigenschaften durchführen.
- *Speicherfunktion*: Das System muss die jeweiligen Nutzerdaten speichern.

Für die Nutzung der personalisierten Informationen ist der Zugriff darauf von besonderer Bedeutung: Die Daten müssen einerseits sicher vor unbefugtem Zugriff, andererseits aber möglichst ohne große Hürden für die Kartenerstellung verfügbar sein. Dies kann in einem zweistufigen System erfolgen, wie es beispielsweise viele Online-Shops benutzen. Durch dauerhaftes Hinterlegen kleiner Informationseinheiten im Browser eines Anwenders kann der aufgerufene Server diesen Anwender direkt identifizieren, mit den benötigten personalisierten Daten verknüpfen und diese Daten auf eine Kartendarstellung anwenden. Der explizite lesende und schreibende Zugriff von einem Client-Rechner ist dagegen erst nach einem Login mit Benutzernamen und Passwort möglich.

8.1.2.2 Analyse und Gestaltung

Die Analyse dient der Feststellung vorhandener graphischer Darstellungen und Darstellungsmöglichkeiten von Daten; die Gestaltung stimmt diese dann aufeinander ab.

Im Falle der raumbezogenen Daten beginnt die Analyse mit der Abfrage und Auswertung der Capabilities-Dokumente. Die wesentliche Unterscheidung der WMS ist durch eine Einteilung nach ihrer SLD-Fähigkeit gegeben (vgl. auch Architektur im vorangehenden Abschnitt):

- Für Daten, die von WMS ohne SLD-Unterstützung bezogen werden, sind anhand der Capabilities-Dokumente lediglich sehr eingeschränkte Informationen über die graphi-

schen Darstellungen verfügbar, weiterhin sind die Ausgabeformate meist rasterbasiert. Weitere Informationen können allerdings aus der Abfrage von Probedaten und deren Analyse durch Bildbearbeitungsprogramme gewonnen werden. Als Ergebnis dieser Analyse müssen mindestens die Farben der dargestellten Daten erhalten werden, wünschenswert ist darüber hinaus die Information, ob es sich um komplette vorgefertigte Karten oder lediglich um punkt-, linien- oder flächenhafte Objekte handelt.

- Gemäß der Beschreibungen im Abschnitt 2.3.3.3 enthält das Capabilities-Dokument eines WMS, der den SLD-Standard unterstützt, diejenigen Informationen, die für die Beeinflussung der Darstellung von außen notwendig sind. Allerdings sind dort keine Hinweise auf die konkrete graphische Darstellung (die Darstellung, die ein Dienst standardmäßig anbietet) von Daten enthalten. Diese muss ebenfalls wieder über die Abfrage und Analyse von Probedaten ermittelt werden.
- Wie bereits angerissen, ist die Darstellung der Contents in der Karte völlig unabhängig von der genutzten Schnittstelle und damit frei wählbar.

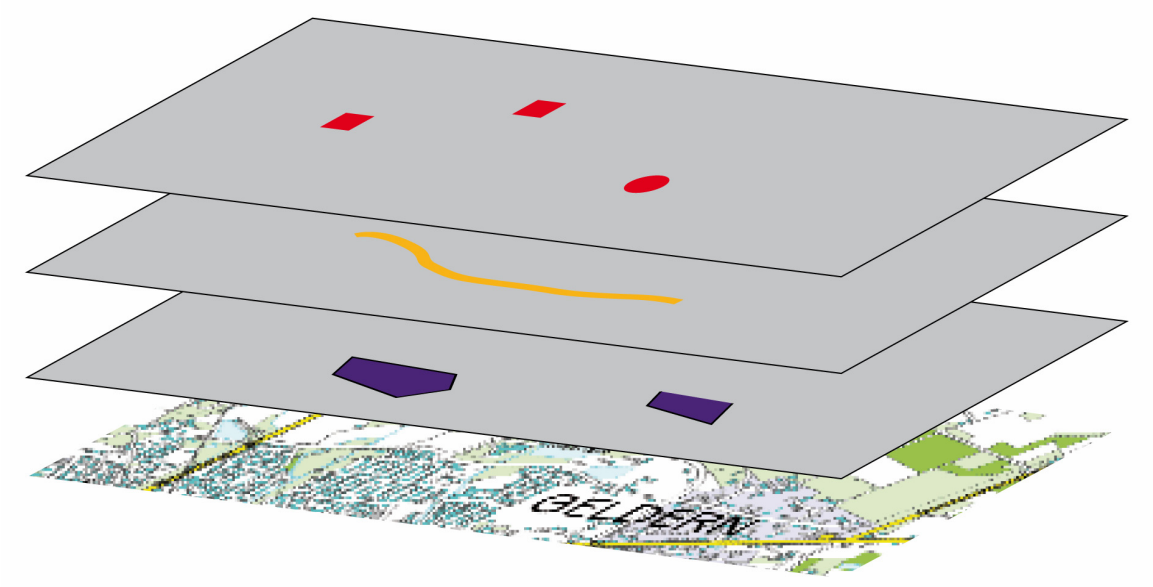


Abbildung 8-3: Integration von punkt-, linien- und flächenhaften Objekten in die Beispielkarte

Nach Analyse der Daten und ihrer vorliegenden bzw. möglichen Darstellungsform, müssen die benötigten graphischen Umsetzungen bestimmt werden. Dazu wird gemäß Abbildung 8-3 von einem ebenenbasierten Aufbau des Kartenbildes ausgegangen.

Die unterste Ebene ist durch vorgefertigte Karten gegeben, darüber werden in dieser Reihenfolge flächen-, linien- und punkthafte Objekte dargestellt. Dabei bedeutet die Integration in einem Kartenbild nicht, dass jederzeit alle Daten tatsächlich angezeigt werden, sondern alle Ebenen können von einem Nutzer dynamisch ein- und ausgeblendet werden. Damit dies möglich ist, findet die Überlagerung im Client statt.

Das Vorgehen bei der Anwendung der im Abschnitt 4.2 vorgestellten graphischen Variablen bzw. der im Abschnitt 4.4.2 zusammengefassten Darstellungsmöglichkeiten für diese Arbeit zeigt Abbildung 8-4 beispielhaft für punkthafte Objekte. Die Darstellung linien- und flächen-

hafter Objekte verläuft unter Anwendung der im Abschnitt 4.4.2 genannten Darstellungsmöglichkeiten ähnlich.

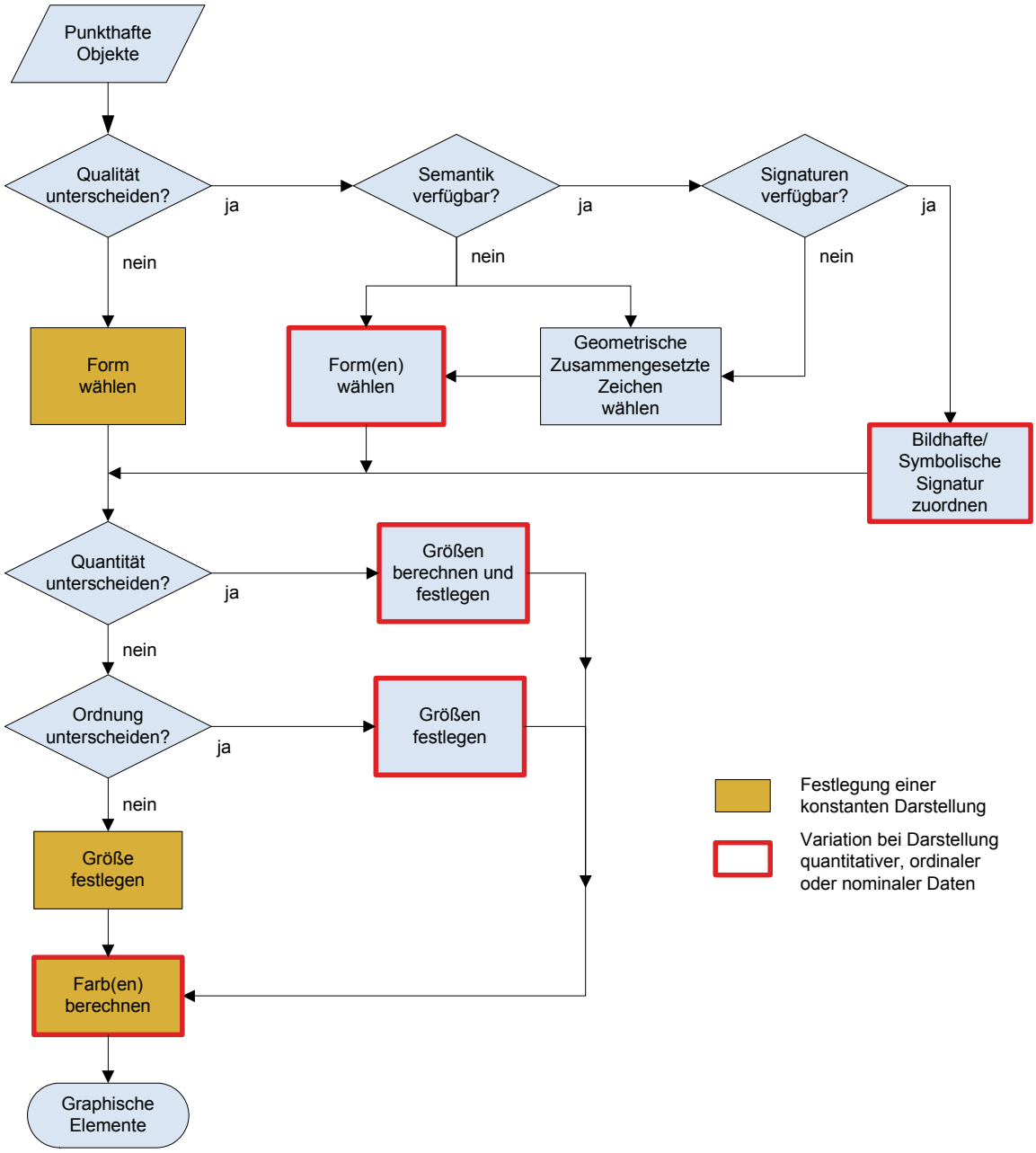


Abbildung 8-4: Ablauf der Anwendung der graphischen Variablen auf punkthafte Objekte

Der wesentliche Schritt der Gestaltung ist die Berechnung einer Farbpalette, die es erlaubt, alle darzustellenden Daten visuell zu unterscheiden. Diese Berechnung setzt das im Kapitel 7 beschriebene Vorgehen um. An dieser Stelle werden auch die personalisierten Informationen eines Nutzers einbezogen.

Die Berücksichtigung bzw. Anwendung der ermittelten Gestaltungsvorschriften ist im Wesentlichen abhängig von der genutzten Schnittstelle:

- Im Fall der SLD-WMS werden die Gestaltungsvorschriften in Symbology-Encoding-Dokumente (SE, vgl. Abschnitt 2.3.3.4) umgesetzt und als Datei auf dem Server gespeichert. Die URL der Datei wird dem Client übermittelt. Dieser bezieht nun die Daten durch direkte Anfrage der SLD-WMS unter Nutzung des SE-Dokuments.
- Für WMS, die kein SLD unterstützen, ist die Situation komplizierter. In diesem Fall ist eine direkte Abfrage von Seiten des Clients nur möglich, falls die standardmäßig durch einen WMS angebotenen Darstellungen unverändert übernommen werden können. Ist dies nicht möglich, müssen alle Anfragen über den Server-Rechner erfolgen. Dort werden die Gestaltungsvorschriften dann unter Nutzung von Methoden der Bildbearbeitung auf die Daten angewandt.
- Die graphische Repräsentation der Contents erfolgt im Client durch die Platzierung geometrischer oder symbolischer Bilder in der Karte. Diese Bilder müssen gemäß der Gestaltungsvorschriften auf dem Server-Rechner erstellt und vorgehalten werden.

8.2 Prototypische Umsetzung

Eine prototypische Umsetzung ist mit den in Kapitel 3 beschriebenen Techniken erfolgt. Basis der Implementierung ist das Open Source Content Management System TYPO3[58], das sich sehr gut für alle beschriebenen Anforderungen eignet. Den Aufbau und wesentliche genutzte Techniken gibt Abbildung 8-5 wieder.

TYPO3 ist in der Programmiersprache PHP[59] implementiert und nutzt standardmäßig den Webserver Apache[60] und die Datenbank MySQL[61], beides Open Source Produkte. Als alternativer Webserver ist der Microsoft IIS[62] einsetzbar, über eine Datenbankabstraktionsschicht lassen sich auch Datenbanken wie Oracle[63] oder Postgres[64] einbinden. Weiterhin integriert TYPO3 mit ImageMagick[65] ein Open Source Bildbearbeitungsprogramm.

