Intelligente Bildersuche durch den Einsatz inhaltsbasierter Techniken

Thomas Käster

Dipl.-Inform. Thomas Käster
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: tkaester@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 19.01.2005 vorgelegt von Thomas Käster,
am 09.05.2005 verteidigt und genehmigt.

Gutachter:

Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Helge Ritter, Universität Bielefeld

Prüfungsausschuss:

Prof. Dr. Ralf Möller, Universität Bielefeld
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Helge Ritter, Universität Bielefeld
Dr. Stefan Kopp, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Intelligente Bildersuche durch den Einsatz inhaltsbasierter Techniken

Der Technischen Fakultät der Universität Bielefeld

zur Erlangung des Grades

Doktor-Ingenieur

vorgelegt von

Thomas Käster

Bielefeld – Januar 2005

Für meine Eltern

Danksagung

An dieser Stelle möchte ich mich bei all denen bedanken, die mir mit ihrer Unterstützung die Anfertigung dieser Arbeit ermöglicht haben.

Mein besonderer Dank gilt Gerhard Sagerer, der mich zur Promotion ermutigt und meine Arbeit stets vorbehaltlos unterstützt hat. Als Leiter der Arbeitsgruppe Angewandte Informatik an der Technischen Fakultät hat er ein Arbeitsumfeld geschaffen, das ideale Bedingungen für eine Promotion bietet, wobei nicht nur wissenschaftliche Aspekte hervorragend gefördert werden, sondern ebenfalls die menschliche Komponente nie vernachlässigt wird. Helge Ritter danke ich sowohl für die Begutachtung der Arbeit als auch für die gute Zusammenarbeit im Rahmen des LOKI Projekts. Die projektbezogenen Diskussionen mit ihm haben mich immer begeistert und in meiner Arbeit vorangetrieben.

Des Weiteren bleibt mir die gute Zusammenarbeit mit meinen Kollegen der Angewandten Informatik in sehr guter Erinnerung. Die wissenschaftlichen Gespräche mit ihnen haben entscheidend zum Gelingen dieser Arbeit beigetragen. Dabei möchte ich insbesondere Franz Kummert erwähnen, der stets bereit war, fachliche Details zu erörtern und konstruktive Vorschläge zur Fertigstellung dieser Arbeit zu geben. Meinem Kollegen Michael Pfeiffer danke ich für die hervorragende Zusammenarbeit in den vergangenen Jahren und für die sehr angenehme Arbeitsatmosphäre in unserem Büro. Im Laufe der Jahre hat sich eine Freundschaft entwickelt, die hoffentlich lange bestehen bleibt. Außerdem danke ich Christian Bauckhage für die wertvollen Hinweise, die er mir während der Anfertigung dieser Dissertation gegeben hat. Meinen Arbeitskollegen Marc Hanheide, Marcus Kleinehagenbrock, Thomas Plötz und Volker Wendt bin ich für die kritische Durchsicht und die nützlichen Anmerkungen dankbar, die sicherlich zur Verbesserung der Arbeit beigetragen haben. Bei Lisabeth van Iersel möchte ich mich besonders für ihre Hilfe in bürokratischen Angelegenheiten bedanken.

Ein großer Dank gebührt auch meinen Eltern, die mich stets unterstützen und mir somit diesen Werdegang erst ermöglicht haben. Darüber hinaus danke ich Nadine für ihre liebevolle Unterstützung, ihr Verständnis und ihre Geduld.

Inhaltsverzeichnis

Notation

Es wird versucht, in der vorliegenden Arbeit eine konsistente Notation beizubehalten. Vektoren werden durch fett gedruckte Kleinbuchstaben dargestellt (z.B. $\mathbf{r}$, $\mathbf{x}$) und repräsentieren grundsätzlich Spaltenvektoren. Dementsprechend sind Zeilenvektoren als transponiert gekennzeichnet (z.B. $\mathbf{r}^T$). Matrizen werden als fett gedruckte Großbuchstaben dargestellt (z.B. $\mathbf{W}$). Die wichtigsten Symbole sind in der folgenden Tabelle aufgelistet:

Symbol	Bedeutung
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}^N$	N-dimensionaler Vektorraum der reellen Zahlen
$\|\mathbf{x}\|$	euklidische Norm von $\mathbf{x}$
$d(\mathbf{x}, \mathbf{y})$	Abstand zweier Vektoren $\mathbf{x}$ und $\mathbf{y}$
$\mathbf{x} \oplus \mathbf{y}$	Verkettung der Vektoren $\mathbf{x}$ und $\mathbf{y}$
$\mathbf{e}_i$	(normierter) Einheitsvektor in Richtung i
$\sphericalangle(\mathbf{x}, \mathbf{y})$	Winkel zwischen den Vektoren $\mathbf{x}$ und $\mathbf{y}$
$\mathbf{A} = [a_{mn}]_{M,N}$	Matrix A der Größe $M \times N$ mit den Einträgen a_{mn}
$\det(\mathbf{A})$	Determinante von $\mathbf{A}$
$\mathbf{E}$	Einheitsmatrix
$\boldsymbol{\mu}$	Mittelwertsvektor
$\mathbf{C}$	Kovarianzmatrix
$\mathbf{C}^{-1}$	inverse Kovarianzmatrix
$\boldsymbol{\phi}$	Eigenvektor
$\boldsymbol{\Phi}$	Eigenvektormatrix
λ	Eigenwert
$\boldsymbol{\Lambda}$	Eigenwertmatrix
$\mathcal{O}_k$	k-tes Bildobjekt der Datenbank
$\mathbf{r}_j^k$	j-ter Repräsentant des k-ten Bildobjekts
$\mathcal{Q}$	Anfrage- oder auch Beispielobjekt
$\mathbf{q}_j$	j-ter Anfragevektor des Anfrageobjekts
$\mathbf{W}$	Gewichtsmatrix
$d(\mathbf{x}, \mathbf{y}, \mathbf{W})$	mit $\mathbf{W}$ gewichteter Abstand der Vektoren $\mathbf{x}$ und $\mathbf{y}$
$S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$	Menge S mit den Elementen $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$
$\mathrm{card}(S)$	Kardinalität der Menge S
$\mathrm{card}\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$	Kardinalität der Menge $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$

1 Warum inhaltsbasierte Bildersuche?

Stellen Sie sich vor, Sie arbeiten in einer Werbeagentur. Ihre Aufgabe ist es, einen täglichen visuellen Stimulus zu entwerfen, der uns sowohl in Form von Hochglanzwerbungen in Zeitschriften als auch durch Großflächenwerbungen in den städtischen Einkaufspassagen erreicht. Ihr Arbeitsprozess besteht dabei unter anderem aus dem Durchsuchen von umfangreichen Bildkatalogen. Sie recherchieren nach thematisch eingegrenzten Bildern und Bildelementen, die in weiteren Verarbeitungsschritten zu einem neuen Gesamtwerk zusammengefügt werden.

Ebenso gut könnten Sie aber auch ein Journalist sein, der für die kommende Fußballweltmeisterschaft im eigenen Land eine Reportage über vergangene Weltmeisterschaftserfolge der deutschen Fußballnationalmannschaft schreiben soll. Neben Daten und Fakten sollte die Reportage auch Bilder enthalten, um die größten Erfolge auch optisch zu untermalen. Welche Techniken stehen Ihnen zur Verfügung, das Bildarchiv Ihrer Zeitung nach Bildern der Weltmeisterschaften von 1954, 1974 und 1990 zu durchsuchen? Wie navigieren Sie in dem umfangreichen und reichhaltigen Datenbestand?

Die beschriebenen Szenarien sind klassische Anwendungsfälle, in denen große Datenbestände von Bildern die Basis eines Arbeitsprozesses darstellen und unter Berücksichtigung gewisser Kriterien durchsucht werden müssen. Eine strukturierte Datenhaltung ist dabei ebenso wichtig wie der einfache Zugang zum gespeicherten Datenbestand.

Datenbanksysteme ermöglichen die einfache Verwaltung umfangreicher, vorwiegend textueller Daten und stellen Techniken zur Verfügung, darin schnell und effizient suchen zu können. Die einfachste Möglichkeit, ein Bilddatenbanksystem zu realisieren, ist daher ein textbasierter Ansatz. Bildinhalte werden manuell erfasst und in der Datenbank gespeichert. Die textuellen Annotationen dienen schließlich als Grundlage der Suchanfrage und ermöglichen das Auffinden von Bildern verschiedener semantischer Inhalte. Neben diesen stark manuell gestützten Systemen existieren jedoch auch zahlreiche Systeme, deren Grundlage verschiedene Methoden der Bildverarbeitung bilden. Visuelle Eigenschaften wie Farbe, Textur und Form werden algorithmisch extrahiert sowie in der Datenbank gespeichert und ermöglichen die inhaltsbasierte Suche in dem Datenbestand. Somit lassen sich Bilder finden, die ähnliche Merkmale aufweisen oder deren Merkmale spezielle Kriterien erfüllen. Die elementaren Bestandteile inhaltsbasierter Bilddatenbanksysteme sind daher die merkmalsbasierte Repräsentation der verschiedenen Bildinhalte und deren Ähnlichkeitsvergleich.

Die vorliegende Arbeit befasst sich mit den Techniken zur Konstruktion inhaltsbasierter Bildsuchsysteme und demonstriert deren Leistungsfähigkeit am Beispiel des Bilddatenbanksystems INDI [Käm02, Käs03a, Käs04]. Dieses System wurde im Rahmen des LOKI[1] Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“[2] entwickelt, wobei zwei wesentliche Anforderungen an die Applikation gestellt wurden, die den Entwicklungsprozess grundlegend prägten. Einerseits sollte das Bildsuchsystem natürlich und intuitiv zu bedienen sein. Andererseits sollten inhaltsbasierte Suchtechniken zur Navigation in der gespeicherten Datenmenge verwendet werden. Die einfache und natürliche Mensch-System Interaktion kann sowohl durch die Kombination von Sprache und Gestik als auch durch die Integration von Sprach- und Bildverstehenskomponenten erzielt werden. Die inhaltsbasierte Navigation in einem großen Datenbestand digitaler Bilder basiert auf der Verknüpfung von verschiedenen Techniken unterschiedlicher Forschungsbereiche. Methoden der Bildverarbeitung und der Mustererkennung werden ebenso verwendet wie Ansätze aus der Datenbanktechnologie und des maschinellen Lernens.

1.1 Motivation

Die visuelle Wahrnehmung ist eine wichtige und hoch entwickelte Eigenschaft des Menschen. Sie ermöglicht uns das Betrachten eines schönen Gemäldes ebenso wie den Anblick der beeindruckenden Landschaften Kanadas. Noch vor einigen Jahren war der Einsatz einer analogen Fotokamera die gängigste Variante, derartige visuelle Ereignisse im Bild festzuhalten und einen anhaltenden Eindruck vom Erlebten zu bewahren. Obwohl der Aufnahmeprozess ebenso einfach wie der einer aktuellen digitalen Bildkamera ist, sind die mit der analogen Bildaufnahme verbundenen Arbeitsschritte aufwendig, kostspielig und vor allem zeitintensiv. Um in den Genuss des Aufnahmeergebnisses zu kommen, muss der Film zuerst zum Entwickeln in ein Fotostudio gebracht werden. Dieser Verarbeitungsprozess nimmt einige Zeit in Anspruch, sodass die entwickelten Hochglanzbilder häufig erst nach ein paar Tagen abgeholt werden können. Noch im Fotostudio werden die Resultate auf ihre Qualität untersucht und die schlechten von den guten Aufnahmen getrennt. Aufbewahrt werden die fertigen Bilder meistens in einem Fotoalbum, wo sie durch ergänzende Kommentare, wie z.B. Zeitpunkt oder Ort der Aufnahme, dokumentiert werden.

Durch die rasante Entwicklung im Multimediabereich werden derartige Prozesse stark vereinfacht. Im Gegensatz zur analogen Kamera können die mit einer aktuellen, digitalen Fotokamera aufgenommenen Bilder direkt auf dem Kameradisplay angeschaut

[1] Bei LOKI handelt es sich um ein BMB+F Verbundprojekt mit dem Titel „**L**ernen zur **O**rganisation **k**omplexer Systeme der **I**nformationsverarbeitung“. Das Projekt startete im Juni 2000 und endete nach drei Jahren Laufzeit.

[2] Das Teilprojekt ist eine Kooperation der Arbeitsgruppen Angewandte Informatik und Neuroinformatik der Technischen Fakultät der Universität Bielefeld.

werden. Schlechte Aufnahmen können daher einfach und sofort gelöscht sowie durch einen erneuten Aufnahmeversuch ersetzt werden. Eine Selektion der Bilder wird demnach direkt vor Ort und nicht erst im Fotostudio vorgenommen. Archiviert werden die Bilder gewöhnlich auf der Festplatte des heimischen Computers oder auf externen Medien wie z.B. CD oder DVD. Die Einfachheit der digitalen Bildaufnahme und der Fortschritt im Bereich der Speichermedien haben dazu geführt, dass die Menge an digitalen Bildern drastisch angewachsen ist. Waren vor Jahren noch Festplatten mit einem Speichervolumen von einigen hundert Megabyte wahre Speichergiganten, sind heutzutage private Festplatten mit einer Kapazität von über hundert Gigabyte eine Selbstverständlichkeit. Aber nicht nur im privaten Bereich werden tagtäglich mehrere Gigabyte von Bilddaten erzeugt, sondern auch in anderen Bereichen wie beispielsweise Medizin, Unterhaltung, Militär und Weltraumforschung. Das Erdbeobachtungssystem der NASA kann beispielsweise an einem Tag eine Bildmenge von einem Terabyte erzeugen [Gud95]. Wie aber können solche Datenmengen einfach und effizient verwaltet werden? Wie können Bilder gefunden werden, die bestimmte Kriterien erfüllen oder zueinander ähnlich sind?

Bilddatenbanken bieten eine einfache Möglichkeit digitale Bilder zu organisieren und stellen Techniken zur Verfügung effizient und einfach in der gespeicherten Datenmenge zu suchen. Die Anfänge der Bilddatenbanksysteme liegen in den 70er Jahren und seitdem haben sie sich zu einem sehr aktiven Forschungsfeld entwickelt. Ihre Hauptimpulse erhielten sie aus den Bereichen der Datenbanktechnologie und des Computersehens [Rui99]. Während die einen Bilddatenbanken aus der Sicht traditioneller, textbasierter Datenbanksysteme erforschten, verfolgten die anderen eine bildbasierte Lösung.