Die besondere Eignung von TYPO3 für Anwendungen jeglicher Art liegt im flexiblen Systementwurf begründet (vgl. Abbildung 8-5). Das System nutzt einen Kern, der lediglich grundlegende Funktionen zur Datenbank-, Datei- und Benutzerverwaltung enthält. Alle anderen Funktionen sind durch sogenannte Extensions realisiert. Diese nach einem definierten Schema erstellten Erweiterungen werden über eine Schnittstelle (Extension API) in das System integriert. Innerhalb dieses definierten Schemas lassen sich beliebige benutzerdefinierte Funktionen umsetzen. Besonderer Vorteil ist die problemlose Austauschbarkeit und Einbin-

[58] http://typo3.org (Zuletzt geprüft am 08.01.2009)

[59] http://www.php.net (Zuletzt geprüft am 08.01.2009)

[60] http://www.apache.org (Zuletzt geprüft am 08.01.2009)

[61] http://www.mysql.com (Zuletzt geprüft am 08.01.2009)

[62] http://www.iis.net (Zuletzt geprüft am 08.01.2009)

[63] http://www.oracle.com (Zuletzt geprüft am 08.01.2009)

[64] http://www.postgresql.org (Zuletzt geprüft am 08.01.2009)

[65] http://www.imagemagick.org (Zuletzt geprüft am 08.01.2009)

dung der Extensions: Jeder Entwickler hat die Möglichkeit, seine Erweiterungen auf http://typo3.org allgemein zur Verfügung zu stellen. Dementsprechend findet sich dort eine Vielzahl von Extensions für die unterschiedlichsten Aufgaben.

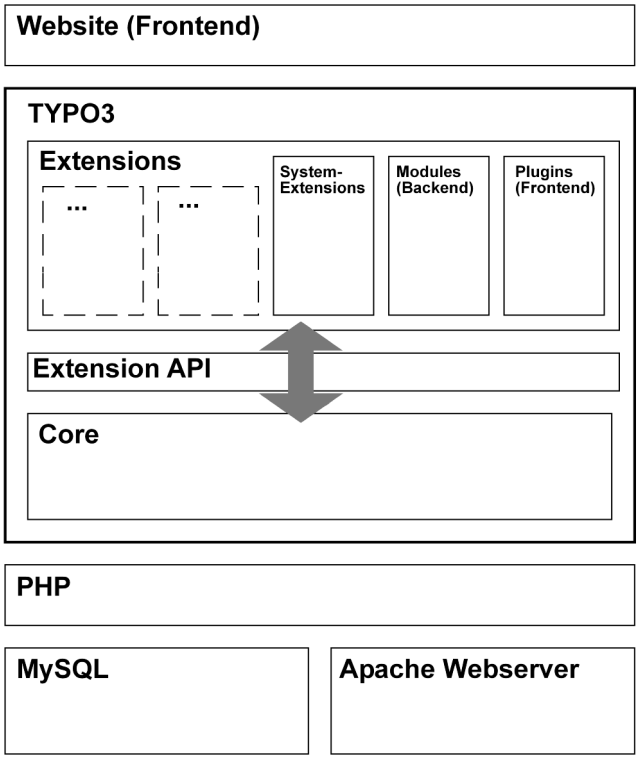


Abbildung 8-5: Systemaufbau des Content Management Systems TYPO3 (nach Laborenz et al. 2006)

Die Ausgabe auf Seiten des Frontends ist beliebig konfigurierbar und kann von einfachem (X)HTML bis hin zu DHTML bzw. AJAX reichen.

Damit bündelt TYPO3 alle für eine Umsetzung benötigten Werkzeuge und macht sie über definierte Schnittstellen nutzbar. Weiterhin sind bereits Funktionen verfügbar, die eine Umsetzung sehr unterstützen (z.B. eine fortgeschrittene Nutzerverwaltung).

Für das Mapping on Demand wurden als wesentliche Erweiterungen umgesetzt:

- Ein DHTML- bzw. AJAX-Client erlaubt dem Nutzer zunächst die Eingabe gewünschter Datenquellen. Der Client kommuniziert mit einer Gegenseite auf dem Webserver und tauscht mit diesem die Datenquellen bzw. im Gegenzug die Gestaltungsvorschriften aus.
- Die Datenintegration auf Seiten des Webservers stellt die im vorangehenden Abschnitt beschriebenen Analysefunktionen bereit, indem Services abgefragt, Capabilities-Dokumente ausgewertet und Probedaten mit Hilfe von ImageMagick analysiert werden. Alle Informationen werden zur schnelleren Verfügbarkeit in der Datenbank gespeichert und sporadisch auf Aktualität überprüft. Weiterhin besitzt diese Erweiterung eine Schnittstelle zur Gestaltungsfunktion, die außerhalb von TYPO3 umgesetzt wurde (siehe unten).

- Die vom System bereitgestellten Möglichkeiten der Benutzerverwaltung wurden um die beschriebene Vorhaltung von Daten zur Sehschwäche und zu Geräteeigenschaften erweitert. Zur Erfassung dieser Daten wurden einfache Tests integriert (vgl. Abschnitt 5.6.3).

Die im Abschnitt 8.1.2.2 skizzierte Gestaltungsfunktion wurde in einem eigenständigen Werkzeug in der Programmiersprache C umgesetzt. Kern ist die Methode zur Berechnung der gut unterscheidbaren Farben. Dazu integriert die Funktion verschiedene andere Werkzeuge. Das Programm „Qhull“[66] dient der Berechnung von Konvexen Hüllen, Voronoi-Diagrammen, Delauny-Triangulationen und Halbraumschnitten (vgl. auch Abschnitt 6.2.2.2); das Programm ist ebenfalls in der Programmiersprache C umgesetzt. Als lokales Optimierungsverfahren ist das SQP-Verfahren der Funktion „*fmincon*“ des Programms MATLAB eingebunden. Der Aufruf erfolgt über die C-Schnittstelle von MATLAB. Der Algorithmus zur Bestimmung von Startpunkten ist ebenfalls in MATLAB umgesetzt. Der Aufruf der gesamten Gestaltungsfunktion erfolgt in Form eines Web Services.

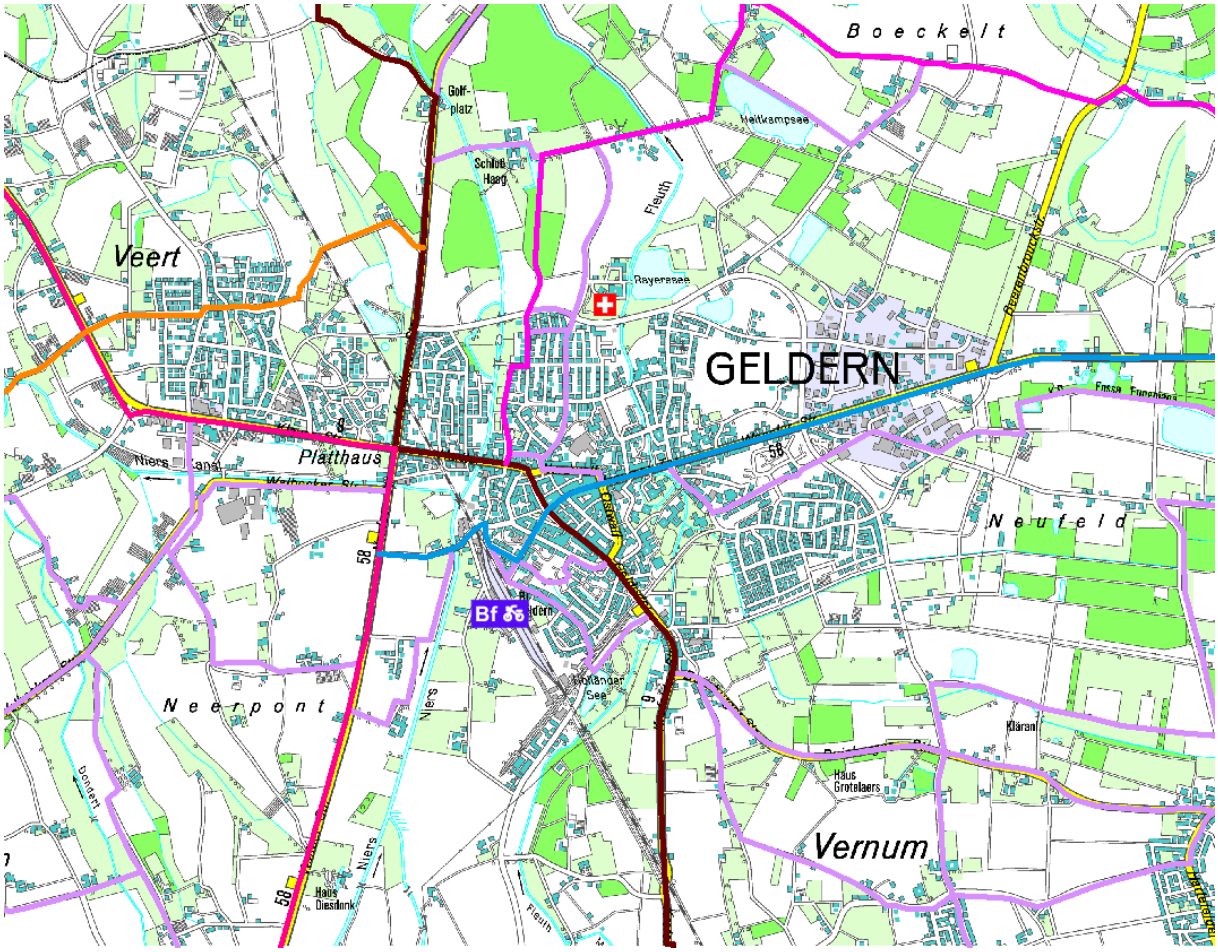


Abbildung 8-6: Ergebnis einer ad-hoc-Kartenerstellung mit Hilfe des Prototypen; die Beispielkarte ist durch punkt- und linienhafte Objekte ergänzt

[66] http://www.qhull.org (Zuletzt geprüft am 08.01.2009)

Ein Ergebnis einer Kartenerstellung mit Hilfe des Prototypen ist in Abbildung 8-6 dargestellt. Die Beispielkarte, die bereits im letzten Kapitel genutzt wurde, ist darin mit punkt- und linienhaften Objekten überlagert. Durch punkthafte Symbole sind Bahnhof und Krankenhaus repräsentiert, die Linien stellen ein Radwegenetz mit thematisch ausgezeichneten Routen dar.

9 Zusammenfassung und Ausblick

Ziel dieser Arbeit ist es, personalisierte Karten mit einem hohen Informationsgehalt so zu erstellen, dass wesentliche Informationen von einem Nutzer möglichst effektiv erfasst werden können. Die dafür erforderlichen Daten werden bei Bedarf (on demand) gesucht, zusammengeführt und in eine gemeinsame Darstellung integriert.

Zum Erreichen dieses Ziels wurden im Abschnitt 1.3 sechs Arbeitshypothesen formuliert, deren wesentlichen Zusammenhänge in Abbildung 1-1 dargestellt wurden; Abbildung 9-1 greift diese Darstellung noch einmal auf.