Erste textbasierte Bildsuchsysteme entstanden Anfang der 70er Jahre und dienten beispielsweise dazu, Konflikte beim Registrieren von Markenzeichen zu vermeiden (vgl. z.B. [Mül01]). Weitere Systeme auf der Basis von textuellen Annotationen und Datenbankmanagementsystemen folgten [Cha80, Tam84, Cha88]. Welche Vor- und Nachteile ein textbasiertes Bilddatenbanksystem besitzt, kann durch eine genauere Betrachtung des Beispiels der Reportage über die Erfolge der deutschen Fußballnationalmannschaft verdeutlicht werden.

Um den Gewinn der drei Weltmeisterschaftstitel bildlich dokumentieren zu können, ist es notwendig, die Inhalte der gesuchten Bilder genauer zu spezifizieren. So sind zum Beispiel die gespeicherten Bilder im Archiv von Interesse, auf denen die Nationalmannschaft mit dem Weltmeisterpokal posiert. Neben solchen Gruppenaufnahmen können außerdem Spielszenen oder sogar Torszenen der jeweiligen Endspiele wichtige Bestandteile einer bebilderten Dokumentation darstellen. Eine Anfrageformulierung bestünde dann aus der Kombination verschiedener Schlüsselwörter, die die Inhalte der relevanten Bilder umschreiben. Um beispielsweise Fotos der Finalbegegnung von 1974 zu erhalten, könnte folgende Suchanfrage an ein textbasiertes Bilddatenbanksystem formuliert werden:

*„Zeige mir alle Bilder, auf denen die **deutsche Fußballnationalmannschaft** in einem **Endspiel** gegen die **Niederlande** spielt."*

Als Systemantwort wird die Datenbank alle diejenigen Bilder liefern, für die die spezifizierten Schlüsselwörter *deutsche Fußballnationalmannschaft*, *Endspiel* und *Niederlande* gespeichert wurden. Sollte für ein Bild der Begriff *Holland* statt *Niederlande* erfasst worden sein, hätte dies zur Folge, dass das entsprechende Bild nicht gefunden wird.

Obwohl verschlagwortete[3] Bilddatenbanksysteme den einfachen semantischen Zugriff auf gespeicherte Bilddaten ermöglichen, bleiben zwei große Probleme bestehen:

1. Die manuelle Erfassung der Bildinhalte ist sehr zeitintensiv, aufwendig und bei den gegenwärtigen Datenmengen kaum effizient durchführbar. Aufgrund der Komplexität eines Bildes werden meistens nur so viele Annotationen erfasst, wie sie für eine bestimmte Aufgabenstellung notwendig sind. Eine neue Anwendung führt zwangsläufig dazu, dass eine erneute bzw. überarbeitete Verschlagwortung erstellt werden muss.

2. Aufgrund der individuellen visuellen Wahrnehmung des Menschen werden Bildinhalte unterschiedlich interpretiert. Was für den einen Benutzer ein rotes Auto auf einer Rennstrecke ist, ist für den anderen der Ferrari F2003-GA von Michael Schumacher auf dem Hockenheimring. Der Verschlagwortungsprozess ist demnach stark subjektiv geprägt und schränkt somit die Suchmöglichkeiten in der gespeicherten Datenmenge ein.

Motiviert durch die Probleme textbasierter Bilddatenbanken ist Anfang der 90er Jahre das Interesse an der Entwicklung inhaltsbasierter Bildsuchsysteme gewachsen. In der rein inhaltsbasierten Bildersuche (engl. *Content-Based Image Retrieval*) wird auf die Verwendung von textuellen Annotationen verzichtet. Stattdessen werden Bilder einzig und allein auf der Grundlage ihres visuellen Inhalts indiziert. Inhärente Merkmale wie Farbe, Textur oder Form werden automatisch aus den Bildern extrahiert und als Index in der Datenbank abgelegt. Spezielle Techniken ermöglichen schließlich die Suche nach ähnlichen Bildern sowie die einfache Navigation im gespeicherten Datenbestand [Rui99, Sme00, Lew01, Mül01].

Einem inhaltsbasierten Bildsuchsystem sind allerdings Grenzen gesetzt, deren Ursache in der unterschiedlichen Bildbetrachtung von Mensch und System liegt. Während der Mensch Bildinhalte symbolisch betrachtet und mit ihnen eine Semantik assoziiert, basiert ein rein merkmalsbasiertes System auf der formalen Beschreibung eines Bildes. Im Idealfall korrespondiert diese Bildrepräsentation mit den semantischen Bildinhalten. Gewöhnlich ist dies allerdings nur ansatzweise oder in einem eingegrenzten Anwendungsfeld gegeben. Die Diskrepanz zwischen der formalen Bildrepräsentation des

[3] Unter Verschlagwortung wird das Erfassen textueller Annotationen verstanden.

Systems und der semantischen Bildbetrachtung eines Benutzers wird als semantische Lücke (engl. *Semantic Gap*) bezeichnet. Da es sehr schwierig ist, diese Lücke automatisch zu schließen, bietet es sich an, dass ein Bildsuchsystem von einem Anwender lernt. Durch den Lernprozess wird die formale Bildbeschreibung an die Betrachtungsweise eines Benutzers angeglichen, sodass die für die Suchintention eines Anwenders relevanten Bilder der Datenbank gefunden werden können. Die für den Lernvorgang essentiellen Trainingsbeispiele werden aus der Interaktion zwischen Anwender und Bilddatenbanksystem erworben. Aktuelle Bildsuchsysteme sind daher interaktive Systeme, die ausgehend von Benutzerbewertungen die Fähigkeit besitzen, zu lernen und sich an die Suchintention eines Benutzers zu adaptieren.

Durch die Entwicklung inhaltsbasierter Bildsuchsysteme eröffnen sich neue Aspekte und Lösungen. Bilddatenbanken werden dadurch aus einer neuen Perspektive betrachtet und bilden ein weit gefächertes und heterogenes Forschungsfeld, in dem Techniken aus verschiedenen Bereichen miteinander kombiniert werden, um leistungsfähige Systeme zu entwickeln. Methoden der Bildverarbeitung werden ebenso eingesetzt wie Verfahren aus der Mustererkennung und der Datenbanktechnologie. Techniken des maschinellen Lernens ermöglichen die Konstruktion adaptiver Systeme, während Methoden der Mensch-Maschine Interaktion das Design von einfach und natürlich zu bedienenden Anfrageschnittstellen erlauben.

Für spezielle Anwendungen wie Zugangskontroll- und Überwachungssysteme sind inhaltsbasierte Techniken essentiell. Spezielle Verfahren der Bildverarbeitung und Mustererkennung ermöglichen den Vergleich unterschiedlicher Aufnahmen von Fingerabdrücken und Gesichtern [Pen94, Mal03, Zha03]. Ähnliche biometrische Charakteristika können somit gefunden und Personen gegebenenfalls identifiziert werden. Eine derartige Personenidentifikation ist nicht nur zur Verbrechensprävention vorteilhaft, sondern kann darüber hinaus auch Prozesse des täglichen Lebens vereinfachen. So ließe sich z.B. die persönliche Identifikationsnummer (PIN) einer Bankkarte durch einen charakteristischen Fingerabdruck ersetzen. Da ein Kartenbesitzer seine persönliche Kennung als biologisches Attribut immer bei sich trägt, kann er an einem entsprechenden Geldautomaten jederzeit Geld vom eigenen Konto abheben. Bei dem gegenwärtig üblichen PIN-Systemen ist dies nur der Fall, wenn er seine persönliche Identifikationsnummer nicht vergessen hat. Außerdem erfordert dieses Verfahren eine gewissenhafte Handhabung der Geheimzahl, sodass einem Kartenmißbrauch vorgebeugt wird. Eine biometrische Identifikation ist demgegenüber viel sicherer und vor allem viel einfacher zu handhaben, da sich der Nutzer keinerlei Information merken muss.

Je leistungsfähiger die Techniken und Verfahren der inhaltsbasierten Suche werden, desto mehr Anwendungsfelder eröffnen sich. Durch den Einsatz der visuellen Ähnlichkeitssuche kann beispielsweise der Prozess der medizinischen Diagnostik unterstützt werden. Pathologen können Gewebeschnitte vergleichen und Dermatologen Hautbilder auf Krebs oder andere Krankheiten untersuchen [Mül01]. Auch die Verwaltung von Fotografien von Kunstobjekten, wie z.B. Gemälde und Skulpturen, in digitalen Bildka-

talogen von Museen und Kunstgalerien kann durch den Einsatz inhaltsbasierter Techniken unterstützt werden (vgl. z.B. [Ser98]). Darüber hinaus bietet ein inhaltsbasiertes Bildsuchsystem durch seine visuelle Bildrepräsentation und dem damit verbundenen visuellen Zugang zur gespeicherten Datenmenge die Möglichkeit, die Prozesse eines Onlineshops zu vereinfachen. Anstatt die komplette Produktpalette manuell oder auf der Grundlage einiger weniger Schlüsselwörter zu durchsuchen, kann die Navigation in dem Produktkatalog durch die Spezifikation visueller Eigenschaften des gesuchten Objekts beschleunigt werden. Dies ist besonders dann sinnvoll, wenn eine konkrete visuelle Vorstellung vom gesuchten Produkt existiert, diese aber nur unzureichend verbal beschrieben werden kann.

1.2 Zielsetzung und Gliederung der Arbeit

Obwohl textbasierte Bilddatenbanken einen einfachen semantischen Zugang zum gespeicherten Datenbestand ermöglichen, ist es gerade der enorme Umfang der zu verwaltenden Bilddaten und der mangelnde visuelle Zugang, der die Entwicklung alternativer Bildsuchsysteme motiviert. Es ist eine automatische Verarbeitung erforderlich, die neben dem Prozess der Datenbankinitialisierung auch das Durchsuchen des Datenbestandes vereinfacht. Inhaltsbasierte Systeme verfolgen diese Zielsetzung und stellen Techniken zur Verfügung, die auf der Grundlage automatisch extrahierter Bildcharakteristika die Navigation in der gespeicherten Datenmenge ermöglichen. Die bestehende Diskrepanz zwischen den unterschiedlichen Bildbetrachtungsweisen von Mensch und System erfordert allerdings, dass ein Bildsuchsystem lernfähig ist und sich an die Suchintention eines Benutzers adaptieren kann. Es sollte also erkennen, welche Eigenschaften die gesuchten Bilder besitzen, sodass die in der Datenbank gespeicherten Bilder gefunden werden können, die gemäß der Suchintention eines Anwenders relevant sind.

Das Ziel dieser Arbeit ist die Entwicklung der für das INDI System zur Bildersuche benötigten inhaltsbasierten Techniken, die einem Anwender die merkmalsbasierte Navigation in der gespeicherten Datenmenge ermöglichen. Dabei soll das System lernfähig sein und die Eigenschaft besitzen, sich an die Suchintention eines Anwenders zu adaptieren. Die Verwaltung umfangreicher Datensätze erfordert außerdem die Berücksichtigung der speziellen Aspekte der Skalierbarkeit, sodass auch der Fragestellung nachgegangen wird, inwieweit die adaptiven Techniken auf große Datenmengen skalierbar sind.

Die vorliegende Arbeit ist wie folgt strukturiert: In Kapitel 2 wird der aktuelle Stand der Forschung der inhaltsbasierten Bildersuche beschrieben. Dabei werden grundlegende Anfrageparadigmen erläutert und verschiedene Verfahren zur Bildrepräsentation vorgestellt. Außerdem werden einige Abstandsmaße präsentiert, die sich zur Ähnlichkeitsbestimmung zweier Bilder eignen. Die unterschiedlichen Arten der Bildbe-

trachtung von Mensch und System werden ebenso beschrieben sowie die daraus motivierten adaptiven Suchmechanismen, die dazu dienen, die semantische Lücke auf der Basis von Mensch-Maschine Interaktion zu verringern. Abschließend werden repräsentativ einige inhaltsbasierte Bildsuchsysteme vorgestellt.

Das im Rahmen des BMB+F Teilprojekts entwickelte Bildsuchsystem INDI bildet den Schwerpunkt des 3. Kapitels. Dort werden sowohl die Anforderungen an das Bildsuchsystem erläutert als auch seine verschiedenen Bestandteile vorgestellt. Das als verteilte Anwendung entwickelte Bildsuchsystem besteht neben dem Datenbank-Backend und dem zentralen Applikations-Server aus dem Datenbank-Client, der die multimodale Interaktion mit dem Bildsuchsystem ermöglicht. Neben den Aspekten der Datenhaltung und Datenrepräsentation werden außerdem die Erweiterung des SQL Sprachumfangs und der Prozess der Datenbankinitialisierung vorgestellt. Komplettiert wird das Kapitel durch die Beschreibung des System-Servers, des grafischen Datenbank-Clients sowie der verfügbaren Modalitäten zur natürlichen Mensch-Maschine Interaktion.

Kapitel 4 beginnt mit einer formalen Einführung des in INDI realisierten Suchschemas und der Erläuterung der distanz- und rangbasierten Bildersuche. Aufbauend auf der formalen Modellierung werden verschiedene Varianten des Systemlernens vorgestellt. Die theoretische Grundlage des Lernprozesses bildet ein Optimierungsansatz, der von der Zielsetzung geleitet wird, die Abstände der Beispielobjekte zum idealen Anfrageobjekt zu minimieren. Zusätzlich wird neben der Berücksichtigung spezieller numerischer Aspekte auch ein Lernansatz beschrieben, der neben klassifizierten Bildern auch unklassifizierte Bilder der Datenbank zur Systemadaption verwendet. Abschließend werden in einer Evaluation die verschiedenen Ansätze miteinander verglichen und die experimentellen Ergebnisse diskutiert.

Die für die Skalierbarkeit auf große Datenbestände notwendige multidimensionale Indizierung ist Thema des 5. Kapitels. Darin werden neben der Vorstellung etablierter multidimensionaler Indizierungsstrukturen verschiedene Clusterverfahren zur Indizierung vorgestellt. Aufbauend auf den theoretischen Ausführungen wird der für das INDI System entwickelte Indizierungsmechanismus erläutert. Das Kapitel endet mit einer Analyse des Skalierungsverhaltens.

Die Arbeit schließt in Kapitel 6 mit einer Zusammenfassung und einem Ausblick auf potentielle Anschlußarbeiten. Im Anhang werden neben der mathematischen Motivation für die Nebenbedingungen des Optimierungsansatzes die für diese Arbeit relevanten Farbräume vorgestellt und das Verfahren der Hauptachsentransformation beschrieben. Komplettiert wird der Anhang durch weitere Ergebnisse der in Kapitel 4 beschriebenen experimentellen Untersuchungen.

2 Techniken für inhaltsbasierte Bildersuche

Die wichtigste Eigenschaft der inhaltsbasierten Bildersuche ist die automatische Extraktion inhärenter Bildcharakteristika und deren Verwendung zur Navigation in der gespeicherten Bildmenge. Dabei wird im Gegensatz zu traditionellen Ansätzen auf eine textuelle Beschreibung der verschiedenen Bildinhalte verzichtet. Aus der merkmalsbasierten Bildrepräsentation resultiert ein visueller Zugang zur gespeicherten Datenmenge, der es einem Anwender ermöglicht, auf der visuellen Ebene mit dem Bildsuchsystem zu interagieren. Die Entwicklung des Suchverfahrens wird daher von der Zielsetzung motiviert, Bilder in der Datenbank zu finden, deren visuelle Charakteristika denen der Anfrage ähneln. Merkmalsextraktion und Ähnlichkeitsvergleich sind zwar essentielle Bestandteile eines inhaltsbasierten Bildsuchsystems, aber sie alleine sind noch nicht ausreichend, um leistungsfähige Systeme zu entwickeln. Inhaltsbasierte Bildsuchsysteme stellen ein interdisziplinäres Forschungsfeld dar, in dem Techniken und Algorithmen aus unterschiedlichen Forschungsbereichen miteinander kombiniert werden. Aspekte der Datenhaltung und Datenrepräsentation spielen ebenso eine wichtige Rolle wie die Erweiterung der Bildersuche zu einem interaktiven Prozess, in dem versucht wird, die Suchintention eines Anwenders zu lernen. Bevor in den weiteren Kapiteln dieser Arbeit die Details des Bildsuchsystems INDI erläutert werden, soll in den folgenden Abschnitten eine theoretische Grundlage für das Verständnis und die Funktionsweise der inhaltsbasierten Bildersuche geschaffen werden. Daher werden sowohl der aktuelle Stand der Forschung beschrieben als auch repräsentativ einige bestehende Bilddatenbanksysteme vorgestellt.

2.1 Suchintention und Suchparadigmen

Die Nutzung einer Bilddatenbank setzt voraus, dass der Benutzer eine bestimmte Suchintention verfolgt. Allerdings kann diese Intention stark variieren und von einer eher vagen Vorstellung der gesuchten Bilder bis hin zu konkreten Bildinhalten reichen. Im Allgemeinen werden die folgenden Suchansätze unterschieden [Cox00, Sme00]:

Zielsuche: Der Benutzer beabsichtigt, ein konkretes Bild in der Datenbank zu finden. Der Suchprozess kann nicht durch das Finden anderer Bilder beendet werden, egal wie stark diese Bilder dem gesuchten auch ähneln. In einigen Arbeiten wird dieses Suchschema zur Evaluation inhaltsbasierter Bilddatenbanken eingesetzt (vgl. z.B. [Cox00]

oder [Käs03a]). In der Verwaltung von industriellen oder anderen beliebigen Markenzeichen kann dieser Ansatz beispielsweise dazu verwendet werden, um herauszufinden, ob ein bestimmtes Markenzeichen bereits registriert wurde.

Kategoriesuche: Ziel dieses Suchschemas ist das Auffinden von Bildern, die zu einer bestimmten semantischen Kategorie gehören, wie z.B. Flugzeuge, Landschaften, Autos oder Szenen eines Fußballspiels. Oftmals bildet ein Beispielbild die Grundlage einer Suche und es werden alle diejenigen Bilder gesucht, deren Bildinhalte den Inhalten des Beispielbildes ähneln und die zur selben semantischen Klasse gehören.[1]

Browsing: Der Benutzer durchsucht die Datenbank ohne eine konkrete Zielsetzung. Seine Suchintention ist sehr vage und kann während des Suchprozesses wechseln. Die Navigation in der Datenbank ist in diesem Fall hauptsächlich dadurch motiviert, interessante Bilder zu finden.

Um einem Benutzer die inhaltsbasierte Suche in der gespeicherten Bilddatenmenge zu ermöglichen, wird eine „intelligente“ Anfrageschnittstelle benötigt, die es ihm erlaubt, seine visuelle Vorstellung vom Gesuchten dem System gegenüber als Anfrage zu formulieren. Eine textbasierte Schnittstelle, deren Attribute zwar einfach und direkt auf ein SQL[2] Kommando abgebildet werden können, ist allerdings nur wenig praktikabel, da sie einem Anwender keinen visuellen Zugang zum gespeicherten Datenbestand bietet. Dies ist aber gerade eine der elementaren Eigenschaften eines inhaltsbasierten Bildsuchsystems, die es von verschlagworteten Systemen unterscheidet. Die Anforderungen an die Schnittstelle eines inhaltsbasierten Bilddatenbanksystems sind jedoch sehr vielfältig, da die Art und Weise der Anfrage stark vom Anwendungsgebiet abhängt. Für einen Mediziner sind unter Umständen lokale Bilddetails von besonderem Interesse, während ein Besucher einer virtuellen Kunstgalerie vielleicht eher nach Bildern sucht, die eine bestimmte globale Charakteristik aufweisen.

Die beiden häufigsten Anfrageparadigmen sind in Abbildung 2.1 dargestellt. Die sogenannte Beispielsuche (engl. *Query-By-Example*) basiert auf der initialen Auswahl eines Beispielbildes [Cha80, Sme00, Lew01]. Der Benutzer wählt dabei zu Beginn einer Suche aus einem gegebenenfalls zufällig präsentierten Bildsatz das Bild aus, das seiner Suchintention am besten entspricht. Als Resultat liefert das Bildsuchsystem die Bilder, die dem Beispielbild am ähnlichsten sind. Oftmals ist auch die Auswahl mehrerer initialer Beispielbilder möglich. Neben der Auswahl eines Beispielbildes existiert in vielen Bildsuchsystemen auch die Möglichkeit, eine oder mehrere Beispielregionen in einem Bild auszuwählen. Die Voraussetzung dafür ist allerdings, dass durch die Anwendung eines geeigneten Bildsegmentierungsverfahrens zuvor die notwendigen Bildausschnitte wie Regionen oder sogar Objekte extrahiert wur-

[1] Dabei ist zu beachten, dass eine visuelle Ähnlichkeit zweier Bilder nicht automatisch deren Zugehörigkeit zur selben semantischen Klasse impliziert. Auf diese Problematik und die Dehnbarkeit des Ähnlichkeitbegriffs wird im Rahmen der vorliegenden Arbeit wiederholt eingegangen.

[2] SQL (engl. *Structured Query Language*): Etablierte und standardisierte Anfragesprache für relationale Datenbanksysteme (vgl. z.B. [Lan95, Kapitel 22] oder [Mat97, Kapitel 4]).

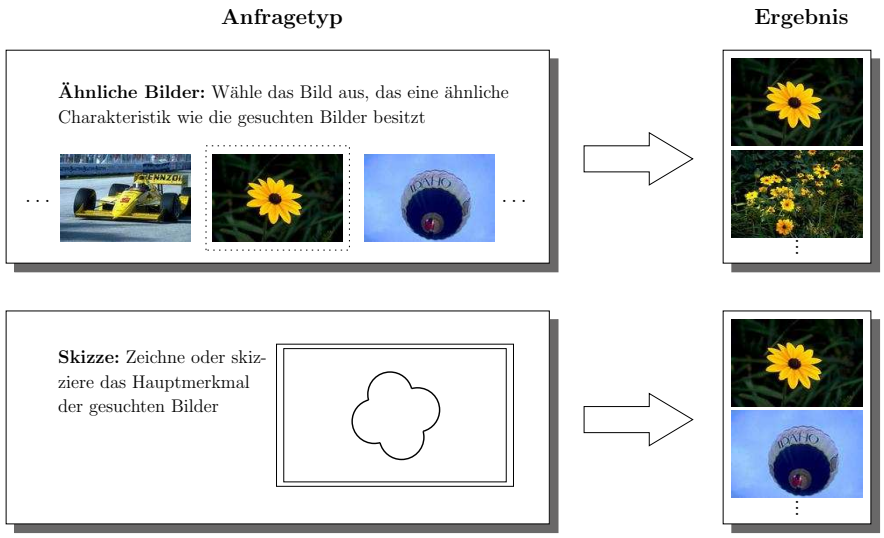


Abb. 2.1: Anfrageparadigmen in der inhaltsbasierten Bildersuche (basierend auf [Lew01]). Die obere Abbildung demonstriert die Suche auf der Basis eines Beispielbildes. Die untere Grafik veranschaulicht die Anfrageformulierung durch Skizzieren.

den [Ma97b, Car99, Mes99, Fau02]. In der Literatur wird dieser Spezialfall der Beispielsuche auch als Regionensuche bezeichnet (engl. *Query-By-Region*).

Falls die gesuchten Bildinhalte bestimmte Formcharakteristika besitzen, bietet sich die in Abbildung 2.1 dargestellte Anfrageformulierung durch Skizzieren an (engl. *Query-By-Sketch*). Diese Art der Anfrage setzt jedoch voraus, dass ein Benutzer Übung im Anfertigen von Skizzen besitzt und die verschiedenen Bildobjekte in den Bildern bei der Erstellung der Datenbank detektiert wurden.[3] Allerdings ist die Objekterkennung eine Aufgabe, für die noch keine generelle Lösung existiert und die nur unter Einschränkung der Anwendungsdomäne zufriedenstellend gelöst werden kann. Dieses Anfrageparadigma ist daher auf Bilddatenbanken beschränkt, in denen das Objekterkennungsproblem entweder manuell oder durch die Einschränkung der Bilddomäne vereinfacht wurde.

Einige Systeme bieten dem Benutzer auch die Möglichkeit, relevante Bildcharakteristika wie Farbe und Textur genauer zu spezifizieren [Fli95, Bac96]. Auf diese Weise können Anfragen wie z.B. „Zeige mir alle Bilder mit 30% grün und 40% blau, wobei die grünen Bereiche eine gras-ähnliche Texturierung besitzen sollen“ formuliert werden. Die Anfrageformulierung erfolgt dabei allerdings nicht auf der textuellen Ebene, sondern auf der Grundlage von visuellen Werkzeugen der grafischen Benutzerschnittstelle, wie beispielsweise Farbregler oder Texturmuster. Die Kombination mit dem

[3]Obwohl mit dem Verfahren des *Template Matching* eine alternative Strategie für die Formsuche existiert, die keine Objekterkennung voraussetzt, ist sie für die inhaltsbasierte Bildersuche viel zu rechenintensiv und daher im Allgemeinen nicht anwendbar.

Query-By-Sketch Paradigma ermöglicht die Suche nach Bildinhalten, die neben bestimmten Farb- und Texturcharakteristika auch bestimmte Formmerkmale besitzen.

Für alle Anfrageparadigmen liefert ein Bilddatenbanksystem in der Regel eine Liste von Bildern, die nach ihrer Relevanz bezüglich der Benutzeranfrage sortiert sind. Die Relevanz ergibt sich aus dem Vergleich von visuellen Merkmalen und erfordert sowohl die automatische Extraktion der inhärenten Bildcharakteristika als auch die Verwendung eines Vergleichsmaßes. In den nächsten Abschnitten werden diese elementaren Bestandteile eines inhaltsbasierten Bildsuchsystems näher erläutert.

2.2 Merkmale und Bildrepräsentationen

Die Extraktion von Bildmerkmalen ermöglicht die Repräsentation visueller Inhalte eines Bildes und bildet somit die Grundlage eines inhaltsbasierten Bildsuchsystems (vgl. Abbildung 2.2). Unterschieden werden generelle und domänenabhängige Merkmale [Rui99, Zho03b]. Während Farbe, Textur und Form zur ersten Kategorie gehören, sind die Elemente der zweiten Kategorie stark anwendungsabhängig und auf spezielle Bildklassen beschränkt, wie z.B. Gesichter und Fingerabdrücke. Da im Rahmen dieser Arbeit die allgemeine Anwendbarkeit der vorgestellten Methoden ein wesentlicher Aspekt ist, wird auf die weitere Betrachtung spezieller Merkmale verzichtet.[4] Stattdessen werden ausschließlich verschiedene generelle Bildcharakteristika vorgestellt.

Motiviert durch verschiedene Aspekte der menschlichen Wahrnehmung wird in der Merkmalsextraktion versucht, die wichtigsten Eigenschaften eines Bildes zu erfassen. Sie sollten sowohl das Auffinden von ähnlichen Bildinhalten ermöglichen als auch unterschiedliche voneinander separieren. Mit der Entwicklung des MPEG-7 Standard [Sik01, ISO02] wurde ein Schema entworfen, das neben der Beschreibung von Videodaten auch die Charakterisierung statischer Bilder ermöglicht. Es werden Algorithmen zur Verfügung gestellt, die die Berechnung verschiedener Repräsentanten der Merkmale Farbe, Textur und Form erlauben. Sie sind in ihrer Anwendung auf keine Bilddomäne beschränkt und ihre Leistungsfähigkeit wurde in verschiedenen Experimenten evaluiert [Man01]. Allerdings stellen sie nur eine Möglichkeit dar, Bildinhalte mathematisch zu beschreiben. Die möglichen Transformationen, die den visuellen Inhalt eines Bildes auf einen beschreibenden Merkmalsvektor abbilden, sind zahlreich. Eine einzelne Repräsentation für ein Merkmal, wie z.B. Farbe, existiert nicht. Vielmehr existieren eine Vielzahl unterschiedlicher Repräsentanten, die ein Merkmal aus unterschiedlichen Perspektiven charakterisieren. Je nach Art ihrer Berechnung werden sie als globale oder lokale Repräsentanten bezeichnet. Während erstgenannte bei der

[4] Der interessierte Leser sei an dieser Stelle auf die spezielle Literatur der Mustererkennung verwiesen, wie z.B. [Pen94], [Mal03] oder [Zha03].