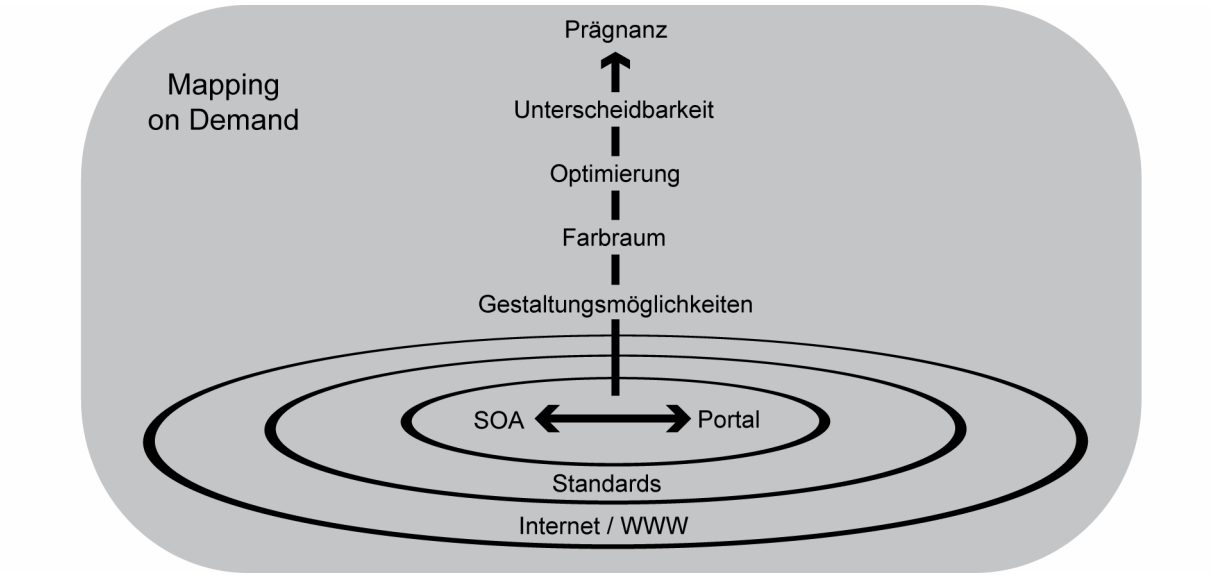


Abbildung 9-1: Übersicht über die wesentlichen Bereiche und Zusammenhänge dieser Arbeit (aus Kapitel 1)

Aus der Abbildung wird deutlich, dass in dieser Arbeit zwei Schwerpunkte von Bedeutung sind: Für die Beschaffung und Integration von Daten wird ein breites Spektrum von Technologien und Konzepten aus Internet und WWW benötigt. Auf dieser Basis erfolgt die Erstellung von ad-hoc-Karten. Für das Erreichen einer prägnanten Darstellung sind vertiefende Betrachtungen von Gestaltungsregeln, der Nutzung von Farbe und der Formulierung und Lösung von Optimierungsproblemen erforderlich.

Die Arbeitshypothesen wurden im Verlauf dieser Arbeit verifiziert; im Einzelnen waren dies:

Die *Verfügbarkeit der für ein Mapping on demand erforderlichen Technologien* wurde durch die Beschreibung wesentlicher Grundlagen und Technologien von Internet und WWW gezeigt. Dabei wurde zunächst deutlich, dass mit dem Internet und dem WWW eine Infrastruktur existiert, die eine physikalische Zusammenführung und Verknüpfung weltweit verteilter Daten ermöglicht. Der Nutzung dieser Infrastruktur kommt besonders die derzeitige Entwicklung des Internets von einer reinen Client-Server-Architektur hin zu einer Service-orientierten

Architektur in Form von Web Services entgegen: Daten und Anwendungen werden gekapselt und in Form von Diensten, die plattformunabhängig über veröffentlichte Schnittstellen zugänglich sind, bereitgestellt. Interoperabilität und damit eine allgemeine Nutzbarkeit wird mit der Beschreibung der Schnittstellen durch offene Standards erreicht. Ebenso sind die gebräuchlichsten Formate des WWW, die dem Austausch und der Darstellung dienen, in offenen Standards festgeschrieben.

Der konkrete, *homogene Zugriff auf raumbezogene Daten einer Service-orientierten Architektur wurde durch die Beschreibung verfügbarer Standards und die Bündelung und Erschließung von Diensten in Portalen* aufgezeigt. Als wichtigste Standards wurden Empfehlungen des World Wide Web Consortiums zum Austausch (Extensible Markup Language, XML) und zur Darstellung (z.B. Hypertext Markup Language, Cascading Style Sheets) von Daten, sowie Spezifikationen des Open Geospatial Consortiums zum Zugriff (Web Map Service, Styled Layer Descriptor Profile) und zur erweiterten Nutzung (Symbology Encoding, Filter) von Schnittstellen vorgestellt.

Das Konzept des Portals ermöglicht potentiellen Nutzern den strukturierten Zugang zu verfügbaren Diensten und Daten. In allgemeiner Form eines Webportals werden eine Vielzahl allgemeiner Funktionen und Dienstleistungen (z.B. Chaträume, E-Mail) angeboten; die spezielleren Formen des Geoportals und des Webportals mit Raumbezug fokussieren Dienste zum Zugriff auf raumbezogene Daten. Das Webportal mit Raumbezug wurde anhand dreier Beispiele interaktiver Freizeitplaner illustriert, die seit längerem erfolgreich über das WWW zugänglich sind. Aus diesen Beispielen ging besonders die Umsetzung einer Integration von Daten verteilter Quellen in einer Karte hervor.

Die *graphische Darstellung von Daten wurde durch deren Abbildung auf graphische Elemente und visuelle Variablen beschrieben und daraus die Farbe als vorrangige Variable für eine prägnante Darstellung* hergeleitet. Ziel einer kartographischen Gestaltung ist es, durch eine prägnante Darstellung die in einer Karte enthaltenen Informationen effizient zu kommunizieren. Dafür ist ein umfangreiches Regelwerk verfügbar, das im Kern auf der Kodierung von Daten durch graphische Elemente und visuelle Variablen beruht. In ihrer Anwendung sind diese Variablen in unterschiedlicher Weise für die Umsetzung einer prägnanten graphischen Darstellung geeignet. Aus einer Gesamtschau verschiedener Arbeiten, die sich mit dieser Eignung befasst haben, wurde deutlich, dass aus der Menge der Variablen die Farbe für eine effiziente Kommunikation von primärer Bedeutung ist.

Die *Umsetzung einer prägnanten Darstellung wurde auf die Nutzung visuell gut unterscheidbarer Farben zurückgeführt und die Bestimmung solcher Farben als Distanzproblem im dreidimensionalen Farbraum* formuliert. Die wesentliche Voraussetzung für eine prägnante Darstellung ist die Forderung nach einer eindeutigen visuellen Differenzierbarkeit der dargestellten Objekte. Im Fall der Farbnutzung bedeutet dies die Verwendung von Farben, die für das menschliche Auge möglichst gut unterscheidbar sind. Farbräume, die eine auf das visuelle System des Menschen abgestimmte Metrik bieten und die Modellierung der guten Unterscheidbarkeit ermöglichen, sind mit dem CIELUV- und dem CIELAB-Farbraum verfügbar.

Die Bestimmung der benötigten Farben erfolgt durch die Formulierung eines Distanzproblems im CIELUV-Raum. Dabei werden *n* Farben derart bestimmt, dass diese Farben sowohl untereinander als auch gegenüber *m* gegebenen Farben einen möglichst großen Abstand besitzen. Ansätze bzw. Verfahren zur Lösung von Problemen dieser Art sind im Bereich der Mathematischen Optimierung und der Algorithmischen Geometrie verfügbar; erstere bietet auch einen geeigneten Formalismus für die Modellierung und Charakterisierung des Problems. In diesem Kontext sind Distanzprobleme der Klasse der nichtlinearen Programme zuzuordnen. Diese Programme sind allgemein dadurch gekennzeichnet, dass es neben einer global optimalen Lösung auch lokal optimale Lösungen gibt; für Distanzprobleme erwachsen eine Vielzahl lokaler Lösungen aus der Kombinatorik und Symmetrie möglicher Lösungen. Eine Berechnung durch Standardverfahren der lokalen Optimierung ist effizient durchführbar, allerdings wird eine dem gewählten Startpunkt nahe liegende Lösung gefunden, deren globale Optimalität nicht gesichert ist. Verfügbare Verfahren der globalen Optimierung, die die Bestimmung einer solchen global optimalen Lösung zum Ziel haben, sind in der Anwendung auf das Farbproblem NP-vollständig und ermöglichen keine effiziente Lösung.

Zur Überwindung dieser Schwierigkeiten wird in dieser Arbeit ein Verfahren entwickelt, das mehrere Lösungsverfahren und –paradigmen integriert. Durch einen randomisierten Algorithmus wird zunächst eine Menge von Punkten auf dem Rand des Optimierungsfarbraums bestimmt. Diese Randpunkte eignen sich sehr gut als Startpunkte für Lösungsverfahren der lokalen Optimierung. Durch das SQP-Verfahren, ein lokales Verfahren, das sehr gut zur Lösung allgemeiner nichtlinearer Probleme nutzbar ist, wird eine verbesserte Platzierung der Randpunkte im Farbraum erreicht. Da bei diesem Vorgehen sowohl gute lokal optimale Lösungen – lokale Lösungen, die einem globalen Optimum nahe kommen – als auch schlechte lokale Lösungen – lokale Lösungen, die weit von einem globalen Optimum entfernt sind – auftreten können, werden letztere mit Hilfe einer geometrischen Betrachtung auf Basis des Voronoi-Diagramms, die den am wenigsten dicht besetzten Ort des Farbraums findet, aufgedeckt und verbessert. Dieses beschriebene Vorgehen ermöglicht für die Größenordnungen, in denen das Farbproblem gemäß der Ziele dieser Arbeit zu lösen ist, eine Berechnung des Problems on demand, d.h. in wenigen Sekunden.

Eine weitere *Steigerung der Prägnanz wurde durch eine Personalisierung, d.h. die Anpassung der Farbgebung an individuelle Seheigenschaften eines Nutzers, Eigenschaften seines Anzeigegeräts und äußere Einflüsse* erreicht. Diese Steigerung folgt unmittelbar aus der Verwendung der Farbe als wichtigstem Gestaltungsmittel: Die Effizienz des Farbeinsatzes wird wesentlich von den genannten spezifischen Möglichkeiten und Eigenschaften auf Seiten des Nutzers beeinflusst, der farbtreuen Wiedergabe durch das Anzeigegerät, den herrschenden Umgebungsbedingungen und Farbsehschwächen des Nutzers. Die Berücksichtigung dieser Faktoren sichert die Verwendung solcher Farben, deren Wiedergabe und Wahrnehmbarkeit gewährleistet ist. Werden diese Einflüsse durch geeignete Werkzeuge und Tests über das WWW erfasst, lassen sie sich in Form von Restriktionen, die den Optimierungsfarbraum beschneiden, in das Optimierungsmodell und damit in die Kartengestaltung einbringen.

Die beschriebene *Bestimmung der Farben ist in ein allgemeines Verfahren überführbar, das abhängig von Nutzer, Anzeigegerät und Umgebungssituation ist, aber unabhängig vom genutzten Farbraum oder einer spezifischen Anwendung*. Mit der Personalisierung fließen subjektive Randbedingungen in die Bestimmung der Farben ein. Diese Randbedingungen lassen sich durch Transformationen in einen geeigneten Farbraum, in dem das beschriebene Lösungsverfahren angewendet werden kann, objektivieren. Die Problemformulierung und dessen Lösung sind dabei auf beliebige Farbräume, die eine auf das visuelle System des Menschen abgestimmte Metrik bieten, übertragbar. Für die Nutzung des Startpunktverfahrens muss als Randbedingung die Modellierung des Farbraums durch eine konvexe Oberfläche, idealerweise ein konvexes Polyeder, möglich sein. Der weitere Verlauf des Verfahrens ist unabhängig von der Form des Optimierungsfarbraums.

Die Allgemeingültigkeit des Verfahrens beinhaltet ebenso die Möglichkeit, spezifischere Anforderungen an die gesuchten Farben zu stellen. Die im Kapitel 7 genutzte Modellierung, die die Maximierung des Minimums der Abstände zwischen allen gegebenen und gesuchten Farben betrachtete, ergab Lösungen, die in der Systematik der visuellen Variablen durch Variation aller drei Dimensionen der Farbwahrnehmung (Farbton, Helligkeit, Sättigung) zustande kamen. Diese Farben ermöglichen die Darstellung von Qualitäten und eignen sich damit sehr gut zur Repräsentation räumlicher Objekte. Das genutzte Optimierungsmodell ist allerdings so flexibel, dass auch spezifischere Gestaltungsvorschriften, die aus der Theorie der Anwendung der visuellen Variablen folgen, in Form von Restriktionen in das Modell integrierbar sind. Insbesondere sind so auch Farbskalen zur Darstellung von Ordnungen oder Quantitäten bestimmbar.

Weiterhin ist das Verfahren nicht auf den Einsatz für die Erstellung von Karten beschränkt, sondern auf Anwendungen, die eine Farbdarstellung auf digitalen Geräten unter unterschiedlichen Umgebungsbedingungen erfordern, übertragbar. Als häufig auftretendes Beispiel sei die Darstellung von Farben in Vortragsfolien über Beamer genannt: Folien, die auf einem Notebook oder Desktop-Rechner erstellt wurden, sind in ihren Farben auf die dabei genutzten Bildschirme abgestimmt; bei einer Wiedergabe über beliebige, nicht kalibrierte, Beamer sind diese Farben dann meist schlecht oder gar nicht erkennbar. Mit dem in dieser Arbeit beschriebenen Verfahren ließe sich die jeweilige Farbdarstellung ad hoc, d.h. auch noch kurz vor einem Vortrag, auf die jeweilige Situation im Vortragsraum abstimmen.