Merkmalsextraktion: Unter dem Prozess der Merkmalsextraktion eines Bildes B wird die automatische Berechnung eines oder mehrerer Bildrepräsentanten R_i verstanden.[a] Formal lässt sich dieser Prozess wie folgt definieren:

$$B \mapsto M_R = \{R_1, R_2, \ldots, R_I\}$$

Dabei kann ein Repräsentant, auch Deskriptor genannt, verschiedene Formen besitzen. Die gebräuchlichste Form ist die Darstellung durch einen Merkmalsvektor $\mathbf{r}_i$:

$$R_i \mathrel{\widehat{=}} \mathbf{r}_i = (r_{i_1}, r_{i_2}, \ldots, r_{i_N})^T, \quad \text{mit } \mathbf{r}_i \in \mathbb{R}^N$$

Jedes Bild der Datenbank lässt sich somit als ein Punkt im N-dimensionalen Merkmalsraum $\mathbb{R}^N$ darstellen. Im Gegensatz zu Merkmalsvektoren besteht eine beschreibende Signatur aus einer Menge von Vektorclustern [Rub01]:

$$R_i \mathrel{\widehat{=}} S_i = \{\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \ldots, \mathcal{S}_{i_{L_B}}\} = \{(\mathbf{p}_{i_1}, w_{\mathbf{p}_{i_1}}), (\mathbf{p}_{i_2}, w_{\mathbf{p}_{i_2}}), \ldots, (\mathbf{p}_{i_{L_B}}, w_{\mathbf{p}_{i_{L_B}}})\}$$

Jeder Cluster $\mathcal{S}_{i_j} = (\mathbf{p}_{i_j}, w_{\mathbf{p}_{i_j}})$ wird durch einen Clusterprotoypen[b] $\mathbf{p}_{i_j} \in \mathbb{R}^N$ sowie die relative Anzahl seiner Elemente $w_{\mathbf{p}_{i_j}} \in \mathbb{R}^+$ beschrieben. Da die Signatur eines Merkmals für jedes Bild individuell definiert ist, variiert ihre Länge L_B mit der Komplexität eines Bildes. Ein Beispiel für eine Signatur, die die Farben eines Bildes repräsentiert, wird in Abschnitt 2.2.1 gegeben.

[a] Repräsentanten können auch für Teilbilder und Bildregionen berechnet werden.
[b] Meistens korrespondiert der Prototyp mit dem Mittelwertsvektor des Clusters.

Abb. 2.2: Mathematische Beschreibung visueller Bildinhalte

Merkmalsberechnung alle Bildpunkte berücksichtigen, ist die Berechnung der lokalen Charakteristika auf die Elemente einer Bildregion beschränkt. In den nächsten Abschnitten werden sowohl einige sogenannte *low-level*[5] Repräsentanten der Merkmale Farbe, Textur und Form vorgestellt als auch verschiedene Verfahren zur Detektion interessanter Bildbereiche beschrieben.

2.2.1 Farbe

Farbe ist ein wichtiger Bestandteil der visuellen Wahrnehmung. Die unterschiedlichen Photopigmente innerhalb der Zapfen in der Fovea des menschlichen Auges ermöglichen uns das Farbensehen und erlauben schon Kleinkindern, zwischen blauen und roten Bauklötzen zu unterscheiden. Farbe ist das am häufigsten eingesetzte Merkmal in

[5] Unter low-level oder auch subsymbolischen Bildrepräsentanten werden Beschreibungen verstanden, die rein pixelbasiert sind und mit denen noch keine abstrakte semantische Interpretation assoziiert ist.

der inhaltsbasierten Suche, nicht zuletzt wegen seiner Unabhängigkeit von Bildgröße und -orientierung. Die Grundlage für die Entwicklung von leistungsfähigen Farbmerkmalen bilden Studien der Farbwahrnehmung und Farbräume (vgl. z.B. [Wys82] oder [Gon02]).

Der verbreitetste Ansatz zur Repräsentation der Farben innerhalb eines Bildes sind Farbhistogramme. In dieser nicht parametrischen Wahrscheinlichkeitsverteilung ist kodiert, wie oft die verschiedenen Farbwerte in einem Bild auftreten. Sie sind sowohl translations- als auch rotationsinvariant und außerdem relativ einfach zu berechnen. Swain und Ballard [Swa91] demonstrieren die Leistungsfähigkeit der histogrammbasierten Bildersuche am Beispiel einer Objektdatenbank und stellen mit dem Histogrammschnitt ein Ähnlichkeitsmaß vor, das den Vergleich zweier Bilder auf der Basis ihrer Farbhistogramme ermöglicht. Die Anzahl der Farben innerhalb eines Bildes ist oftmals auf einen Teil des verfügbaren Spektrums beschränkt. Die Konsequenz sind spärlich besetzte Histogramme, die deshalb sensitiv gegenüber Rauschen sind. Diese Problematik lässt sich durch die Berechnung der von Stricker und Orengo [Str95] verwendeten kumulativen Histogramme lösen. Ausführliche experimentelle Untersuchungen von Farbhistogrammen als Grundlage der inhaltsbasierten Bildersuche finden sich in den Arbeiten von Stricker und Swain [Str94] sowie Zhang et al. [Zha95].

Allen Histogrammvarianten ist die Problematik ihrer hohen Dimensionalität gemeinsam, die durch Anpassung der Farbraumquantisierung auch nur bedingt gelöst werden kann. Um eine schnelle und effiziente Bildersuche garantieren zu können, ist eine kompakte Darstellung der Merkmale erforderlich, deren Informationsgehalt so groß wie möglich sein sollte. Das von Stricker und Orengo [Str95] vorgestellte Verfahren der Farbmomente (engl. *Color Moments*) erfüllt genau diese Kriterien. Ausgehend von der Eigenschaft, dass jede beliebige Farbverteilung durch ihre dominanten Merkmale repräsentiert werden kann, werden die Farbwerte jedes Bildkanals durch die drei Momente Mittelwert, Varianz und Schiefe (engl. *Mean*, *Variance* und *Skewness*) beschrieben. Die daraus resultierende kompakte Farbrepräsentation ermöglicht nicht nur eine schnelle Bildersuche, sondern liefert in der Evaluation von Stricker und Orengo [Str95] auch bessere Ergebnisse als die histogrammbasierte Suche.

Räumliche Anordnung

Die bisher betrachteten auf Farbe basierenden Bildrepräsentanten sind ausschließlich globale Deskriptoren, die auf die Integration struktureller Bildeigenschaften verzichten. Sie beinhalten Informationen darüber, welche Farbwerte in einem Bild enthalten sind, nicht jedoch wo diese Werte im Bild lokalisiert sind oder wie die lokale Nachbarschaft der korrespondierenden Bildpunkte beschaffen ist. Obwohl diese Farbdeskriptoren einfach zu berechnen sind und das Auffinden von farblich ähnlichen Bildern ermöglichen, liefern sie gerade in großen Bilddatenbanken zu viele falsch positive Bil-

Abb. 2.3: Beispiel für ein falsch positives Ergebnisbild. Das System liefert als Antwort auf die initiale Auswahl des Beispielbildes die dargestellte Ergebnisliste. Obwohl alle Bilder eine ähnliche Farbcharakteristik aufweisen, gehört das zweite Bild nicht zur gesuchten Domäne der Sonnenuntergangaufnahmen und wird daher als falsch positiv bezeichnet.

der (vgl. Abbildung 2.3)[6]. Ganz unterschiedlich strukturierte Bilder können ähnliche Farbhistogramme besitzen. Bessere Ergebnisse können erst dann erzielt werden, wenn auch die räumliche Anordnung der Farbwerte berücksichtigt wird. Die daraus resultierende Repräsentation ermöglicht sowohl die Lokalisierung der im Bild enthaltenen Farben als auch eine detaillierte Beschreibung ihrer lokalen Nachbarschaft.

Siggelkow [Sig02] erzielt die Integration lokaler Bildcharakteristika durch die Berechnung invarianter Merkmale. Die Invarianten ergeben sich aus der Mittelung von Funktionswerten, die auf der Basis einer monomialen Kernfunktion innerhalb einer Pixelnachbarschaft bestimmt werden. Die Berechnung erfolgt für jedes Pixel der Ebenen des RGB Farbraumes (vgl. Anhang B.1). Anstelle der Farbwerte werden die invarianten Merkmale zur Konstruktion von Fuzzyhistogrammen verwendet, in denen somit lokale Bildeigenschaften kodiert sind. Pass et al. [Pas96] klassifizieren ein Pixel einer bestimmten Farbe, je nachdem ob es Element einer größeren Farbregion ist oder nicht, entweder als zusammenhängend oder als nicht zusammenhängend. Anstatt der Anzahl der verschiedenen Farbwerte beinhaltet der resultierende Farbkohärenzvektor für jeden quantisierten Farbwert die Anzahl der zusammenhängenden und nicht zusammenhängenden Pixel. Die Ergebnisse demonstrieren, dass durch diese Separierung eine feinere Bildunterscheidung möglich ist als durch die Verwendung der klassischen Farbhistogramme. Die von Huang et al. [Hua97] entwickelte Bildbeschreibung wird als *Color Correlogram* bezeichnet. Die Grundlage dieses Verfahrens bilden Farbwert-Übergangsmatrizen, in denen das Auftreten unterschiedlicher Farben kodiert ist, deren Bildpunkte einen bestimmten Abstand zueinander haben. Aus der Normierung der Einträge der Übergangsmatrizen resultieren schließlich die zur Ähnlichkeitssuche eingesetzten *Correlograms* und *Auto-Correlograms*. Rubner und Tomasi [Rub01] verwenden zur Beschreibung eines Pixels sowohl die dreidimensionale Darstellung im CIE $L^*a^*b^*$ Farbraum als auch seine Bildposition (x- und y-Koordinaten). In einem zweistufigen Clusterprozess, bestehend aus Splittung und Rekombination, werden die

[6]Ein Großteil der in dieser Arbeit abgedruckten Bilder stammt aus der Fotokollektion der „ArtExplosion® 600000 Images“ Bildsammlung der Nova Development Corporation. Ausführlichere Produktinformationen sind unter `http://www.novadevelopment.com` verfügbar.

farbigen Bildpunkte in dem fünfdimensionalen Merkmalsraum gruppiert. Jeder resultierende Cluster wird durch sein Clusterzentrum $\mathbf{p}$ sowie den Anteil der Pixel $w_{\mathbf{p}}$, die zu dem Cluster gehören, repräsentiert. Da die Anzahl der Farben von Bild zu Bild variiert, können die resultierenden Signaturen aus unterschiedlich vielen Komponenten bestehen. Ein Vergleich dieser Charakteristika erfordert daher ein flexibles Ähnlichkeitsmaß, wie z.B. die in Abschnitt 2.3.5 beschriebene *Earth-Mover's*-Distanz.

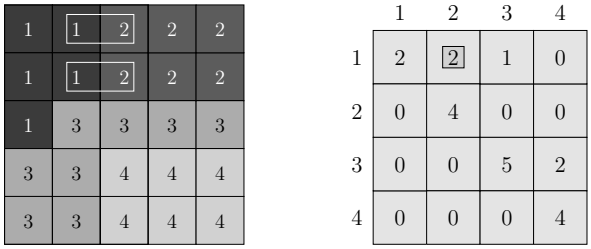
Eine weitere Variante, die räumliche Farbverteilung innerhalb eines Bildes zu erfassen basiert auf der Rasterung des Bildes. In jedem der resultierenden Bildblöcke werden Farbrepräsentanten extrahiert, die als Bildindex in der Datenbank gespeichert werden und das Auffinden ähnlich strukturierter Bilder ermöglichen [Tia00]. Die von Stricker und Dimai [Str97] vorgestellten lokalen Bildbeschreibungen basieren auf der Unterteilung eines Bildes in fünf teilweise überlappende Fuzzyregionen. In jeder dieser Regionen wird sowohl die Durchschnittsfarbe als auch die Kovarianzmatrix der Farbverteilung berechnet. Aufgrund der Überlappung sind die berechneten Merkmalsvektoren robust gegenüber kleinen Translationen und Rotationen von Bildausschnitten. Die Ähnlichkeit zweier Bilder ergibt sich aus dem Vergleich der korrespondierenden lokalen Farbbeschreibungen und der Kombination der Einzelergebnisse.

2.2.2 Textur

Textur ist eine visuelle Eigenschaft von Oberflächen, die es uns ermöglicht, gleichfarbige Flächen zu unterscheiden. Jeder Mensch weiß, dass Zebras gestreift und Kühe gefleckt sind. Auch fällt es uns nicht schwer, gelocktes vom glatten Haar zu unterscheiden. Doch obwohl wir eine Textur sofort erkennen, sobald wir sie sehen, existiert keine allgemeingültige Texturdefinition [Tuc93, Rub01, Seb01]. Gewöhnlich wird mit einer Textur ein visuelles Muster assoziiert, das sich innerhalb einer Bildregion wiederholt und diese daher homogen erscheinen lässt.[7] Eine Textur besitzt also eine räumliche Ausdehnung und ist im Gegensatz zur Farbe nicht an einen einzelnen Bildpunkt gebunden. Stattdessen ist sie durch signifikante Variationen der Intensitätswerte nahegelegener Pixel gekennzeichnet.

Letztendlich sind in einer Textur strukturelle Eigenschaften kodiert, die allerdings von der gewählten Bildskalierung abhängen. Betrachtet man beispielsweise eine Mauer aus großer Entfernung so lässt sich lediglich die regelmäßige Anordnung der Mauersteine wahrnehmen. Steht man jedoch unmittelbar vor der Mauer, so kann auch die feine Struktur der einzelnen Steine erkannt werden. Eine ausführliche Übersicht über verschiedene Verfahren zur Texturbeschreibung findet sich beispielsweise in den Arbeiten von Haralick [Har79], Tuceryan und Jain [Tuc93] oder Sebe und Lew [Seb01]. In den folgenden Abschnitten werden stellvertretend für die Vielzahl der verfügbaren Algorithmen einige Ansätze zur Texturrepräsentation beschrieben.