Das Verfahren ist durch den Einsatz gängiger Werkzeuge automatisierbar. Durch eine prototypische Umsetzung wurde die Einbindung der ad-hoc-Kartenerstellung in eine Anwendung des WWW skizziert. Der Prototyp nutzt die standardisierten Technologien des WWW und verknüpft sie mit gängigen Werkzeugen (u.a. dem Content Management System TYPO3, dem Mathematikprogramm MATLAB und dem Programm Qhull zur Berechnung von Voronoi-Diagrammen). Die Werkzeuge sind als Beispiele zu verstehen und könnten durch alternative Implementierungen der benötigten Funktionen ersetzt werden.

Das beschriebene Vorgehen der Kartenerstellung lässt sich auf die eingangs angeführten Szenarien anwenden und ermöglicht mit der dadurch erreichten Prägnanz einen wesentlich effektiveren Kartengebrauch.

In der im Szenario 1 skizzierten Notfallsituation wird die geforderte klare und eindeutige Darstellung und damit die schnelle Unterscheidung des Wesentlichen vom Unwesentlichen sichergestellt. Die Auswahl von möglichst gut unterscheidbaren Farben gewährleistet, dass die dargestellte Route vor dem Kartenhintergrund klar erkennbar ist. Dazu trägt weiterhin auch die Berücksichtigung der Farbdarstellung auf dem PDA unter den beschriebenen Umgebungsbedingungen (helles Umgebungslicht) bei.

Die Anwendung in den Szenarien 2 und 3 sichert die Verfügbarkeit einer großen Anzahl von Farben und damit die Wahrnehmbarkeit der Vielzahl von dargestellten Objekten. Die Unterscheidbarkeit gilt dabei sowohl für die Objekte untereinander als auch gegenüber vorhandenen Kartenhintergründen. Die Möglichkeit, Karten on demand aus verteilten Quellen zusammenzustellen beinhaltet auch, dass diese Karten on demand in verschiedenen graphischen Layouts reproduzierbar sind. Es lassen sich so zu beliebigen Zeiten die gleichen Daten auf die im Szenario 2 geforderten Darstellungsmedien (stationärer Bildschirm, PDA, Beamer) abstimmen, indem gewissermaßen auf Anfrage durch einen Nutzer Farbprofile für unterschiedliche Anwendungskontexte erstellt werden.

Insgesamt wurde damit gezeigt, dass mit verfügbaren Technologien und Daten die Prägnanz einer sorgfältigen kartographischen Gestaltung mit der Kartenerstellung in Echtzeit in Einklang zu bringen ist und sich in ein allgemeines Verfahren überführen lässt.

Zukünftige Erweiterungen können in der Verfeinerung der Kartengestaltung und dem Ausbau des Prototypen liegen.

Ersteres bedeutet eine Steigerung der Unterscheidbarkeit von Objekten in der Karte, indem die zugrunde liegenden graphischen Elemente beachtet werden. Konkret bedeutet dies, dass bei der Nutzung einer Farbe A für ein linienhaftes Objekt diejenigen Farben, die am nächsten zu A sind, möglichst nur für punkt- oder flächenhafte Objekte verwendet werden. Dies erfordert bei der Integration einzelner Objektklassen mit bereits vorgefertigten topographischen Karten (vgl. Beispiele der Freizeitportale im Abschnitt 3.2, Ausgabe des Prototypen in Abbildung 8-6) eine weitergehende Analyse der Graphik dieser Karten: Außer den genutzten Farben muss festgestellt werden, für welche Art von Objekten bzw. welches graphische Element eine bestimmte Farbe verwendet wurde.

Eine weitere Steigerung der Unterscheidbarkeit könnte durch Freistellung oder linienhafte Abgrenzung von Objekten erreicht werden; dies betrifft insbesondere die Differenzierbarkeit punkt- und linienhafter Objekte vor einem flächenhaften Hintergrund. Allerdings ist bei der Nutzung von Abgrenzungen zu beachten, dass diese nicht zu Verwechslungen mit ebenfalls vorkommenden Signaturen anderer Objekte führen.

Eine Erweiterung des Prototypen könnte zunächst in der Einbindung von Geo-Katalogen, d.h. Sammlungen von Datenquellen, liegen. Durch die Bereitstellung geeigneter Vorschau- und

Auswahlwerkzeuge würde dies das Konzept des Portals mit dem des Katalogs verknüpfen und eine Vielzahl von Datenquellen nutzerfreundlich verfügbar machen. Weiterhin ließe sich das Nutzerprofil um Funktionen zur Speicherung und Verwaltung von Kartenzusammenstellungen erweitern. Ein Anwender könnte so verschiedene, für bestimmte Zwecke erstellte, Karten vorhalten und zu späteren Zeitpunkten in gleicher oder angepasster Form erneut nutzen.

Literaturverzeichnis

ADOBE (2007): Using Adobe Gamma on Windows. Online verfügbar unter http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=321608, zuletzt geprüft am 12.09.2008.

ALT, W. (2002): Nichtlineare Optimierung. Eine Einführung in Theorie, Verfahren und Anwendungen. Friedr. Vieweg & Sohn Verlag, Braunschweig/Wiesbaden.

ALTHEIM, M.; MCCARRON, S. (2001): XHTML™ 1.1. Module-based XHTML. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/2001/REC-xhtml11-20010531, zuletzt geprüft am 25.01.2008.

AMBROSE, G.; HARRIS, P. (2007): Grundlagen der Typografie. Stiebner, München.

APT, K. R. (2003): Principles of Constraint Programming. Cambridge University Press, Cambridge.

AURENHAMMER, F. (1991): Voronoi diagrams - a survey of a fundamental geometric data structure. In: ACM Computing Surveys, Jg. 23, H. 3, S. 345–405. Online verfügbar unter http://doi.acm.org/10.1145/116873.116880, zuletzt geprüft am 07.03.2008.

AUSTIN, D.; BARBIR, A.; FERRIS, C.; GARG, S. (2004): Web Services Architecture Requirements. The World Wide Web Consortium. (W3C Working Group Note). Online verfügbar unter http://www.w3.org/TR/wsa-reqs, zuletzt geprüft am 25.01.2008.

AXELSSON, J.; BIRBECK, M.; DUBINKO, M.; EPPERSON, B.; ISHIKAWA, M.; MCCARRON, S.; NAVARRO, A.; PEMBERTON, S. (2006): XHTML™ 2.0. The World Wide Web Consortium. (W3C Working Draft). Online verfügbar unter http://www.w3.org/TR/xhtml2, zuletzt geprüft am 25.01.2008.

BARBER, B.; DOBKIN, D.; HUHDANPAA, H. (1996): The Quickhull Algorithm for Convex Hulls. In: ACM Transactions on Mathematical Software, Jg. 22, H. 4, S. 469–483.

BEHME, H. (2007): Das letzte Siebtel. In: iX Spezial, H. 1, S. 6–7.

BERNERS-LEE, T. (1994): Universal Resource Identifiers in WWW. (RFC, 1630). Online verfügbar unter ftp://ftp.rfc-editor.org/in-notes/rfc1630.txt, zuletzt geprüft am 25.01.2008.

BERNERS-LEE, T.; FIELDING, R. T.; MASINTER, L. (2005): Uniform Resource Identifier (URI): Generic Syntax (RFC 3986). (RFC, 3986). Online verfügbar unter http://gbiv.com/protocols/uri/rfc/rfc3986.html, zuletzt geprüft am 25.01.2008.

BERTIN, J. (1967): Sémiologie graphique. Paris.

BERTIN, J. (1974): Graphische Semiologie. Diagramme Netze Karten. Walter de Gruyter & Co, Berlin.

BOLLMANN, J.; KOCH, W. G. (Hg.) (2002): Lexikon der Kartographie und Geomatik. 2 Bände. Spektrum Akademischer Verlag GmbH, Heidelberg.

BOOTH, D.; HAAS, H.; MCCABE, F.; NEWCOMER, E.; CHAMPION, M.; FERRIS, C.; ORCHARD, D. (2004): Web Services Architecture. The World Wide Web Consortium. (W3C Working Group Note). Online verfügbar unter http://www.w3.org/TR/ws-arch/, zuletzt geprüft am 25.01.2008.

BOOTH, D.; LIU, C. K. (2007): Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/wsdl20-primer, zuletzt geprüft am 25.01.2008.

BOS, B.; ÇELIK, T.; HICKSON, I.; LIE, H. W. (2007): Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. The World Wide Web Consortium. (W3C Candidate Recommendation). Online verfügbar unter http://www.w3.org/TR/CSS21, zuletzt geprüft am 25.01.2008.

BOS, B.; LIE, H. W.; LILLEY, C.; JACOBS, I. (1998): Cascading Style Sheets, level 2. CSS2 Specification. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/REC-CSS2/, zuletzt geprüft am 25.01.2008.

BOYD, S.; VANDENBERGHE, L. (2004): Convex Optimization. Cambridge University Press, Cambridge.

BRAINARD, D. H.; PELLI, D. G.; ROBSON, T. (2002): Display Characterization. In: HORNAK, J. (HG.): Encylopedia of Imaging Science and Technology. Wiley, S. 172–188.

BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F.; COWAN, J. (2006): Extensible Markup Language (XML) 1.1 (Second Edition). The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/xml11, zuletzt geprüft am 25.01.2008.

BRETTEL, H.; VIÉNOT, F.; MOLLON, J. D. (1997): Computerized simulation of color appearance for dichromats. In: Journal of the Optical Society of America A, Jg. 14, H. 10, S. 2647–2655.

BRUNNER, K. (2001): Kartengestaltung für elektronische Bildanzeigen. In: KOCH, W. G. (HG.): Theorie 2000. Kartographische Bausteine, Dresden, S. 76–88.

BUZIEK, G. (2001): Eine Konzeption der kartographischen Visualisierung. Habilitation. Hannover, Fachbereich Bauingenieur- und Vermessungswesen.

CAMPADELLI, P.; POSENATO, R.; SCHETTINI, R. (1999): An algorithm for the selection of high-contrast color sets. In: Color Research & Application, Jg. 24, H. 2, S. 132–138.

CAPPANERA, P. (1999): A Survey on Obnoxious Facility Location Problems. Dipartimento di Informatica, Univ. di Pisa. (TR-99-11).

CARTER, R. C.; CARTER, E. C. (1982): High-contrast sets of colors. In: Applied Optics, Jg. 21, H. 16, S. 2936–2939.

CARTWRIGHT, W. (1999): Development of Multimedia. In: CARTWRIGHT, W.; PETERSON, M. P.; GARTNER, G. (HG.): Multimedia Cartography. Springer-Verlag, Berlin Heidelberg, S. 11–30.

CASTILLO, I.; KAMPAS, F. J.; PINTÉR, J. D. (2008): Solving circle packing problems by global optimization: Numerical results and industrial applications. In: European Journal of Operational Research, Jg. 191, H. 3, S. 786–802.

CHESNEAU, E.; RUAS, A.; BONIN, O. (2005): Colour Contrasts Analysis for a better Legibility of Graphic Signs on Risk Maps. La Corogne, Espagne (Actes de conférences de l'Association de Cartographie Internationale (ICC' 2005)).

CHINNICI, R.; HAAS, H.; LEWIS, A. A.; MOREAU, J.-J.; ORCHARD, D.; WEERAWARANA, S. (2007a): Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/wsdl20-adjuncts, zuletzt geprüft am 25.01.2008.

CHINNICI, R.; MOREAU, J.-J.; RYMAN, A.; WEERAWARANA, S. (2007b): Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/wsdl20, zuletzt geprüft am 25.01.2008.

CHVÁTAL, V. (1983): Linear Programming. W. H. Freeman and Company.

CLEMENT, L.; HATELY, A.; RIEGEN, C. VON; ROGERS, T. (2008): UDDI Version 3.0.2. OASIS. Online verfügbar unter http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm, zuletzt aktualisiert am 01.04.08.

CLEVELAND, W. S.; MCGILL, R. (1984): Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. In: Journal of the American Statistical Association, Jg. 79, H. 387, S. 531–554. Online verfügbar unter http://www.jstor.org/stable/2288400, zuletzt geprüft am 01.09.2008.

COOK, W. J.; CUNNINGHAM, W. H.; PULLEYBLANK, W. R.; SCHRIJVER, A. (1998): Combinatorial Optimization. John Wiley & Sons, Inc, New York.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. (2002): Verteilte Systeme. Konzepte und Design. 3. Aufl. Pearson Studium, München.