[7] Auf eine detaillierte Betrachtung grundsätzlicher Strukturen wie reguläre (periodische) oder irreguläre Texturen wird an dieser Stelle verzichtet.



(a) Grauwertbild (b) GÜM

Abb. 2.4: Beispiel einer Grauwertübergangsmatrix (GÜM) (nach Sebe and Lew [Seb01]). (a) Ein auf vier Intensitätswerte quantisiertes Bild. (b) Die korrespondierende GÜM für den Pixelabstand $d = 1$ und die Orientierung $\theta = 0°$. Dies entspricht einer relativen Verschiebung von $(\Delta x, \Delta y) = (1, 0)$. Die markierten Rechtecke veranschaulichen wie für ein Pixelpaar mit den Intensitätswerten 1 und 2 ein Eintrag in der GÜM bestimmt wird.

Haralick et al. [Har73] haben in den frühen 70er Jahren ein Verfahren zur Texturbeschreibung vorgestellt, das auf Grauwertübergangsmatrizen (GÜM, engl. *Gray Level Co-Occurence Matrix*) basiert. Eine GÜM ist eine punktbasierte Statistik zweiter Ordnung, in der Informationen darüber kodiert sind, wie häufig Paare von Pixeln innerhalb eines Bildes oder einer Bildregion auftreten, die bestimmte Grauwerte besitzen. Definiert wird das Pixelpaar sowohl durch einen Pixelabstand d als auch durch eine Orientierung θ. Ein Beispiel für die Berechnung einer Grauwertübergangsmatrix ist in Abbildung 2.4 gegeben. Ausgehend von dieser statistischen Modellierung von benachbarten Intensitäten lassen sich statistische Merkmale berechnen, die die Texturierung eines Bildes oder einer Bildregion charakterisieren, wie z.B. Energie, Entropie, Kontrast oder Homogenität. Zusammengefasst ergeben diese Merkmale den endgültigen Texturdeskriptor. Im Gegensatz zum GÜM-basierten Verfahren verzichtet Unser [Uns86] auf die Konstruktion der Grauwertübergangsmatrizen und berechnet stattdessen für die verschiedenen Pixelpaare Summen- und Differenzhistogramme. Diese ermöglichen ebenfalls die Berechnung der unterschiedlichen Texturstatistiken. Gegenüber der Variante von Haralick et al. [Har73] besitzt dieser Ansatz allerdings den Vorteil, dass er einfacher zu berechnen ist und weniger Speicherplatz erfordert.

Die von Tamura et al. [Tam78] entwickelten Texturrepräsentationen sind durch psychologische Untersuchungen der menschlichen Wahrnehmung motiviert. Aus den Experimenten geht hervor, dass zur Texturbeschreibung die sechs Charakteristika Grobkörnigkeit, Kontrast, Ausrichtung, Linienähnlichkeit, Regularität und Rauheit (engl. *Coarseness*, *Contrast*, *Directionality*, *Linelikeness*, *Regularity* und *Roughness*) eine besonders wichtige Rolle spielen. Die entwickelten Merkmale entsprechen daher mathematischen Beschreibungen dieser grundlegenden Textureigenschaften. Modifi-

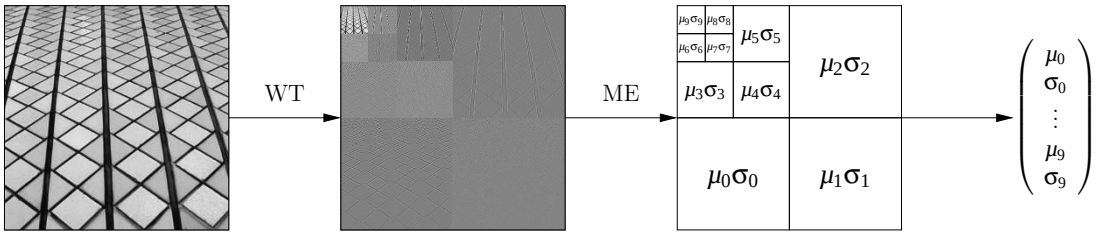


Abb. 2.5: Texturbeschreibung durch Waveletzerlegung. Das Texturbeispiel aus der Brodatz Bildsammlung (*Tile.0001.jpg*) wird durch Wavelettransformation (WT) in Skalierungs- und Detailinformationen zerlegt. Für jede der drei Zerlegungsstufen werden anschließend Mittelwert μ und Standardabweichung σ der Koeffizienten der verschiedenen Energiebänder berechnet. Die aus der Merkmalsextraktion (ME) resultierenden Charakteristika werden abschließend zu einem Merkmalsvektor kombiniert.

zierte Varianten dieser Merkmale werden in den Bildsuchsystemen QBIC [Nib93] und MARS [Hua96, Ort97] zur inhaltsbasierten Bildersuche verwendet.

Ebenfalls motiviert durch die Ergebnisse von psychologischen Untersuchungen existieren weitere Verfahren, die sich auf die Repräsentation der spektralen Eigenschaften einer Textur konzentrieren. Während in einigen Verfahren zur Frequenzanalyse die Bildfilterung mit Gaborfiltern [Man96, Rub01] verwendet wird, basieren andere Ansätze auf der Wavelettransformation [Smi94]. Beide Varianten haben den Vorteil, dass sie eine gegebene Textur für verschiedene Skalierungsstufen analysieren und nicht auf eine spezielle Skalierung beschränkt sind. Eine solche Multiskalenanalyse ist vorteilhaft, da die Skalierungsinformationen der zu repräsentierenden Texturen eines Bildes gewöhnlich a priori nicht bekannt sind.

In dem von Manjunath und Ma [Man96] vorgestellten Verfahren ist eine Filterbank von Gaborfiltern, die jeweils ein bestimmtes Frequenzband erfassen, Ausgangspunkt der Texturbeschreibung. Das Design der Filterbank erfolgt durch Variation der Skalierungs- und Rotationsparameter einer zentralen Gaborfunktion, dem sogenannten Mutterwavelet. Aus der Anwendung der Filterbank resultiert für jedes Pixel eine Menge von Filterantworten bzw. Koeffizienten. Eine kompakte Texturbeschreibung wird durch die statistische Analyse der Filterantworten erzielt. Dabei werden die Koeffizienten eines Frequenzbandes durch den statistischen Mittelwert und die Standardabweichung ihrer Absolutbeträge repräsentiert. Der daraus resultierende kompakte Texturdeskriptor umfasst in der Arbeit von Manjunath und Ma bei 24 Gaborfiltern lediglich 48 Komponenten, die jeweils die verschiedenen Mittelwerte und Standardabweichungen repräsentieren. Aus den ebenfalls in dieser Arbeit durchgeführten experimentellen Untersuchungen geht hervor, dass diese Variante der Texturbeschreibung im Vergleich zu

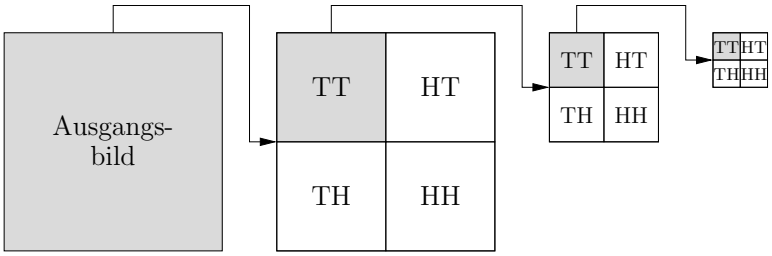


Abb. 2.6: Wavelettransformation nach Zeller [Zel94]. Bei der Wavelettransformation wird ein Bild in Skalierungs- (grau) und Detailinformationen zerlegt. Dies entspricht der sequentiellen Anwendung von speziellen Hoch- (H) und Tiefpaßfiltern (T).

anderen Multiskalen-Techniken die besten Ergebnisse auf der Brodatz Bildsammlung[8] erzielt.

Bei der Wavelettransformation [Mal89, Zel94] wird ein Bild sukzessive in Skalierungs- und Detailinformationen zerlegt. Abbildung 2.5 demonstriert diesen Vorgang anhand eines Bildes aus der Brodatz Bildsammlung [Bro66]. Das Beispiel veranschaulicht außerdem, dass die Waveletzerlegung Filterung und Subsampling beinhaltet. Dies entspricht im wesentlichen der in Abbildung 2.6 dargestellten sequentiellen Anwendung von speziellen Hoch- (H) und Tiefpaßfiltern (T). Die für die inhaltsbasierte Bildersuche notwendige kompakte Bildbeschreibung wird durch die Extraktion statistischer Charakteristika erreicht. Ähnlich wie bei der Gaborfilterung wird die Energieverteilung der Koeffizienten der verschiedenen Energiebänder durch den Mittelwert μ und die Standardabweichung σ repräsentiert [Smi94].

2.2.3 Form

Neben intensitätsbasierten Eigenschaften wie Farbe und Textur werden geometrische Charakteristika wie Formen dazu verwendet, die verschiedenen Bildinhalte zu repräsentieren. Generell werden zwei Arten der Formrepräsentation unterschieden, die als kontur- und regionenbasierte Verfahren klassifiziert werden können.[9] Erstere erfordern die Detektion der Randpixel einer Bildregion und beschreiben den entsprechenden Bildausschnitt ausschließlich auf der Basis dieser Konturelemente. Letztere dagegen beziehen alle Elemente einer Region in die Merkmalsberechnung mit ein. In den folgenden Abschnitten werden verschiedene Verfahren der beiden Kategorien vorgestellt. Da jedoch zahlreiche Algorithmen zur Formrepräsentation existieren und diese nicht alle im Rahmen dieser Arbeit vorgestellt werden können, wird sich auf ei-

[8] Bei der Brodatz Bildsammlung [Bro66] handelt es sich um eine Menge von Texturaufnahmen, die in vielen Arbeiten zur Evaluation von Texturbeschreibungen verwendet wird.

[9] In der Literatur werden alternativ zu kontur- und regionenbasiert ebenfalls die Bezeichnungen extern (engl. *external*) und intern (engl. *internal*) verwendet (vgl. z.B. [Lon98] oder [Bra99]).

nige repräsentative Ansätze beschränkt. Weitere Ausführungen zu verschiedenen Varianten der Formrepräsentation können beispielsweise in den Arbeiten von Mehtre et al. [Meh97], Loncaric [Lon98], Brandt [Bra99] sowie Rui und Huang [Rui99] nachgeschlagen werden. Ein Vergleich von zwei kontur- und regionenbasierten Ansätzen findet sich außerdem in der Arbeit von Zhang und Lu [Zha01b].

Gewöhnlich werden bei der Formbeschreibung die Intensitätswerte einer Region vernachlässigt. Stattdessen wird eine binäre Darstellung bevorzugt, in der alle Pixel innerhalb einer Kontur und auf der umgebenen Kontur durch Einsen und alle Pixel außerhalb der Kontur durch Nullen repräsentiert werden. Eine sehr prominente Variante der regionenbasierten Darstellung basiert auf Momenten. Hu [Hu62] stellt in seiner Arbeit einen Satz von sieben Momenten vor, deren wichtigste Eigenschaft die Invarianz gegenüber affinen Transformationen wie Translation, Rotation und Skalierung ist. Die Grundlage der von Teague [Tea80] präsentierten Zernike-Momente sind Zernike-Polynome, welche im Einheitskreis einen Satz von komplexwertigen, orthogonalen Funktionen bilden. Ebenfalls zur Klasse der momentbasierten Formbeschreibungen gehört die *Angular Radial Transformation* des MPEG-7 Standards [Bob01, Sik01]. Sie ist auf einem in Polarkoordinaten dargestellten Einheitskreis definiert und unterscheidet sich von den Zernike-Momenten in der Wahl der Basisfunktionen. Des Weiteren werden zur regionenbasierten Formbeschreibung neben der Gitterdarstellung [Zha01b] vor allem heuristische Merkmale, wie beispielsweise Fläche, Zirkularität oder Exzentrizität verwendet. Die mehrdimensionale Formrepräsentation folgt aus der Verkettung dieser skalaren Attribute.

Eine populäre Methode zur konturbasierten Formrepräsentation stellen Fourierdeskriptoren [Zah72] dar. Sie resultieren aus der Anwendung der Fouriertransformation auf einer Regionenkontur, wobei die Kontur gewöhnlich durch eine Signatur repräsentiert wird. Für eine gute Formbeschreibung ist die Wahl dieser eindimensionalen Repräsentation besonders entscheidend. Zhang und Lu [Zha01a] untersuchen in ihrer Arbeit vier verschiedene Kontursignaturen. Ihre Ergebnisse demonstrieren, dass die Signatur, die den Abstand der Konturelemente zum Zentroiden berücksichtigt (Zentroiddistanzfunktion), die anderen Varianten (Komplexe Koordinaten, Krümmungssignatur und kumulative Winkelfunktion) übertrifft. Formal ist eine Regionenkontur als ein Satz von Koordinaten (x_i, y_i), $i = 1, 2, \ldots, I$, definiert. Dementsprechend sind die Elemente r_i der Signatur unter Berücksichtigung des Abstandes zum Zentroiden (x_c, y_c) durch

$$r_i = ([x_i - x_c]^2 + [y_i - y_c]^2)^{1/2},\ i = 1, 2, \ldots, I$$

gegeben. Diese auf das Regionenzentrum normierte Kontursignatur bildet schließlich die Grundlage für die Berechnung der Fourierkoeffizienten.

Andere konturbasierte Methoden [Mok96, Bob01] sind auf der Grundlage der *Curvature Scale Space* (CSS) Darstellung definiert. Dabei handelt es sich um eine Multiskalenorganisation von Wendepunkten, in der die Krümmungseigenschaften der zu

beschreibenden Kontur kodiert sind [Mok92]. Die Hauptidee dieses Verfahrens ist die Kombination von schrittweiser Glättung der in Bogenlänge u parametrisierten Kontur und die Analyse ihres Krümmungsverhaltens. Als Glättungsfunktion wird eine eindimensionale Gaußfunktion verwendet, deren Standardabweichung σ ein Maß für den Grad der Glättung darstellt und dementsprechend eine Skalierungsstufe repräsentiert. Für jede Skalierung können die Wendepunkte der Kontur bestimmt werden. Diese trennen konkave und konvexe Konturabschnitte. Die in Bogenlänge repräsentierten Wendepunkte der Kurve werden für jedes σ in eine (σ, u)-Ebene eingetragen. Dabei wird der Glättungsparameter solange erhöht, bis keine Wendepunkte mehr detektiert werden können. Dies ist gleichbedeutend mit einem schrittweisen Ausglätten der konkaven Strukturen bis die Kontur nur noch konvex ist. Die resultierende CSS-Darstellung wird als CSS-Bild bezeichnet. Jeder Peak des CSS-Bildes entspricht einem konkaven bzw. konvexen Abschnitt der Originalkontur. Die Regionenkontur wird schließlich durch die Maxima der ausgeprägtesten Peaks repräsentiert [Mok96].

Der größte Nachteil der bisher beschriebenen Verfahren ist, dass ihre Anwendung die Detektion von Bildregionen oder Objekten erfordert. Die Güte der Bildbeschreibung ist somit eng an die Qualität der Bildsegmentierung bzw. Objektdetektion geknüpft. Diese Ansätze sind daher gewöhnlich auf Anwendungen beschränkt, in denen eine einfache automatische Segmentierung der zu speichernden Bilddaten garantiert ist oder eine geeignete Bildsegmentierung manuell erzielt wurde. Beispiele für einfach zu segmentierende Bildsammlungen sind sowohl die Bilder der Meereslebewesen des SQUID Systems[10] als auch die Objektaufnahmen der *Columbia Object Image Library* [Nen96]. Darüber hinaus existieren jedoch auch Algorithmen zur Beschreibung inhärenter Formcharakteristika, die keine derartige Vorverarbeitung erfordern. Brandt et al. [Bra99, Bra00] stellen in ihren Arbeiten einige dieser Ansätze vor. Ihre Grundlage bildet die Extraktion von verschiedenen Bildkanten. Darauf aufbauend können sowohl Kantenhistogramme als auch ihre Erweiterung, Kantenübergangsmatrizen, berechnet werden. Ausgehend von der binären Kantenrepräsentation lassen sich außerdem verschiedene Fourierdeskriptoren berechnen. Die Ergebnisse dieser Arbeiten demonstrieren, dass die besten Suchergebnisse mit dem einfachen Fourierdeskriptor erzielt werden können. Die experimentellen Untersuchungen belegen zusätzlich, dass sich auch mit dem einfach zu berechnenden Kantenhistogramm sowie der Kantenübergangsmatrix gute Suchergebnisse erzielen lassen.

2.2.4 Fokuspunkte

Oftmals sind globale Bildbeschreibungen nicht ausreichend, um unterschiedliche Bilder voneinander zu separieren oder ähnliche zu gruppieren, sodass es sinnvoll er-

[10] Das SQUID (Shape Queries Using Image Databases) System wurde an der Universität von Surrey entwickelt. Seine Haupteigenschaft ist die inhaltsbasierte Bildersuche auf der Grundlage von CSS-Formdeskriptoren. Eine Demoversion des Systems ist unter folgender Adresse verfügbar (Stand 18.01.2005): `http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`

scheint, lokale Bildcharakteristika zu extrahieren und zur Bildklassifikation zu verwenden. Diese Deskriptoren werden nur in den Bildregionen berechnet, die spezielle visuelle Eigenschaften besitzen und für den Inhalt eines Bildes daher besonders charakteristisch sind. Im Gegensatz zu den in Abschnitt 2.2.1 dargestellten globalen Bildbeschreibungen, in denen lokale Bildeigenschaften kodiert sind, werden die lokalen Deskriptoren nicht zu einem Gesamtrepräsentanten kombiniert, sondern separat zur inhaltsbasierten Bildersuche verwendet. Wie aber können interessante Bildbereiche gefunden werden und welche Eigenschaften besitzen sie? Eine Idee zur Lösung dieser Fragestellungen stellt die Detektion von sogenannten interessanten Punkten (engl. *Points of Interest*) dar, die im weiteren Verlauf dieser Arbeit auch als Fokuspunkte bezeichnet werden. Sie entsprechen Stellen im Bild, die häufig spezielle geometrische (Kanten, Ecken) oder nicht geometrische (Kontrast) Eigenschaften besitzen. Ihre Umgebung bildet die Grundlage zur Extraktion lokaler Bildinformationen, wie z.B. Farbe oder Textur. Ein Überblick über existierende Verfahren und ein Vergleich der unterschiedlichen Punktdetektoren kann in den Arbeiten von Schmid et al. [Sch00], Wolf [Wol00] sowie Sebe et al. [Seb02] gefunden werden. Im Folgenden werden repräsentativ einige punktbasierte Verfahren zur Bildersuche vorgestellt.

Schmid und Mohr [Sch97] verwenden zur inhaltsbasierten Bildersuche lokale Merkmale, die für jeden Fokuspunkt eines Bildes berechnet werden. Die notwendigen interessanten Bildpunkte resultieren aus dem Einsatz des Harris-Punktdetektors [Har88]. Die Grundidee dieses Verfahrens bildet die Verwendung der Autokorrelationsfunktion, um Stellen im Bild zu finden, an denen sich das Signal in zwei Richtungen ändert. Darauf aufbauend wird eine Matrix bestimmt, deren Einträge auf der Berechnung der ersten Ableitungen innerhalb eines Fensters basieren. Die Eigenvektoren der Matrix entsprechen den Hauptkrümmungen der Autokorrelationsfunktion. Große Eigenwerte deuten auf die Präsenz eines interessanten Bildpunktes hin. Der für die Umgebung eines Fokuspunktes berechnete *Local Jet* beinhaltet lokale Ableitungen unterschiedlicher Ordnung und bildet die Grundlage zur Berechnung der verschiedenen Invarianten, die zu einem lokalen Deskriptor zusammengefasst werden. Die Bestimmung der Ähnlichkeit zwischen einem gespeicherten Bild und dem Beispielbild erfordert den Vergleich der bildspezifischen lokalen Repräsentanten. Ein Wahl-Algorithmus (engl. *Voting Algorithm*) ermöglicht dabei die Kombination der Ergebnisse der verschiedenen Einzelvergleiche zu einem Ähnlichkeitswert. Liegt der Abstand zweier Repräsentanten unter einem bestimmten Schwellwert, dann erhält das gespeicherte Bild eine Stimme (engl. *Vote*). Die Bilder mit den meisten Stimmen bilden die Ergebnismenge und werden dem Benutzer präsentiert.

Die Arbeit von Gouet und Boujemaa [Gou01] basiert auf der Verbesserung des Harris-Punktdetektors, der in der Literatur auch als *Precise Harris Keypoint Detector* bezeichnet wird [Sch00, Seb02]. Das ursprünglich auf den Grauwerten eines Bildes basierende Verfahren wird auf die einzelnen Kanäle des RGB Farbraumes erweitert. Für jeden gefundenen Fokuspunkt wird ein lokaler Bilddeskriptor berechnet. Dieser beinhaltet zusätzlich zu den drei Intensitäten und den drei Gradientenstärken der Farbkanäle,

zwei farbspezifische Invarianten [Mon98, Gou01]. Gegenüber anderen Verfahren hat dieser Ansatz den Vorteil, dass bei nahezu gleichen Kosten zur Speicherung der lokalen Merkmalsvektoren keine Beschränkung auf Grauwerte stattfindet, sondern die lokalen Farbinformationen eines Bildes erfasst werden.

Die meisten Punktdetektoren konzentrieren sich auf die Detektion von Ecken und Kanten, da sich dort das Bildsignal am stärksten ändert und daher gefolgert werden kann, dass in diesen Bereichen die wichtigsten Bildinformationen lokalisiert sind. Allerdings muss der visuelle Fokus nicht notwendigerweise in Bereichen liegen, in denen scharfe Kanten und Ecken konzentriert sind. Da außerdem zur effizienten Indizierung nur eine begrenzte Anzahl von Bildpunkten verwendet wird, hat diese Beschränkung gerade bei stark texturierten Bildern einen entscheidenden Nachteil. Die interessanten Bildpunkte, die mit Verfahren detektiert werden, die auf Kanten und Ecken fokussieren, sind fast ausschließlich in texturierten Bereichen lokalisiert. Anderen Bildbereiche werden daher nicht oder nur unzureichend repräsentiert. Weiche Bildkanten zum Beispiel werden mit diesen Verfahren nicht gefunden, obwohl sie ebenfalls visuell interessante Bildbereiche repräsentieren können.

Loupias et al. [Lou00, Seb02] stellen ein Verfahren vor, das neben scharfen Kanten und Ecken auch weiche Kanten findet. Die aus der Anwendung des Algorithmus resultierenden Fokuspunkte bilden in texturierten Bereichen keine Häufungsgebiete, sodass die lokalen Bildinformationen der verschiedenen Bildbereiche repräsentiert werden können. Wichtigster Bestandteil dieses Ansatzes ist die Waveletzerlegung [Mal89]. Dabei wird ein Bild in mehreren aufeinander folgenden Schritten in Detail- und Skalierungsinformationen zerlegt (vgl. Abbildung 2.6 auf S. 19). Durch Rückverfolgung der Waveletkoeffizienten von der untersten bis zur obersten Skalierungsstufe lassen sich schließlich die interessanten Punkte eines Bildes detektieren. Durch die Verarbeitung unterschiedlicher Bildskalierungen werden auch Bildkanten erfasst, die in der ursprünglichen Auflösung nicht scharf, sondern weich sind. Aus einer zugehörigen Evaluation geht hervor, dass dieses Verfahren unter Berücksichtigung der Kriterien Wiederholbarkeit (engl. *Repeatability*) und Informationsgehalt (engl. *Information Content*) bessere Ergebnisse erzielt als der Harris-Punktdetektor [Sch00, Seb02]. Zusätzlich wird demonstriert, dass mit den in der Umgebung der Fokuspunkte extrahierten Farb- und Textureinformationen signifikant bessere Sucherergebnisse in einer Objektdatenbank erzielt werden können als durch global berechnete Bildrepräsentanten.

2.2.5 Bildsegmentierung

Neben der Detektion von Fokuspunkten stellt die Bildsegmentierung eine weitere Vorverarbeitung zur Extraktion lokaler Bildcharakteristika dar. Zielsetzung dieses Prozesses ist die Unterteilung eines Bildes in unverbundene, homogene Regionen, deren

Bildpunkte im Sinne eines bestimmten Charakteristikums, wie z.B. Farbe oder Textur, ähnlich sind. Alternativ dazu kann die Bildpartitionierung auch durch das Finden der Regionengrenzen erreicht werden. Im Idealfall lassen sich die resultierenden Regionen in einem weiteren Verarbeitungsschritt zu Objekten zusammenfassen.[11] Für die Berechnung bestimmter Form- und Layoutcharakteristika ist die Bildsegmentierung eine notwendige Voraussetzung (vgl. z.B. [Lon98] oder [Rub01]). Dabei ist jedoch zu beachten, dass für die Extraktion von Layoutmerkmalen schon eine grobe Segmentierung ausreichend ist, während die Berechnung der Formcharakteristika eine präzisere Regionendarstellung erfordert.

Die in der Literatur beschriebenen Segmentierungsverfahren sind zahlreich. Eingesetzt werden beispielsweise Gruppierung im Merkmalsraum, Schwellwertverfahren, *Split and Merge*-Techniken, Regionenwachstum, kantenbasierte Algorithmen oder Neuronale Netze [Luc01]. Einige der Ansätze, die in inhaltsbasierten Bildsuchsystemen angewandt werden, sind in den folgenden Abschnitten näher erläutert.

Das von Boujemaa und Fauqueur [Bou00, Fau02] vorgestellte Segmentierverfahren basiert auf dem *Competitive Agglomeration* (CA) Algorithmus, einem Fuzzy *k*-means Clusterverfahren, das den Vorteil besitzt, die optimale Clusteranzahl automatisch zu bestimmen. In dem mehrstufigen Segmentierungsprozess werden die Pixel eines Bildes zuerst in den CIE $L^*u^*v^*$ Farbraum abgebildet. Aus der Farbraumquantisierung durch CA-Clustering resultiert eine Menge von Farbprototypen, deren Rückprojektion zu einer quantisierten Darstellung des Ausgangsbildes führt. Im nächsten Schritt werden abhängig von der Farbraumquantisierung lokale Farbverteilungen (engl. *Local Distributions of Quantized Colors*, LDQC) berechnet und mittels CA-Clustering klassifiziert. Anschließend wird jedem Bildpunkt der korrespondierende LDQC-Prototyp zugeordnet. In einem Nachbearbeitungsschritt werden in dem segmentierten Bild zu kleine Regionen auf der Grundlage eines Regionengraphen mit benachbarten Gebieten verschmolzen. Die Repräsentation einer Bildregion durch eine verfeinerte Farbdarstellung, die sogenannte *Adaptive Distribution of Color Shades*, bildet die Grundlage für den Regionenvergleich im Rahmen des inhaltsbasierten Suchprozesses.

In dem von Carson et al. [Car99, Car02] entwickelten Bildsuchsystem Blobworld wird ein Pixel durch einen Merkmalsvektor bestehend aus Farbe (dreidimensionale Darstellung im CIE $L^*a^*b^*$ Farbraum), Textur (Polarität, Anisotropie und Kontrast) und Bildkoordinaten repräsentiert. Unter der Annahme, dass sich die verschiedenen Bildsegmente durch Gaußverteilungen in dem achtdimensionalen Merkmalsraum modellieren

[11] Die Objekterkennung ist ohne Einschränkung auf eine spezielle Bilddomäne bzw. ein spezielles Anwendungsszenario eine sehr schwer zu bewältigende Aufgabe, für die noch keine generelle Lösung existiert. Deutlich wird diese Problematik an dem von Flickner et al. [Fli95, Abschnitt „Semantic versus nonsemantic information“] formulierten Beispiel der Hunderkennung in einem Kinderbuch. Die Teilnehmer einer KI Konferenz wurden aufgefordert eine Software zu entwickeln, die alle in einem Kinderbuch auftretenden Hunde erkennen sollte. Obwohl schon ein dreijähriges Kind diese Aufgabe zu lösen vermag, fühlte sich kein Konferenzteilnehmer in der Lage, dieses Objekterkennungsproblem zufriedenstellend lösen zu können.

lassen, werden die Verteilungsparameter durch Anwendung des *Expectation Maximation*[12] (EM) Algorithmus geschätzt. Die resultierenden Pixel-Cluster Zugehörigkeitswerte ermöglichen schließlich die Bildsegmentierung.