DAIN, S. J. (2004): Clinical colour vision tests. In: Clinical and Experimental Optometry, Jg. 87, H. 4-5, S. 276–293.

DIBIASE, D.; MACEACHREN, A. M.; KRYGIER, J. B.; REEVES, C. (1992): Animation and the role of map design in scientific visualization. In: Cartography and Geographic Informations Systems, Jg. 19, H. 4, S. 201–214.

DONAUBAUER, A. J. (2004): Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services. Dissertation. München. Technische Universität, Institut für Geodäsie, GIS und Landmanagement.

DOSTAL, W.; JECKLE, M.; MELZER, I.; ZENGLER, B. (2005): Service-orientierte Architekturen mit Web Services. Konzepte - Standards - Praxis. Elsevier GmbH, München.

DREZNER, Z.; ERKUT, E. (1995): Solving the Continuous p-Dispersion Problem Using Non-Linear Programming. In: The Journal of the Operational Research Society, Jg. 46, H. 4, S. 516–520.

DUSCHENE, P.; SONNET, J. (2005): WMS Change Request: Support for WSDL & SOAP. (OpenGIS Discussion Paper, OGC 04-050r1). Online verfügbar unter http://portal.opengeospatial.org/files/?artifact_id=9541, zuletzt geprüft am 10.02.2008.

ECMA (1999): Standard ECMA-262. ECMAScript Language Specification. European Computer Manufacturers Association. Online verfügbar unter http://www.ecma-international.org/publications/standards/Ecma-262.htm, zuletzt geprüft am 26.08.2008.

EGENHOFER, M.; FRANZOSA, R. (1991): Point-Set Topological Spatial Relations. In: International Journal of Geographical Information Systems, Jg. 5, H. 2, S. 161–174.

EISELT, H. A.; LAPORTE, G. (1995): Objectives in Location Problems. In: DREZNER, Z. (HG.): Facility Location. A Survey of Applications and Methods. Springer-Verlag, New York (Springer Series in Operations Research), S. 151–180.

ELLSIEPEN, I. (2005): Methoden der effizienten Informationsübermittlung durch Bildschirmkarten. Dissertation. Bonn. Rheinische Friedrich-Wilhelms Universität, Institut für Kartographie und Geoinformation. Online verfügbar unter http://hss.ulb.uni-bonn.de/diss_online/landw_fak/2005/ellsiepen_iris, zuletzt geprüft am 30.08.2008.

ELLSIEPEN, M. (2002): Formalisierung kartographischen Wissens zur Schriftplazierung. Dissertation. Bonn. Rheinische Friedrich-Wilhelms Universität, Institut für Kartographie und Geoinformation. Online verfügbar unter http://hss.ulb.uni-bonn.de/diss_online/landw_fak/2001/ellsiepen_matthias, zuletzt geprüft am 30.08.2008.

ERKUT, E. (1990): The discrete p-dispersion problem. In: European Journal of Operational Research, Jg. 46, H. 1, S. 48–60.

ERKUT, E.; NEUMAN, S. (1989): Analytical models for locating undesirable facilities. In: European Journal of Operational Research, Jg. 40, H. 3, S. 275–291.

FALLSIDE, D. C.; WALMSLEY, P. (2004): XML Schema Part 0: Primer Second Edition. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/, zuletzt geprüft am 25.01.2008.

FIELDING, R. T.; GETTYS, J.; MOGUL, J. C.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. (1999): Hypertext Transfer Protocol -- HTTP/1.1. (RFC, 2616). Online verfügbar unter http://www.ietf.org/rfc/rfc2616.txt, zuletzt geprüft am 25.01.2008.

FLANAGAN, D. (2002): JavaScript. Das umfassende Referenzwerk. 2. Aufl. O' Reilly Verlag GmbH & Co. KG, Köln.

FLETCHER, R. (1987): Practical Methods of Optimization. 2. Aufl. John Wiley & Sons Ltd.

FORNEFELD, M.; OEFINGER, P.; JAENICKE, K. (2004): Nutzen von Geodateninfrastrukturen. Herausgegeben von MICUS Management Consulting GmbH. Online verfügbar unter http://www.micus.de/50_publikationen.html, zuletzt geprüft am 27.08.2008.

FORSTER, O. (1984): Analysis 2. 5. Aufl. Friedr. Vieweg & Sohn Verlag, Braunschweig/Wiesbaden.

FRASER, B.; MURPHY, C.; BUNTING, F. (2005): Color Management. Industrial-Strength Production Techniques. 2. Aufl. Peachpit Press, Berkeley.

FYLSTRA, D. (2005): Introducing Convex and Conic Optimization for the Quantitative Finance Professional. In: Wilmott Magazine, S. 18–22.

GARRETT, J. J. (2005): Ajax: A New Approach to Web Applications. Online verfügbar unter http://www.adaptivepath.com/ideas/essays/archives/000385.php, zuletzt geprüft am 08.02.2008.

GEIGER, C.; KANZOW, C. (2002): Theorie und Numerik restringierter Optimierungsaufgaben. Springer-Verlag, Berlin Heidelberg.

GLASBEY, C.; VAN DER HEIJDEN, G.; TOH, V. F. K.; GRAY, A. (2007): Colour displays for categorical images. In: Color Research & Application, Jg. 32, H. 4, S. 304–309. Online verfügbar unter http://dx.doi.org/10.1002/col.20327, zuletzt geprüft am 11.03.2008.

GROOT, R.; MCLAUGHLIN, J. (2000): Introduction. In: GROOT, R.; MCLAUGHLIN, J. (HG.): Geospatial data infrastructure. Concepts, cases, and good practice. Oxford University Press, Oxford, S. 1–12.

GRÜNREICH, D. (1996): Der Standort der Kartographie im multimedialen Umfeld. In: MAYER, F.; KRIZ, K. (HG.): Kartographie im multimedialen Umfeld, Wien (Wiener Schriften zur Geographie und Kartographie, 8), S. 17–28.

GUDGIN, M.; HADLEY, M.; MENDELSOHN, N.; MOREAU, J.-J.; NIELSEN, H. F.; KARMARKAR, A.; LAFON, Y. (2007a): SOAP Version 1.2 Part 1: Messaging Framework (Second Edi-

tion). The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/soap12-part1/, zuletzt geprüft am 25.01.2008.

GUDGIN, M.; HADLEY, M.; MENDELSOHN, N.; MOREAU, J.-J.; NIELSEN, H. F.; KARMARKAR, A.; LAFON, Y. (2007b): SOAP Version 1.2 Part 2: Adjuncts (Second Edition). The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/soap12-part2/, zuletzt geprüft am 25.01.2008.

GÜTING, R. H.; DIEKER, S. (2003): Datenstukturen und Algorithmen. 2. Aufl. B. G. Teubner GmbH, Stuttgart/Leipzig/Wiesbaden.

HAKE, G. (1988): Gedanken zu Form und Inhalt heutiger Karten. In: Kartographische Nachrichten, Jg. 38, S. 65–72.

HAKE, G.; GRÜNREICH, D.; MENG, L. (2002): Kartographie. Visualisierung raum-zeitlicher Informationen. 8. Auflage. Walter de Gruyter & Co, Berlin.

HAMMERSCHALL, U. (2006): Verteilte Systeme und Anwendungen. Architekturkonzepte, Standards und Middleware-Technologien. Pearson Studium, München.

HAN, S. P. (1977): A Globally Convergent Method for Nonlinear Programming. In: Journal of Optimization Theory, Jg. 22, H. 3, S. 297–309.

HAROLD, E. R.; MEANS, W. S. (2005): XML in a Nutshell. Deutsche Ausgabe. 3. Aufl. O' Reilly Verlag GmbH & Co. KG, Köln.

HEIDMANN, F. (1999): Aufgaben- und nutzerorientierte Unterstützung kartographischer Kommunikationsprozesse durch Arbeitsgraphik. Konzeptionen, Modellbildung und experimentelle Untersuchungen. GCA-Verlag, Herdecke.

HELLER, E. (1999): Wie Farben wirken. Farbpsychologie, Farbsymbolik, Kreative Farbgestaltung. 10. Aufl. Rowohlt, Reinbek.

HICKSON, I.; HYATT, D. (2008): HTML 5. The World Wide Web Consortium. (W3C Working Draft). Online verfügbar unter http://www.w3.org/TR/html5/, zuletzt geprüft am 25.01.2008.

HOCK, W.; SCHITTKOWSKI, K. (1983): A comparative performance evaluation of 27 nonlinear programming codes. In: Computing, Jg. 30, H. 4, S. 335–358.

HOMANN, J.-P. (2007): Digitales Colormanagement. Grundlagen und Strategien zur Druckproduktion mit ICC-Profilen, der ISO 12647-2 und PDF/X-1a. 3. Aufl. Springer-Verlag, Berlin Heidelberg (X.media.press).

HOOKER, J. N. (2006): Operations Research Methods in Constraint Programming. In: ROSSI, F.; VAN BEEK, P.; WALSH, T. (HG.): Handbook of Constraint Programming. Elsevier, Amsterdam (Foundations of Artificial Intelligence), S. 527–570.

ICC (2004): Specification ICC.1:2004-10. Image technology colour management — Architecture, profile format, and data structure. Online verfügbar unter http://www.color.org/icc_specs2.xalter, zuletzt geprüft am 09.09.2008.

ISO-Standard, 19123 2005: Geographic Information - Schema for coverage geometry and functions.

JARRE, F.; STOER, J. (2004): Optimierung. Springer-Verlag, Berlin Heidelberg.

JENNY, B.; KELSO, N. V. (2007): Color design for the color vision impaired. In: Cartographic Perspectives, Jg. 58, S. 61–67. Online verfügbar unter

http://jenny.cartography.ch/pdf/2007_JennyKelso_ColorDesign_lores.pdf, zuletzt geprüft am 18.09.2008.

KATZ, M. J.; KEDEM, K.; SEGAL, M. (2002): Improved algorithms for placing undesirable facilities. In: Computers and Operations Research, Jg. 29, S. 1859–1872.

KELLERER, H.; PFERSCHY, U.; PISINGER, D. (2004): Knapsack Problems. Springer-Verlag, Berlin Heidelberg.

KÖBBEN, B. (2001): Publishing maps on the Web. In: KRAAK, M.-J.; BROWN, A. (HG.): Web Cartography. Developments and Prospects. Taylor & Francis, London, S. 73–86.

KOLÁCNÝ (1977): Cartographic Information - A fundamental Concept and Term in modern Cartography. In: GUELKE, L. (HG.): Cartographica Monograph. The Nature of cartographic communication, S. 39–45.

KOLBE, T. H.; STEINRÜCKEN, J.; PLÜMER, L. (2003): Cooperative Public Web Maps: Proceedings of the 21st International Cartographic Conference. Durban, South Africa.

KRAAK, M.-J. (2001): Settings and needs for web cartography. In: KRAAK, M.-J.; BROWN, A. (HG.): Web Cartography. Developments and Prospects. Taylor & Francis, London, S. 1–7.

KUCHENBECKER, J.; RÖHL, F.; WESSELBURG, A.; BERNARDING, J.; BEHRENS-BAUMANN, W. (2007): Untersuchungen zur Validität eines webbasierten Farbsehtests für Screeninguntersuchungen des Farbensehens. In: Der Ophthalmologe, Jg. 104, H. 1, S. 47–53.

LA BEAUJARDIERE, J. DE (2006): OpenGIS Web Map Server Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 06-042). Online verfügbar unter http://www.opengeospatial.org/standards/wms, zuletzt geprüft am 10.02.2008.

LABORENZ, K.; WENDT, T.; ERTEL, A.; DUSSOYE, P.; HINZ, E. (2006): TYPO3 4.0. Das Handbuch für Entwickler. 2. Aufl. Galileo Press, Bonn.

LALONDE, W. (2002): Styled Layer Descriptor Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 02-070).

LANG, H. (1995): Farbwiedergabe in den Medien. Fernsehen Film Druck. Muster-Schmidt Verlag, Göttingen.

LE HORS, A.; LE HÉGARET, P.; WOOD, L.; NICOL, G.; ROBIE, J.; CHAMPION, M.; BYRNE, S. (2004): Document Object Model (DOM) Level 3 Core Specification. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/DOM-Level-3-Core, zuletzt geprüft am 25.01.2008.

LEXIKON DER ZEIT (2005): Web-Portal. In: ZEITVERLAG GERD BUCERIUS GMBH & CO. KG (HG.): Die Zeit - Das Lexikon. 20 Bände. Zeitverlag Gerd Bucerius GmbH & Co. KG, Hamburg.