Die Grundidee des von Ma und Manjunath [Ma97a, Ma97b] vorgestellten Segmentierungsverfahrens bildet die Konstruktion eines Kantenflussmodells, welches das Auffinden von Regionengrenzen ermöglicht. Dabei wird für jedes Pixel ein Flussvektor bestimmt, der auf der Analyse lokaler Farb-, Textur- und Phaseninformation basiert. Neben der Kantenenergie beinhaltet der Vektor die Wahrscheinlichkeit dafür, dass eine Regionengrenze in einer bestimmten Richtung bzw. in der entgegengesetzten Richtung liegt. Das Auffinden der verschiedenen Regionenkanten erfolgt durch die iterative Ausbreitung der lokalen, pixelgebundenen Kantenflüsse auf benachbarte Pixel. Der Vorgang stoppt, sobald ein Kantenfluss auf einen entgegengesetzten Fluss trifft. Da nun die Kantenflussvektoren zweier Bildpunkte entgegengesetzt orientiert sind, deutet dies auf die Existenz einer Kante hin. Nachdem der Fortpflanzungsalgorithmus einen stabilen Zustand erreicht hat, können die Grenzenergien berechnet werden. In einer Nachbearbeitung werden kleinere Grenzabschnitte miteinander verbunden und Regionen konstruiert. Zusätzlich werden die Farb- und Texturinformationen der Regionen dazu verwendet, um diese gegebenfalls miteinander zu verschmelzen. Ein Vorteil des Segmentierungsverfahrens ist die relativ einfache Parametrisierung. Dabei muss lediglich die bevorzugte Skalierung angegeben werden, für die die Regionengrenzen detektiert werden sollen. Für den abschließenden Verschmelzungsvorgang wird lediglich eine näherungsweise Angabe für die Anzahl der Regionen benötigt.

2.3 Ähnlichkeit von Bildcharakteristika

Wie bereits erwähnt, erfordert jede Suche in einem Bilddatenbanksystem die Formulierung einer Suchanfrage (vgl. Abschnitt 2.1). Um relevante Bilder in der Datenbank zu finden, wird die Anfrage auf eine formale Darstellung abgebildet. Beispielbilder sowie vom Benutzer spezifizierte Farben, Texturen und Formen werden durch mathematische Deskriptoren, wie z.B. Merkmalsvektoren und Signaturen, beschrieben. Die Generierung einer Ergebnisliste erfordert den Ähnlichkeitsvergleich dieser Anfragecharakteristika mit den Repräsentanten der in der Datenbank gespeicherten Bilder. Bilder, deren Charakteristika zu denen der Anfrage ähnlich sind, bilden schließlich die Ergebnismenge. Abbildung 2.7 veranschaulicht diesen Prozess am Beispiel einer Query-By-Example Suche.

Mathematisch wird die Ähnlichkeit zweier Bilder durch Abstandsberechnung bestimmt. Je ähnlicher die Deskriptoren zweier Bilder sind, desto geringer sollte ihr

[12] Bei dem Expectation Maximation Algorithmus handelt es sich um ein Verfahren zur iterativen Optimierung von statistischen Modellen.

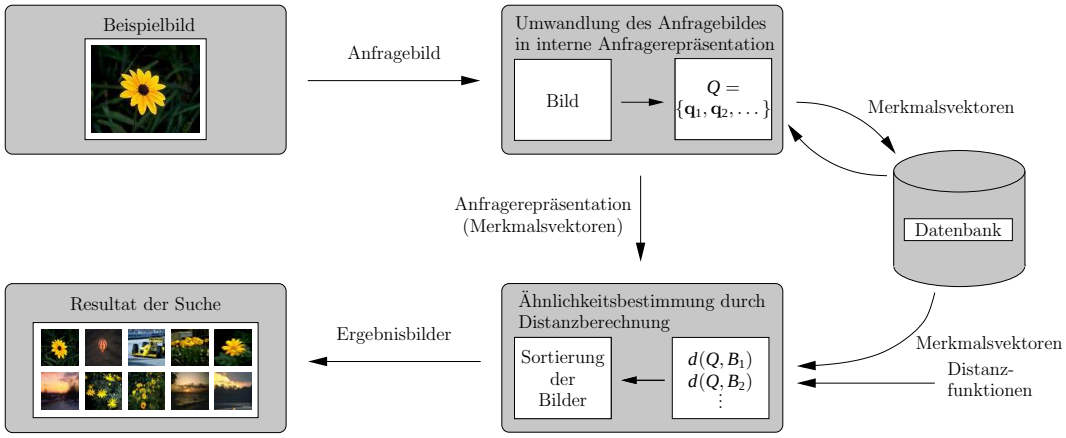


Abb. 2.7: Inhaltsbasierte Suche durch Query-By-Example. Das initial ausgewählte Beispielbild wird auf eine formale Bildbeschreibung, die Merkmalsvektoren, abgebildet. Diese Charakteristika werden mit denen der gespeicherten Bilder verglichen. Daraus resultiert für jedes gespeicherte Bild ein Abstandswert, der ein Maß für die Ähnlichkeit zum Beispielbild darstellt. Eine Sortierung der Bilder auf der Basis ihrer Abstandswerte ermöglicht die Generierung der Ergebnisliste.

Abstand in den korrespondierenden Merkmalsräumen sein.[13] Distanzfunktionen stellen daher neben extrahierten Bildcharakteristika die wesentlichen Komponenten eines inhaltsbasierten Bildsuchsystems dar. Demnach werden Abstandsmaße wie die in Abbildung 2.8 definierten Metriken benötigt, die den Bildvergleich auf der Grundlage von Merkmalsrepräsentanten ermöglichen. In den nächsten Abschnitten werden daher verschiedene Verfahren zur Abstandsbestimmung vorgestellt.

2.3.1 Minkowski-Metriken

Die Minkowski-Metriken zweier Merkmalsvektoren $\mathbf{x}$ und $\mathbf{y}$ sind durch

$$d_{L_p}(\mathbf{x}, \mathbf{y}) = \left(\sum_i^N |x_i - y_i|^p \right)^{\frac{1}{p}}, \quad \text{mit } \mathbf{x}, \mathbf{y} \in \mathbb{R}^N,$$

definiert. Die L_1-Distanz ($p = 1$) wird als Manhattan oder auch City-Block Abstand bezeichnet. Der euklidische Abstand ergibt sich für $p = 2$ (L_2-Distanz) und der Supremumsabstand bildet die Grenze für $p \to \infty$ (L_∞-Distanz). Stricker und Orengo [Str95] verwenden die vorgestellten Metriken zur Berechnung des Abstandes zweier kumulativer Histogramme. In der Arbeit von Gouet und Boujemaa [Gou01] bildet der euklidische Abstand die Grundlage für den Vergleich von lokal berechneten Farbinvarianten. Da die Minkowski-Metriken ausschließlich gleiche Vektorkomponenten bei

[13] Dass dies nicht immer so ist, folgt aus der Diskrepanz zwischen der perzeptiven Wahrnehmung des Menschen und der mathematischen Modellierung innerhalb eines Bildsuchsystems. Die Problematik der verschiedenen Bildbetrachtungsweisen von Mensch und System wird in Abschnitt 2.4 näher erläutert.

Metrische Räume: In einem metrischen Raum hat man einen Abstandsbegriff zur Verfügung. Eine nicht leere Menge X heißt genau dann ein metrischer Raum, wenn jedem geordneten Paar $(\mathbf{x}, \mathbf{y})$ von Punkten $\mathbf{x}$ und $\mathbf{y}$ aus X stets eine reelle Zahl $d(\mathbf{x}, \mathbf{y}) \geq 0$ zugeordnet werden kann, sodass für alle $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ gilt:

(i) $d(\mathbf{x}, \mathbf{y}) = 0$ genau dann, wenn $\mathbf{x} = \mathbf{y}$ (Definitheit)

(ii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ (Symmetrie)

(iii) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (Dreiecksungleichung)

Die Zahl $d(\mathbf{x}, \mathbf{y})$ heißt der *Abstand* zwischen den Punkten $\mathbf{x}$ und $\mathbf{y}$.

Abb. 2.8: Metrische Räume und Abstandsbegriff nach Zeidler [Zei96].

der Distanzberechnung berücksichtigen ($|x_i - y_j|^p$, mit $i = j$), entspricht ein auf deren Grundlage berechneter Abstandswert nicht unbedingt dem perzeptiven Abstandsempfinden eines Menschen. Abbildung 2.9 veranschaulicht diese Problematik am Beispiel der L_1-Distanz zweier Grauwerthistogramme.

2.3.2 Histogrammschnitt

Der Abstand zweier Histogramme $H = \{h_1, h_2, \ldots, h_N\}$ und $K = \{k_1, k_2, \ldots, k_N\}$ lässt sich durch die Berechnung des Histogrammschnitts (engl. *Histogram Intersection*) bestimmen [Swa91]:

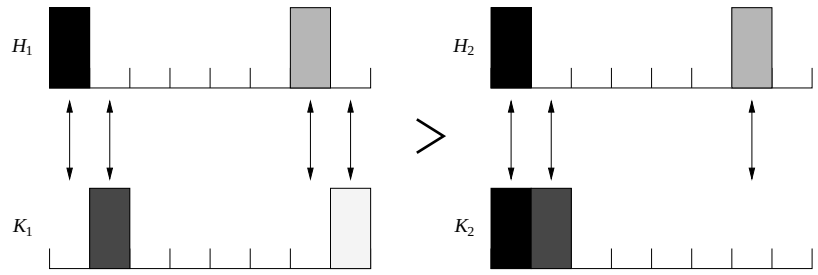
$$d_\cap(H, K) = 1 - \frac{\sum_{i=1}^{N} \min(h_i, k_i)}{\sum_{i=1}^{N} k_i}$$

Wenn beide Histogramme H und K die Bedingung $\sum_{i=1}^{N} h_i = \sum_{i=1}^{N} k_i$ erfüllen, sind der Histogrammschnitt und die normierte L_1-Distanz (vgl. Abschnitt 2.3.1) äquivalent, $d_\cap(H, K) = \frac{1}{2N}\, d_{L_1}(H, K)$ (Beweis siehe [Swa91]).

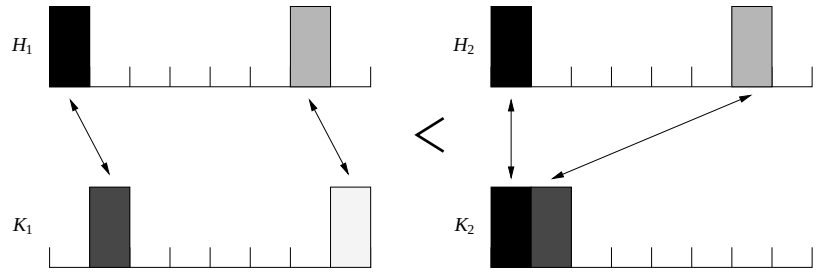
2.3.3 Generalisierter euklidischer Abstand

Die bisher vorgestellten Abstandsmaße basieren ausschließlich auf dem Vergleich von Vektorkomponenten x_i und y_j, die denselben Index besitzen, d.h. $i = j, \forall i, j$. Durch Generalisierung des euklidischen Abstandes können auch Korrelationen zwischen unterschiedlichen Komponenten berücksichtigt werden:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T\, \mathbf{W}\, (\mathbf{x} - \mathbf{y})}$$



(a) Abstandsbestimmung durch L_1-Distanz



(b) Abstandsbestimmung durch ein perzeptives Distanzmaß

Abb. 2.9: Problematik der klassischen Minkowski-Metriken am Beispiel der L_1-Distanz zweier Grauwerthistogramme (nach Rubner und Tomasi [Rub01]). (a) Angenommen die Einträge jedes Histogramms summieren sich zu Eins, dann ergibt sich für ihre Abstände $d_{L_1}(H_1, K_1) = 2$ und $d_{L_1}(H_2, K_2) = 1$. (b) Im Gegensatz zu den klassischen Minkowski-Metriken berücksichtigt ein perzeptives Abstandsmaß die Ähnlichkeit der unterschiedlichen Grauwerte, sodass $d_{perz}(H_1, K_1) < d_{perz}(H_2, K_2)$.