LUPP, M. (2007): Styled Layer Descriptor profile of the Web Map Service Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 05-078r4). Online verfügbar unter http://www.opengeospatial.org/standards/sld, zuletzt geprüft am 10.02.2008.

LUSTIG, I. J.; PUGET, J.-F. (2001): Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming. In: INTERFACES, Jg. 31, H. 6, S. 29–53. Online verfügbar unter http://interfaces.journal.informs.org/cgi/content/abstract/31/6/29.

MACEACHREN, A. M. (1994): Visualization in Modern Cartography: Setting the Agenda. In: MACEACHREN, A. M.; TAYLOR, D. R. F. (HG.): Visualization in Modern Cartography. Elsevier, Oxford, S. 1–12.

MACEACHREN, A. M. (1995): How Maps Work. Representation, Visualization, and Design. The Guilford Press, New York.

MACKINLAY, J. (1986): Automating the Design of Graphical Presentations of Relational Information. In: ACM Transactions on Graphics, Jg. 5, H. 2, S. 110–141.

MALIC, B. (1998): Physiologische und technische Aspekte kartographischer Bildschirmvisualisierung. Dissertation. Bonn. Rheinische Friedrich-Wilhelms Universität, Institut für Kartographie und Geoinformation.

MATLAB (2007): Produkthilfe - Version 7.5.0.

MCCARRON, S.; ISHIKAWA, M. (2007): XHTML™ 1.1 - Module-based XHTML - Second Edition. The World Wide Web Consortium. (W3C Working Draft). Online verfügbar unter http://www.w3.org/TR/xhtml11, zuletzt geprüft am 25.01.2008.

MELACHRINOUDIS, E. (1988): An Efficient Computational Procedure for the Rectilinear MAXIMIN Location Problem. In: Transportation Science, Jg. 22, H. 3, S. 217–223.

MESEGUER, P.; ROSSI, F.; SCHIEX, T. (2006): Soft Constraints. In: ROSSI, F.; VAN BEEK, P.; WALSH, T. (HG.): Handbook of Constraint Programming. Elsevier, Amsterdam (Foundations of Artificial Intelligence), S. 281–328.

MEYER, G. W.; GREENBERG, D. P. (1988): Color-Defective Vision and Computer Graphics Displays. In: IEEE Computer Graphics and Applications, Jg. 08, H. 5, S. 28–40.

MILLER, S. (1999): Design of Multimedia Mapping Products. In: CARTWRIGHT, W.; PETERSON, M. P.; GARTNER, G. (HG.): Multimedia Cartography. Springer-Verlag, Berlin Heidelberg, S. 51–63.

MINTERT, S.; KÜHNEL, C. (2000): Workshop JavaScript. Addison-Wesley, München.

MITRA, N.; LAFON, Y. (2007): SOAP Version 1.2 Part 0: Primer (Second Edition). The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/soap12-part0/, zuletzt geprüft am 25.01.2008.

MORRIS, C. W. (1972): Grundlagen der Zeichentheorie / Ästhetik der Zeichentheorie. Carl Hanser Verlag, München.

MORRISON, J. L. (1974): A theoretical framework for cartographic generalization with the emphasis on the process of symbolization: International Yearbook of Cartography (14), S. 115–127.

MOSHE, B.; KATZ, M. J.; SEGAL, M. (2000): Obnoxious Facility Location: Complete Service with Minimal Harm. In: International Journal of Computational Geometry and Applications, Jg. 10, H. 6, S. 581–592.

MÜLLER, M. (2006): Symbology Encoding Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 05-077r4), zuletzt geprüft am 13.02.2008.

MÜNZ, S.; NEFZGER, W. (1999): HTML 4.0 Handbuch. HTML - JavaScript - DHTML - Perl. 3. Aufl. Franzis´ Verlag GmbH, Poing.

NAGY, A. L.; SANCHEZ, R. R.; HUGHES, T. C. (1990): Visual search for color differences with foveal and peripheral vision. In: Journal of the Optical Society of America A, Jg. 7, H. 10, S. 1995–2001.

NEUDECK, S. (2001): Zur Gestaltung topografischer Karten für die Bildschirmvisualisierung. Dissertation. München. Universität der Bundeswehr, Studiengang Geodäsie und Geoinformation.

NIX, M. (2005): Web Content Management. CMS verstehen und auswählen. Software & Support Verlag GmbH, Frankfurt.

DIN 1979: DIN 5033: Farbmessung, Grundbegriffe der Farbmetrik.

NOWELL, L. T. (1997): Graphical Encoding for Information Visualization: Using Icon Color, Shape, and Size to Convey Nominal and Quantitative Data. Dissertation. Blacksburg, Virginia. Virginia Tech, Computer Science.

OGC (2006): OGC Newsletter - July 2006. Online verfügbar unter http://www.opengeospatial.org/pressroom/newsletters/200607, zuletzt geprüft am 25.01.2008.

OKABE, A.; BOOTS, B.; SUGIHARA, K.; CHIU, S. N. (2000): Spatial Tesselations. Concepts and Applications of Voronoi Diagrams. 2. Aufl. John Wiley & Sons Ltd, Chichester.

OKABE, A.; SUZUKI, A. (1997): Locational optimization problems solved through Voronoi diagrams. In: European Journal of Operational Research, Jg. 98, H. 3, S. 445–456.

OLBRICH, G.; QUICK, M.; SCHWEIKART, J. (2002): Desktop Mapping. Grundlagen und Praxis in Kartographie und GIS. 3. Aufl. Springer-Verlag, Berlin Heidelberg.

OLSON, J. M.; BREWER, C. A. (1997): An Evaluation of Color Selections to Accommodate Map Users with Color-Vision Impairments. In: Annals of the Association of American Geographers, Jg. 87, H. 1, S. 103–134.

O'REILLY, T. (2005): What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. Online verfügbar unter http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, zuletzt geprüft am 25.01.2008.

OTTMANN, T.; WIDMAYER, P. (1996): Algorithmen und Datenstrukturen. 3. Aufl. Spektrum Akademischer Verlag GmbH, Heidelberg.

PERCIVALL, G. (2003): OGC Reference Model. Open Geospatial Consortium Inc. (OGC 03-040). Online verfügbar unter http://portal.opengeospatial.org/files/?artifact_id=3836, zuletzt geprüft am 10.02.2008.

PETERSON, M. P. (1999): Elements of Multimedia Cartography. In: CARTWRIGHT, W.; PETERSON, M. P.; GARTNER, G. (HG.): Multimedia Cartography. Springer-Verlag, Berlin Heidelberg, S. 31–40.

PETZOLD, I. (2003): Beschriftung von Bildschirmkarten in Echtzeit. Konzept und Struktur. Dissertation. Bonn. Rheinische Friedrich-Wilhelms-Universität, Institut für Kartographie und Geoinformation. Online verfügbar unter http://hss.ulb.uni-bonn.de/diss_online/landw_fak/2003/petzold_ingo, zuletzt geprüft am 30.08.2008.

PISINGER, D. (1999): Exact solution of p-dispersion problems. University of Copenhagen. (Technical Report, 99/14).

POPPER, A. (2008): An der frischen Luft. Enterprise Content Management mit Alfresco. In: iX Spezial, H. 7, S. 76–82.

PREPARATA, F. P.; SHAMOS, M. I. (1985): Computational Geometry. An Introduction. Springer-Verlag, New York.

PURVES, D.; LOTTO, R. B. (2003): Why We See What We Do. An Empirical Theory of Vision. Sinauer Associates, Inc., Sunderland.

RAGGETT, D.; LE HORS, A.; JACOBS, I. (1999): HTML 4.01 Specification. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/html401, zuletzt geprüft am 25.01.2008.

RIGDEN, C. (1999): 'The Eye of the Beholder' - Designing for Colour-Blind Users. In: British Telecommunications Engineering, Jg. 17, S. 2–6.

ROBINSON, A. H.; MORRISON, J. L.; MUEHRCKE, P. C.; KIMERLING, A. J.; GUPTILL, S. C. (1995): Elements of Cartography. 6. Aufl. John Wiley & Sons Ltd, New York.

ROCKAFELLAR, R. (1993): Lagrange Multipliers and Optimality. In: SIAM Review, Jg. 35, H. 2, S. 183–238.

ROCKLEY, A. (2003): Managing Enterprise Content. A Unified Content Strategy. Unter Mitarbeit von Pamela Kostur und Steve Manning. New Riders, Berkeley.

RÖSCH, H. (2000): Internetportal, Unternehmensportal, Wissenschaftsportal. Typologie und Funktionalität der wichtigsten Portalkonzeptionen. In: KNORZ, G.; KUHLEN, R. (HG.): Informationskompetenz - Basiskompetenz in der Informationsgesellschaft, Konstanz. UVK Verlagsgesellschaft. Proceedings des 7. Internationalen Symposiums für Informationswissenschaft, S. 245–264.

RUSSELL, S.; NORVIG, P. (2003): Artificial Intelligence. A Modern Approach. 2. Aufl. Pearson Education, Inc., New Jersey.

SALOMON, R. (1996): Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. In: Biosystems, Jg. 39, H. 3, S. 263–278.

SCHERFF, J. (2006): Grundkurs Computernetze. Friedr. Vieweg & Sohn Verlag, Wiesbaden.

SCHILL, A.; SPRINGER, T. (2007): Verteilte Systeme. Grundlagen und Basistechnologien. Springer-Verlag, Berlin Heidelberg.

SCHITTKOWSKI, K. (1985): NLPQL: A Fortran Subroutine solving constrained nonlinear Programming Problems. In: Annals of Operations Research, Jg. 5, S. 485–500.

SCHLÄPFER, K. (1993): Farbmetrik in der Reproduktionstechnik und im Mehrfarbendruck. 2. Aufl. UGRA - Verein zur Förderung wissenschaftlicher Untersuchungen in der grafischen Industrie, St. Gallen.

SCHOPPMEYER, J. (1978): Die Wahrnehmung von Rastern und die Abstufung von Tonwertskalen in der Kartographie. Dissertation. Bonn. Rheinische Friedrich-Wilhelms Universität, Institut für Kartographie und Topographie.

SCHUMANN, H.; MÜLLER, W. (2000): Visualisierung. Grundlagen und allgemeine Methoden. Springer-Verlag, Berlin Heidelberg.

SEGOR, C. (2006): Sicher surfen. In: iX Spezial, H. 9, S. 128–130.

SHAN, Y.-P.; EARLE, R. H. (1998): Enterprise Computing with Objects. From Client/Server Environments to the Internet. Addison Wesley Longman, Inc, Reading, Massachusetts.

SIMON, K. (2008): Farbe im Digitalen Publizieren. Konzepte der digitalen Farbwiedergabe für Office, Design und Software. Springer-Verlag, Berlin Heidelberg.

SMALLMAN, H. S.; BOYNTON, R. M. (1990): Segregation of basic colors in an information display. In: Journal of the Optical Society of America A, Jg. 7, H. 10, S. 1985–1994.

SPANNEBERG, B.; MINTERT, S. (2007): Nadeln im Heu. Clientseitige Ajax- und RIA-Tools. In: iX Spezial, H. 1, S. 8–13.

SPENCE, R. (2001): Information Visualization. ACM Press, Harlow, England.

STEINRÜCKEN, J. (2001): Multimediales GIS mit den Mitteln des Internet am Beispiel der Routenplanung. Diplomarbeit. Bonn. Rheinische Friedrich-Wilhelms-Universität, Institut für Kartographie und Geoinformation. Online verfügbar unter http://www.ikg.uni-bonn.de/uploads/tx_ikgpublication/joerg_steinruecken.pdf, zuletzt geprüft am 15.01.2009.

STEYER, R. (2006): Ajax mit PHP. Beschleunigte Webapplikationen für das Web 2. Addison-Wesley, München.

STRZEBKOWSKI, R.; KLEEBERG, N. (2002): Interaktivität und Präsentation als Komponenten multimedialer Lernanwendungen. In: ISSING, L. J.; KLIMSA, P. (HG.): Information und Lernen mit Multimedia und Internet. Lehrbuch für Studium und Praxis. 3 Aufl. Verlagsgruppe Beltz, Psychologische Verlags Union, Weinheim, S. 229–245.

SZABÓ, P. G.; MARKÓT, M. C.; CSENDES, T. (2005): Global Optimization in Geometry — Circle Packing into the Square. In: AUDET, C.; HANSEN, P.; SAVARD, G. (HG.): Essays and Surveys in Global Optimization. Springer Science+Business Media Inc., New York, S. 233–265.

TAMIR, A. (2006): Locating two obnoxious facilities using the weighted maximin criterion. In: Operations Research Letters, Jg. 34, H. 1, S. 97–105.

TANENBAUM, A. S.; VAN STEEN, M. (2007): Distributed Systems. Principles and Paradigms. Pearson Education, Inc., Upper Saddle River, New Jersey.

TOUSSAINT, G. T. (1983): Computing largest empty circles with location constraints. In: International Journal of Parallel Programming, Jg. 12, H. 5, S. 347–358.

VAN BEEK, P. (2006): Backtracking Search Algorithms. In: ROSSI, F.; VAN BEEK, P.; WALSH, T. (HG.): Handbook of Constraint Programming. Elsevier, Amsterdam (Foundations of Artificial Intelligence), S. 85–134.

VANDERBEI, R. J. (1996): Linear Programming: Foundations and Extensions. Kluwer.

VIÉNOT, F.; BRETTEL, H.; MOLLON, J. D. (1999): Digital Video Colourmaps for Checking the Legibility of Displays by Dichromats. In: Color Research & Application, Jg. 24, H. 4, S. 243–252.

VRETANOS, P. A. (2005a): OpenGIS® Filter Encoding Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 04-095). Online verfügbar unter http://www.opengeospatial.org/standards/filter, zuletzt geprüft am 13.02.2008.

VRETANOS, P. A. (2005b): Web Feature Service Implementation Specification. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 04-094). Online ver-

fügbar unter http://www.opengeospatial.org/standards/wfs, zuletzt geprüft am 10.02.2008.

W3C (2002): XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition). A Reformulation of HTML 4 in XML 1.0. The World Wide Web Consortium. (W3C Recommendation). Online verfügbar unter http://www.w3.org/TR/xhtml1, zuletzt geprüft am 25.01.2008.

WARE, C. (2000): Information Visualization. Perception for Design. Academic Press, San Diego.

WARTALA, R. (2007): Bildergeflimmer. Mashups mit der Flickr-API erstellen. In: iX Spezial, H. 1, S. 19–22.

WEICKER, K. (2007): Evolutionäre Algorithmen. 2. Aufl. Teubner, Stuttgart.

WEIDENMANN, B. (2002): Abbilder in Multimediaanwendungen. In: ISSING, L. J.; KLIMSA, P. (HG.): Information und Lernen mit Multimedia und Internet. Lehrbuch für Studium und Praxis. 3 Aufl. Verlagsgruppe Beltz, Psychologische Verlags Union, Weinheim, S. 83–96.

WELCH, S.; SALHI, S.; DREZNER, Z. (2006): The multifacility maximin planar location problem with facility interaction. In: IMA Journal of Management Mathematics, Jg. 17, S. 397–412.

WHITESIDE, A. (2007): Definition identifier URNs in OGC namespace. Open Geospatial Consortium Inc. (OGC Best Practices Paper, 07-092r1). Online verfügbar unter http://portal.opengeospatial.org/files/?artifact_id=24045, zuletzt geprüft am 26.08.2008.

WHITESIDE, A.; EVANS, J. D. (2008): Web Coverage Service (WCS) Implementation Standard. Open Geospatial Consortium Inc. (OpenGIS Implementation Specification, OGC 07-067r5). Online verfügbar unter http://www.opengeospatial.org/standards/wcs, zuletzt geprüft am 10.06.2008.

WILLIAMS, H. P. (1999): Model Building in Mathematical Programming. 4. Aufl. John Wiley & Sons Ltd, Chichester.

WINSTON, W. L. (1991): Introduction to Mathematical Programming. Applications and Algorithms. PWS-KENT Publishing Company, Boston.

WORBOYS, M.; DUCKHAM, M. (2004): GIS. A Computing Perspective. 2. Aufl. CRC Press, Boca Raton, Florida.

WYSZECKI, G.; STILES, W. S. (1982): Color Science. Concepts and Methods, Quantitative Data and Formulae. 2. Aufl. John Wiley & Sons, Inc, New York.

X-BORDER-GDI: E-RigG. Mit digitaler Wanderkarte über die Grenze. Online verfügbar unter http://www.x-border-gdi.org/de/projekte/projektdetails/index.html?&tx_xbprojekt_pi1[xbproj]=1&tx_xbprojekt_pi1[bckpg]=1&cHash=0df28c0df1, zuletzt geprüft am 29.08.2008.

YOUNG, J. F. (1975): Einführung in die Informationstheorie. R. Oldenbourg Verlag GmbH, München.

ZHANG, J. (2008): Visualization for Information Retrieval. Springer-Verlag, Berlin Heidelberg.

Anhang

A.1 Web Map Service (WMS)

Der Web Map Service ist im Abschnitt 2.3.3.2 dieser Arbeit beschrieben. Im Folgenden werden die Parameter einer GetCapabilities-, GetMap- und GetFeatureInfo-Anfrage zusammengefasst.

A.1.1 Parameter einer GetCapabilities-Anfrage

Anfrageparameter	zwingend/ optional	Beschreibung
VERSION=version	O	Version der Anfrage (entspricht Version des vorliegenden Standards)
SERVICE=WMS	Z	Typ des angefragten Services
REQUEST=GetCapabilities	Z	Name der angefragten Operation
FORMAT=MIME_type	O	Rückgabeformat der Anfrage; ein WMS muss XML unterstützen, andere Formate sind optional
UPDATESEQUENCE=string	O	Fügt den Capabilities die Häufigkeit der Aktualisierung des WMS hinzu

Tabelle A.1-1: Parameter einer GetCapabilities-Anfrage (nach de La Beaujardiere 2006)

A.1.2 Parameter einer GetMap-Anfrage

Anfrageparameter	zwingend/ optional	Beschreibung
VERSION=1.3.0	Z	Version der Anfrage (entspricht Version des vorliegenden Standards)
REQUEST=GetMap	Z	Name der angefragten Operation
LAYERS=layer_list	Z	Kommaseparierte Liste der angefragten Kartenthemen
STYLES=style_list	Z	Kommaseparierte Liste von Darstellungsanweisungen für die LAYER, die Zuordnung zu den Layern erfolgt in Reihenfolge der Liste. Es können nur Werte genutzt werden, die der Service in

		den Capabilities angibt.
CRS=namespace:identifier	Z	Gewünschtes Referenzsystem bzw. Projektion der angefragten Karte[1]
BBOX=minx,miny,maxx,maxy	Z	Umschließendes Rechteck (linke untere Ecke, rechte obere Ecke) in Koordinaten des abgefragten Referenzsystems
WIDTH=output_width	Z	Breite des abgefragten Kartenbildes in Pixeln
HEIGHT=output_height	Z	Höhe des abgefragten Kartenbildes in Pixeln
FORMAT=output_format	Z	Grafik-Format des angefragten Kartenbildes (z.B. image/gif oder image/jpeg)
TRANSPARENT=TRUE|FALSE	O	Hintergrund-Transparenz des Bildes; Standardwert ist FALSE
BGCOLOR=color_value	O	RGB-Farbwert des Hintergrunds in hexadezimaler Kodierung; Standardwert ist weiß (#FFFFFF)
EXCEPTIONS= exception_format	O	Rückgabeformat von Exceptions; Standardformat ist XML
TIME=time	O	Angabe eines Zeitpunkts zur Abfrage zeitabhängiger Daten
ELEVATION=elevation	O	Angabe eines Höhenwertes
Other sample dimension(s)	O	Weitere Dimensionen, beispielsweise Wellenlängen von Fernerkundungsdaten

Tabelle A.1-2: Parameter einer GetMap-Anfrage (nach de La Beaujardiere 2006)

A.1.3 Parameter einer GetFeatureInfo-Anfrage

Anfrageparameter	zwingend/ optional	Beschreibung
VERSION=1.3.0	Z	Version der Anfrage (entspricht Version des vorliegenden Standards)
REQUEST=GetFeatureInfo	Z	Name der angefragten Operation

[1] Die Angabe von CRS kann durch eindeutige Namen erfolgen, die sich auf Definitionen von Referenzsystemen bzw. Projektionen verschiedener Institutionen beziehen. Ein Name besteht aus Namespace, Doppelpunkt und einem Code; die Spezifikation des WMS sieht u.a. die Namespaces CRS und EPSG vor. Ersterer ist eine Definition des OGC, letzterer bezieht sich auf eine sehr umfassende Sammlung weltweiter Referenzsysteme der International Association of Oil and Gas Producers (EPSG: European Petroleum Survey Group). EPSG:31466 steht beispielsweise für Gauss-Krüger, 2. Streifen, Deutsches Hauptdreiecksnetz, Fundamentalpunkt Rauenberg, Bessel-Ellipsoid. Nach einem jüngeren Best Practices Paper des OGC (Whiteside 2007) sollte die Angabe allerdings im URN Namespace des OGC erfolgen: urn:ogc:def:crs:EPSG:Version:Code. Einzusetzen sind in der Praxis jeweils „Version“ und „Code“; während letzterer dem oben genannten Code entspricht, bezeichnet Version die Version der genutzten EPSG-Datenbank.

map request part	Z	Parameter der GetMap-Operation, die das Kartenbild, zu dem jetzt Informationen abgefragt werden, zurückgeliefert hat
QUERY_LAYERS=layer_list	Z	Kommaseparierte Liste der angefragten Kartenthemen
INFO_FORMAT=output_format	Z	Format der zurückgegebenen Informationen
FEATURE_COUNT=number	O	Maximale Anzahl von Features je Layer, die zurückgegeben werden (Standardwert: 1)
I=pixel_column	Z	Rechtswert des Features in Pixelkoordinaten des Kartenbildes
J=pixel_row	Z	Hochwert des Features in Pixelkoordinaten des Kartenbildes
EXCEPTIONS=exception_format	O	Rückgabeformat von Exceptions; Standardformat ist XML

Tabelle A.1-3: Parameter einer GetFeatureInfo-Anfrage (nach de La Beaujardiere 2006)

A.2 Styled Layer Descriptor Profile für den WMS

Das Styled Layer Descriptor Profile ist im Abschnitt 2.3.3.3 vorgestellt worden. Es erweitert einen WMS um die Möglichkeit, Daten von außen in ihrer graphischen Darstellung zu beeinflussen. An dieser Stelle werden die Parameter einer GetMap-Anfrage, die über die eines einfachen WMS hinausgehen, und die Parameter einer DescribeLayer-Anfrage zusammengefasst.

A.2.1 Parameter einer GetMap-Anfrage

Anfrageparameter	zwingend/ optional	Beschreibung
SLD=url	O	URL einer beliebig im WWW abgelegten Datei mit Style-Anweisungen
SLD_BODY=string	O	Style-Anweisungen, als Zeichenkette direkt in die Anfrage-URL kodiert
REMOTE_OWS_TYPE=wfs	O	Typ einer angefragten externen Datenquelle (WFS oder WCS)
REMOTE_OWS_URL=url	O	URL einer angefragten externen Datenquelle
SLD_VERSION=1.1.0	Z	Version der unterstützten SLD-Version

Tabelle A.2-1: Parameter einer GetMap-Anfrage (nach Lupp 2007)

A.2.2 Parameter einer DescribeLayer-Anfrage

Anfrageparameter	zwingend/ optional	Beschreibung
SERVICE=WMS	Z	Typ des angefragten Services
REQUEST=DescribeLayer	Z	Name der angefragten Operation
VERSION=1.3.0	Z	Version der Anfrage (entspricht Version des vorliegenden Standards)
LAYERS=layer_list	Z	Kommaseparierte Liste der angefragten Kartenthemen
SLD_VERSION=1.1.0	Z	Version der unterstützten SLD-Version

Tabelle A.2-2: Parameter einer DescribeLayer-Anfrage (nach Lupp 2007)

A.3 Nutzung von Farbe

A.3.1 Normvalenzen von RGB-Primärvalenzen

RGB-Farbräume werden typischerweise durch die Angabe der Normfarbwertanteile der Primärvalenzen ($x_R, y_R, x_G, y_G, x_B, y_B$) und des Weißpunktes ($x_W, y_W$) charakterisiert (vgl. Abschnitt 5.3.2). Für die Transformation zwischen einem RGB-Farbraum und dem Normvalenzsystem müssen die Normfarbwerte der RGB-Primärvalenzen bekannt sein (vgl. Abschnitt 5.2), d.h. zu bestimmen sind:

$$\vec{R} = \begin{pmatrix} X_R \\ Y_R \\ Z_R \end{pmatrix} \qquad \vec{G} = \begin{pmatrix} X_G \\ Y_G \\ Z_G \end{pmatrix} \qquad \vec{B} = \begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix}.$$

Die folgende Berechnung findet sich beispielsweise in Simon (Simon 2008) oder Lang (Lang 1995).

Für Weiß gilt:

$$\vec{R} + \vec{G} + \vec{B} = \vec{W}$$

bzw.

$$\begin{aligned} X_R + X_G + X_B &= X_W \\ Y_R + Y_G + Y_B &= Y_W \\ Z_R + Z_G + Z_B &= Z_W \end{aligned} \quad . \tag{A.1}$$

Für die Normfarbwertanteile von Weiß gilt gemäß der Projektion der Normfarbwerte in die Ebene (Abschnitt 5.3.1) zusammen mit $Y_W = 1$:

$$y_W = \frac{Y_W}{X_W + Y_W + Z_W} = \frac{1}{X_W + 1 + Z_W}, \tag{A.2}$$

$$\frac{x_W}{y_W} = \frac{X_W}{\frac{1}{X_W + 1 + Z_W}\left(X_W + 1 + Z_W\right)} = X_W, \tag{A.3}$$

$$\frac{1 - x_W - y_W}{y_W} = \frac{z_W}{y_W} = \frac{Z_W}{\frac{1}{X_W + 1 + Z_W}\left(X_W + 1 + Z_W\right)} = Z_W. \tag{A.4}$$

Die Projektion lässt sich andererseits darstellen durch:

$$x_R = \frac{X_R}{X_R + Y_R + Z_R} \quad \Rightarrow \quad X_R = c_R x_R, \tag{A.5}$$

$$y_R = \frac{Y_R}{X_R + Y_R + Z_R} \quad \Rightarrow \quad Y_R = c_R y_R, \tag{A.6}$$

$$z_R = \frac{Z_R}{X_R + Y_R + Z_R} \quad \Rightarrow \quad Z_R = c_R z_R. \tag{A.7}$$

Entsprechend ergeben sich die Konstanten c_G und c_B.Durch Einsetzen der Formeln A.2 – A.7 in das Gleichungssystem A.1 ergibt sich:

$$\begin{aligned} x_r c_r + x_g c_g + x_b c_b &= \frac{x_W}{y_W} \\ y_r c_r + y_g c_g + y_b c_b &= 1 \\ z_r c_r + z_g c_g + z_b c_b &= \frac{1 - x_W - z_W}{y_W} \end{aligned}$$

Wird dieses Gleichungssystem nach c_R, c_G, c_B aufgelöst, sind die Normfarbwerte der Primärvalenzen nach Einsetzen in A.5 bis A.7 bestimmt.

A.3.2 Transformation von RGB nach SML

Die Umrechnung von Farben, die ein Normalsichtiger wahrnimmt, auf den Farbraum eines Dichromaten kann durch eine Transformation in den SML-Raum erfolgen (Abschnitt 5.6.2). Dieser Raum wird durch die Farbrezeptoren des menschlichen Auges (S, M, L) aufgespannt.

Brettel et al. (1997) beschreiben eine solche Umrechnung für einen bestimmten Monitor: Die spektralen Strahlungsverteilungen des Monitors werden gemessen und daraus die maximale Leuchtdichte der Primärvalenzen bestimmt. Diese sind die Parameter einer affinen Transformation des Primärvalenzsystems des Monitors (RGB) in das SML-System. In Letzterem erfolgt dann eine Projektion der Farben des Raumes in die Farbebene eines Dichromaten (vgl. Abschnitt 5.6.2).

Ein vereinfachtes Vorgehen, das ohne Ausmessung der Primärvalenzen auskommt und stattdessen von deren Normfarbwertanteilen ausgeht (vgl. Abschnitt 5.3.2), beschreiben die gleichen Autoren in Viénot et al. (1999) für die häufigsten Formen der Farbsehschwäche, die Protanomalie bzw. Protanopie und die Deuteranomalie bzw. Deuteranopie.

Die Transformation eines RGB-Farbraums in den SML-Raum erfolgt dabei über das Normvalenzsystem XYZ; die Umrechung von RGB in XYZ wurde bereits im Abschnitt 5.3.2 beschrieben. An dieser Stelle werden deshalb lediglich Transformationsparameter für die Überführung des Normvalenzsystems in den SML-Raum und die Projektion der Farben in die Farbebenen der genannten Dichromaten angegeben. Für die Rahmenbedingungen bzw. Annahmen, unter denen diese Simulation gültig ist, und die Ausführung möglicher Fehlerquellen wird auf die Beschreibungen in Viénot et al. (1999) verwiesen.

Die Transformation von Normfarbwerten in den SML-Raum erfolgt durch

$$\begin{pmatrix} S \\ M \\ L \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0{,}01608 \\ -0{,}15514 & 0{,}45684 & 0{,}03286 \\ 0{,}15514 & 0{,}54312 & -0{,}03286 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \qquad \text{(A.8)}$$

Eine weitere lineare Transformation reduziert die Farben eines Normalsichtigen in die Farbebene eines Dichromaten (vgl. Abschnitt 5.6.2). Für Protanope ist dies die SM-Ebene, für Deuteranope die SL-Ebene:

$$\begin{pmatrix} S_p \\ M_p \\ L_P \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2{,}52581 & 2{,}02344 & 0 \end{pmatrix} \cdot \begin{pmatrix} S \\ M \\ L \end{pmatrix} \text{ (Protanop),}$$

$$\begin{pmatrix} S_p \\ M_d \\ L_d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1{,}24827 & 0 & 0{,}494207 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} S \\ M \\ L \end{pmatrix} \text{ (Deuteranop).}$$

Die Transformation dieser Werte zurück in das Normvalenzsystem erfolgt über die inverse Matrix aus Formel A.8:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 67{,}33257 & -3{,}50098 & 2{,}94481 \\ 18{,}39253 & 1{,}00004 & 1{,}00004 \\ 62{,}18906 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} S \\ M \\ L \end{pmatrix}.$$

A.4 Metrische Räume und Norm

Der Begriff des Abstands ist auf allgemeinen *metrischen Räumen* gegeben, angewandt auf reelle Vektorräume folgt daraus der Begriff der *Norm*. Häufig gebrauchte Abstände auf dem $\Re^n$ sind durch die L_p*–Norm* definiert. Sofern nicht anders angegeben, sind die folgenden Inhalte angelehnt an Forster (Forster 1984).

A.4.1 Metrik

Unter einer *Metrik* auf einer Menge X versteht man eine Abbildung

$$\begin{aligned} d : X \times X &\to \Re \\ (x, y) &\mapsto d(x, y) \end{aligned}$$

mit folgenden Eigenschaften:

a) Es gilt $d(x, y) = 0$ genau dann, wenn $x = y$.

b) Für alle $x, y \in X$ gilt $d(x, y) = d(y, x)$ (Symmetrie).

c) Für alle $x, y, z \in X$ gilt $d(x, z) \le d(x, y) + d(y, z)$ (Dreiecksungleichung).

A.4.2 Metrischer Raum und Abstand

Ein Paar (X, d), bestehend aus einer Menge X und einer Metrik d, heißt *metrischer Raum*. $d(x, y)$ wird als *Abstand* oder *Distanz* der Punkte x und y bzgl. der Metrik d bezeichnet. Statt $d(x, y)$ wird häufig auch $\|x, y\|$ geschrieben.

A.4.3 Norm und Normierter Vektorraum

Unter einer *Norm* auf einem reellen Vektorraum V wird eine Abbildung

$$\begin{aligned} \| \| : V &\to \Re \\ x &\mapsto \|x\| \end{aligned}$$

mit folgenden Eigenschaften verstanden:

a) $\|x\| = 0$ genau dann, wenn $x = 0$.

b) $\|\lambda x\| = |\lambda| \cdot \|x\|$ für alle $\lambda \in \Re, x \in V$.

c) $\|x + y\| \le \|x\| + \|y\|$ für alle $x, y \in V$.

Ein Paar $(V, \| \|)$, bestehend aus einem Vektorraum V und einer Norm $\| \|$ auf V, heißt *normierter Vektorraum*.

Sei $(V, \| \|)$ ein normierter Vektorraum. Durch $d(x, y) := \|x - y\|$ für $x, y \in V$ wird eine Metrik d auf V definiert.

Sei V der $\Re^n$ und p eine reelle Zahl ≥ 1. Für Vektoren $x = (x_1, \ldots, x_n) \in \Re^n$ ist die $L_p - Norm$ definiert durch (Boyd & Vandenberghe 2004)

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}.$$

Verbreitete Formen der $L_p - Norm$ sind durch $p = 1, 2, \infty$ gegeben (Boyd & Vandenberghe 2004):

- $p = 1$ ($L_1 - Norm$ oder Manhattan-Norm): $\|x\|_1 = |x_1| + \ldots + |x_n|$,
- $p = 2$ ($L_2 - Norm$ oder Euklidische Norm): $\|x\|_2 = \sqrt{< x, x >} = \sqrt{x_1^2 + \ldots + x_n^2}$, worin $< x, x >$ das Skalarprodukt bezeichnet. Abgeleitet aus der Euklidischen Norm ist der Euklidische Abstand $d(x, y) = \|x - y\|_2 = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$,
- $p = \infty$ ($L_\infty - Norm$ oder Chebyshev-Norm): $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$.

A.5 Konvergenzraten

Für die Charakterisierung iterativer Lösungsverfahren werden einige Begriffe zur Konvergenz von Folgen unterschieden:

Eine Folge $\left\{x^{(k)}\right\} \subset \Re^n$ *konvergiert lokal* gegen einen Grenzwert $\bar{x}$, wenn der Startpunkt $x^{(0)}$ der Folge hinreichend nahe an $\bar{x}$ liegt; eine Folge *konvergiert global* gegen einen Grenzwert $\bar{x}$, wenn für die Konvergenz der Folge die Lage von $x^{(0)}$ zu $\bar{x}$ nicht von Bedeutung ist (vgl. Fletcher 1987).

Für die Konvergenzraten wird unterschieden (Alt 2002):

Es sei $\left\{x^{(k)}\right\} \subset \Re^n$ eine Folge mit Grenzwert $\bar{x}$. Die Folge *konvergiert linear* gegen $\bar{x}$, wenn es ein $0 < L < 1$ gibt mit

$$\| x^{(k+1)} - \bar{x} \|_2 \leq L \| x^{(k)} - \bar{x} \|_2$$

für $k \geq 0$. Die Folge *konvergiert quadratisch* gegen $\bar{x}$, wenn es ein $c > 0$ gibt mit

$$\| x^{(k+1)} - \bar{x} \|_2 \leq c \| x^{(k)} - \bar{x} \|_2^2$$

für $k \geq 0$. Die Folge *konvergiert superlinear* gegen $\bar{x}$, wenn

$$\lim_{k \to \infty} \frac{\| x^{(k+1)} - \bar{x} \|_2}{\| x^{(k)} - \bar{x} \|_2} = 0$$

ist.

Danksagung

An erster Stelle möchte ich mich herzlich bei Professor Dr. Lutz Plümer für die Betreuung und Begutachtung dieser Arbeit bedanken. Er stand mir in vielen ergiebigen Diskussionen mit Anregungen, konstruktiver Kritik und Rat zur Seite. Seine vorbehaltlose Unterstützung gab mir die Freiheit, die für die Erstellung einer Dissertation nötig ist.

Professor Dr.-Ing. Wolfgang Förstner danke ich für die Übernahme des Korreferats und seine wertvollen Anmerkungen und Anregungen.

Ebenfalls bedanken möchte ich mich bei allen Mitarbeitern des Bereichs Geoinformation des Instituts für Geodäsie und Geoinformation für die hervorragende Arbeitsatmosphäre und die unzähligen größeren und kleineren Hilfestellungen inhaltlicher und technischer Art. Besonders danke ich Privatdozent Dr. Gerhard Gröger, den ich immer um Rat fragen und in vielen Diskussionen Details der Arbeit darlegen durfte. Seine kompetente Meinung hat sehr zum Gelingen dieser Arbeit beigetragen.

Ich danke M. S. Bernhard Jenny, der mir bei einem Besuch an der ETH Zürich viele Tipps und Anknüpfungspunkte zum Thema Farbsehschwächen gegeben hat.

Für die tatkräftige Hilfe bei der Korrektur der Arbeit bedanke ich mich bei Dr. Yvonne Hilgers.

Meinen Freunden und meiner Familie möchte ich dafür danken, dass sie meine Launen während der Erstellung dieser Arbeit geduldig ertrugen und immer zu mir standen.