Die Matrix $\mathbf{W} = [w_{ij}]$, mit $i, j \in \{1, 2, \ldots, N\}$, beinhaltet die Beziehungen zwischen den Vektoreinträgen i und j. Der klassische euklidische Abstand resultiert schließlich aus $\mathbf{W} = \mathbf{E} = [\delta_{ij}]_{N,N}$, mit

$$\delta_{ij} = \begin{cases} 1, & \text{falls } i = j \\ 0, & \text{sonst} \end{cases}$$

Faloutsos et al. [Fal94] verwenden eine spezielle Variante des generalisierten Abstandsmaßes zur farbbasierten Bildersuche. Demnach wird der Abstand zweier Histogramme H und K nach folgender Gleichung berechnet:

$$d^2_{\text{hist}}(H, K) = d^2_{\text{hist}}(\mathbf{h}, \mathbf{k}) = (\mathbf{h} - \mathbf{k})^T \, \mathbf{A} \, (\mathbf{h} - \mathbf{k}) = \sum_i^N \sum_j^N a_{ij}(h_i - k_i)(h_j - k_j)$$

$\mathbf{h}$ und $\mathbf{k}$ repräsentieren jeweils die vektorielle Schreibweise der Histogramme H und K. Die Einträge a_{ij} der Matrix $\mathbf{A} = [a_{ij}]$ beschreiben die Ähnlichkeit zweier Farben i und j. Sie können nach folgender Berechnungsvorschrift bestimmt werden [Nib93]:

$$a_{ij} = 1 - \frac{d_{ij}}{d_{\max}}, \quad \text{mit } d_{ij} = \|i - j\| \text{ und } d_{\max} = \max_{ij} d_{ij}$$

Mahalanobis Abstand

Der Mahalanobis Abstand ist ein spezieller generalisierter euklidischer Abstand. Er basiert auf einer Datenmenge $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ mit Mittelwertsvektor $\boldsymbol{\mu} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i$ und wird wie folgt berechnet:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \, \mathbf{C}^{-1} \, (\mathbf{x}_i - \mathbf{x}_j)}$$

Dabei repräsentiert $\mathbf{C}$ die Kovarianzmatrix der Datenmenge, die wie folgt definiert ist:

$$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

2.3.4 Abstandsmaße aus der Informationstheorie

Divergenzen aus der Informationstheorie messen, wie kompakt eine Verteilung H kodiert werden kann, wenn eine andere Verteilung K als Codebuch verwendet wird. Durch die Anzahl der Bild- oder Regionenpunkte normierte Histogramme, die somit Wahrscheinlichkeitsverteilungen darstellen, lassen sich daher durch Berechnung der **Kullback-Leibler (KL) Divergenz** miteinander vergleichen [Puz99, Rub01]:

$$d_{\mathrm{KL}}(H, K) = \sum_{i}^{N} h_i \, \log \frac{h_i}{k_i}$$

Ein weiteres Abstandsmaß aus der Informationstheorie ist die **Jeffrey-Divergenz**. Im Gegensatz zur KL-Divergenz ist dieses Maß symmetrisch, nummerisch stabiler und robuster gegen Rauschen [Jol01, Rub01]:

$$d_{\mathrm{J}}(H, K) = \sum_{i}^{N} \left(h_i \, \log \frac{h_i}{m_i} + k_i \, \log \frac{k_i}{m_i} \right), \quad \text{mit} \quad m_i = \frac{h_i + k_i}{2}$$

2.3.5 Earth-Mover's-Distanz

Rubner und Tomasi stellen mit der Earth-Mover's-Distanz (EMD) ein Abstandsmaß vor, mit dem unterschiedliche Signaturen miteinander verglichen werden können [Rub01]. Der Abstand zweier Signaturen $P = \{(\mathbf{p}_1, w_{\mathbf{p}_1}), \ldots, (\mathbf{p}_m, w_{\mathbf{p}_m})\}$ und $Q = \{(\mathbf{q}_1, w_{\mathbf{q}_1}), \ldots, (\mathbf{q}_n, w_{\mathbf{q}_n})\}$ basiert demnach auf der Lösung eines Transportproblems und wird durch einen Erdbewegungsvorgang modelliert. Dabei werden die Einträge der einen Signatur als Erdhaufen betrachtet und die der anderen als Erdlöcher. Die Arbeit, die notwendig ist, die Löcher der einen Signatur mit der Erde der anderen zu füllen, ist ein Maß für ihren Abstand. Dabei soll so wenig Erde wie möglich bewegt werden, sodass sich folgendes Optimierungsproblem formulieren lässt:

Sei f_{ij} der Fluss zwischen den Signaturelementen $\mathbf{p}_i$ und $\mathbf{q}_j$, dann wird der Fluss $\mathbf{F} = [f_{ij}]$ gesucht, der die Gesamtkosten

$$\mathrm{WORK}(P, Q, \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d(\mathbf{p}_i, \mathbf{q}_j)\, f_{ij},$$

minimiert. Die Lösung dieses Problems erfordert sowohl die Berechnung der Grunddistanz d zweier Vektoren $\mathbf{p}_i$ und $\mathbf{q}_j$ als auch die Einhaltung der folgenden Nebenbedingungen:

$$f_{ij} \geq 0, \quad 1 \leq i \leq m,\ 1 \leq j \leq n \tag{2.1}$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{\mathbf{p}_i}, \quad 1 \leq i \leq m \tag{2.2}$$

$$\sum_{i=1}^{m} f_{ij} \leq w_{\mathbf{q}_j}, \quad 1 \leq j \leq n \tag{2.3}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{\mathbf{p}_i}, \sum_{j=1}^{n} w_{\mathbf{q}_j} \right) \tag{2.4}$$

Bedingung (2.1) gewährleistet, dass die Erde nur von P nach Q bewegt wird und nicht umgekehrt. Dass nur so viel Erde bewegt bzw. aufgenommen wird wie möglich ist, wird durch die Bedingungen (2.2) und (2.3) garantiert. Bedingung (2.4) begrenzt schließlich das Maximum der zu verschiebenden Erdmenge, den sogenannten *Total Flow*. Nach Lösung des Optimierungsproblems berechnet sich die EMD zweier Signaturen P und Q wie folgt:

$$\mathrm{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d(\mathbf{p}_i, \mathbf{q}_j)\, f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

Die wichtigste Eigenschaft der EMD ist, dass sie auch den Vergleich von Signaturen erlaubt, die aus unterschiedlich vielen Komponenten bestehen. Außerdem erfüllt dieses Abstandsmaß die Eigenschaften einer Metrik (vgl. Abbildung 2.8), wenn ihre Grunddistanz eine Metrik darstellt und die Gesamtsummen der Gewichte identisch sind, d.h. $\sum_{i=1}^{m} w_{\mathbf{p}_i} = \sum_{j=1}^{n} w_{\mathbf{q}_j}$ (vgl. [Rub01, S. 17f]).

2.4 Adaptive Bildersuche durch Mensch-Maschine Interaktion

In den vorherigen Abschnitten wurden mit den verschiedenen low-level Bildcharakteristika und den Abstandsmaßen die wichtigsten Bestandteile eines inhaltsbasierten Bildsuchsystems vorgestellt. Ihre Entwicklung und Kombination wird von der Zielsetzung motiviert sowohl die visuelle Wahrnehmung eines Menschen als auch sein perzeptives Ähnlichkeitsempfinden so gut zu modellieren, dass möglichst eine Separierung der verschiedenen semantischen Bildklassen erzielt werden kann. Was in der Theorie so einfach klingt, ist in der Praxis eine kaum zu bewältigende Aufgabe. Die semantische Gruppierung auf der Basis eines fixen Satzes von low-level Merkmalen und Abstandsmaßen ist - wenn überhaupt - nur in einem sehr eingeschränkten Rahmen lösbar. „Ein Bild sagt mehr als tausend Worte“ und woher soll das System wissen, was diese Worte sind?

Die Bildinterpretation sowie die perzeptive Ähnlichkeit zweier Bilder sind Prozesse, die sowohl anwendungsspezifisch als auch benutzerabhängig sind. Ein Bildsuchsystem sollte daher in der Lage sein, sich dynamisch an einen Anwender und seine Suchintention zu adaptieren. Automatisch extrahierte low-level Merkmale wie Farbe, Textur und Form können die *high-level* Konzepte eines Anwenders innerhalb einer Bildersuche oft nur unzureichend repräsentieren. Da aber anzunehmen ist, dass die low-level Charakteristika in irgendeiner Art und Weise mit den semantischen Konzepten eines Anwenders korrelieren [Hua02, Zho03a], ist es notwendig und sinnvoll, den Benutzer in den Suchprozess einzubeziehen.

2.4.1 Lernen durch Relevance Feedback

Am einfachsten kann ein Benutzer[14] in den Suchprozess einbezogen werden, wenn ihm die Möglichkeit gegeben wird, die Gewichte der verwendeten Merkmale entsprechend seiner Suchintention einzustellen (vgl. z.B. [Bac96]). Obwohl gewisse Systemparameter in diesem Fall individuell angepasst werden können, erfordern sie vom Anwender ein Mindestmaß an technischem Wissen und Verständnis über deren Funktionsweise. Um die Anwendung eines inhaltsbasierten Bildsuchsystems einer weniger fachkundigen Anwendergruppe zugänglich zu machen, muss jedoch eine natürliche und einfache Interaktionsmöglichkeit mit dem System zur Verfügung gestellt werden.

Moderne Bilddatenbanksysteme [Rui97, Ish98, Cox00, Laa00] verzichten auf die manuelle Einstellung interner Systemparameter und fordern den Benutzer lediglich auf,

[14] Die Einbeziehung eines Benutzers in einen bestimmten Prozess wird in der Literatur durch den Ausdruck „*Human-in-the-Loop*“ beschrieben.

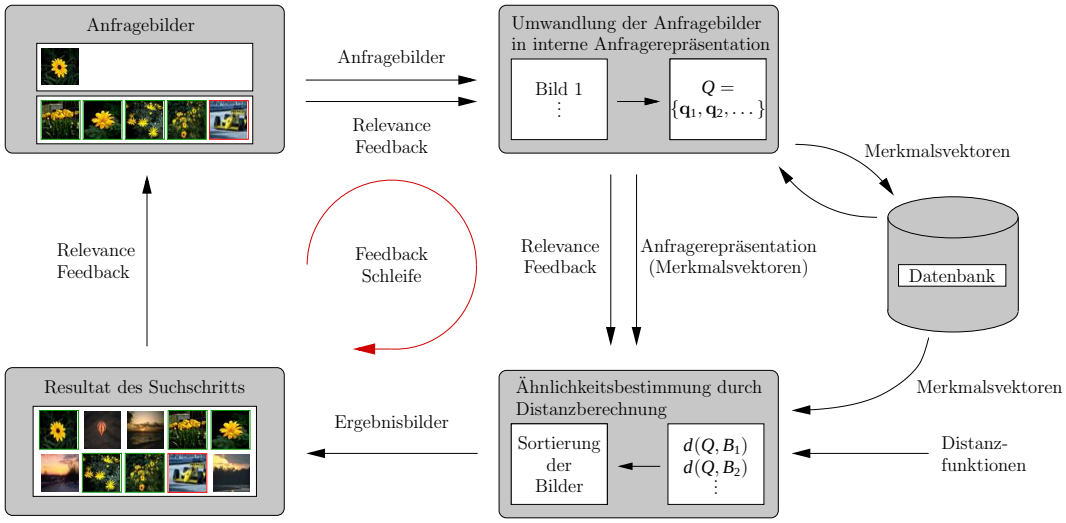


Abb. 2.10: Adaptive Bildersuche durch Mensch-Maschine Interaktion (basierend auf [Mül01]). Das initiale Suchergebnis (unten links) wird vom Benutzer bewertet. Die internen Systemparameter, wie z.B. die formale Anfragebeschreibung und die Parametrisierung der Abstandsfunktionen, werden auf der Basis des Relevance Feedback adaptiert und eine neue Ergebnisliste wird ermittelt. Der iterative Suchprozess läuft solange bis der Benutzer die Bilder der gesuchten Kategorie oder das Zielbild gefunden hat. Alternativ kann der Anwender die Suche auch abbrechen.

die Ergebnisse eines Suchschritts entsprechend ihrer Relevanz bezüglich der Suchintention zu bewerten. Diese benutzerfreundliche Technik wird als *Relevance Feedback* bezeichnet, und obwohl sie ursprünglich aus dem *Information Retrieval* [Sal83, Sal89] stammt, scheint sie für die inhaltsbasierte Bildersuche besser geeignet zu sein. Schließlich ist es einfacher, ein Bild zu bewerten, das durch einen kurzen Blick erfasst werden kann, als ein schriftliches Dokument zu beurteilen, das dazu erst gelesen werden muss.

Das typische Szenario der adaptiven Bildersuche durch Mensch-Maschine Interaktion ist in Abbildung 2.10 dargestellt und lässt sich folgendermaßen skizzieren:

1. Das System präsentiert auf der Basis eines klassischen Suchansatzes, wie z.B. Query-By-Example, ein initiales Suchergebnis.

2. Der Benutzer bewertet, ob und wie relevant die verschiedenen Bilder der Ergebnismenge bezüglich seiner Suchintention sind.

3. Das System lernt und präsentiert eine neue Ergebnisliste.

4. Falls der Benutzer mit dem Suchergebnis zufrieden ist oder die Suche abbricht, wird der Suchprozess beendet. Ansonsten wird mit Schritt 2 fortgefahren.

Ausgehend von der mathematischen, vektoriellen Repräsentation eines Bildes oder einer Bildregion[15] existieren verschiedene Varianten innerhalb eines adaptiven Bildsuchsystems zu lernen (vgl. auch [Zho03a]). Werden zum Lernen ausschließlich als relevant bewertete Bilder betrachtet, kann das Systemlernen durch die Schätzung einer Wahrscheinlichkeitsdichte [Ish98, Rui00] oder die Berechnung einer Hyperkugel in einem hochdimensionalen Merkmalsraum [Che01] realisiert werden. Bei gleichzeitiger Berücksichtigung der als nicht-relevant bewerteten Ergebnisbilder stellt der adaptive Suchprozess eine überwachte Klassifikationsaufgabe bzw. ein *Online*-Klassifikationsproblem im Batchbetrieb dar. Für den Lernvorgang innerhalb einer Bildersuche sollten die folgenden Aspekte berücksichtigt werden [Zho03a]:

1. Da die Interaktion zwischen Mensch und System in Echtzeit erfolgt, sind kurze Antwortzeiten einer Bilddatenbank eine wichtige Systemvoraussetzung. Die eingesetzten Lernverfahren sollten daher ausreichend schnell sein und auf aufwendige Berechnungen, die den kompletten Datenbestand verarbeiten, verzichten.

2. Da einem Benutzer in jedem Schritt des Suchprozesses nur eine gewisse Anzahl an Bildern zugemutet werden kann und der Benutzer auch nur eine begrenzte Anzahl der Ergebnisbilder bewertet (in der Regel werden in einem Suchschritt weniger als zwanzig Bilder bewertet), ist der Umfang der resultierenden klassifizierten Stichprobe im Vergleich zur gespeicherten Bildmenge, der Dimension der Merkmalsvektoren und der implizit vorhandenen Klassenanzahl sehr gering. Unter dieser Voraussetzung liefern einige der existierenden Lernverfahren keine stabilen oder sinnvollen Ergebnisse [Tie00, Zho03a].

3. Da ein Anwender mit seiner Suchintention gewöhnlich die Zielsetzung verfolgt, Bilder einer bestimmten Kategorie zu finden, erfordert die inhaltsbasierte Suche keine binäre Klassifikation der gespeicherten Datenmenge. Vielmehr interessieren nur die Bilder, die der Anfrage am besten entsprechen. Diese sollten abhängig von ihrer Ähnlichkeit zur Anfrage geordnet sein und nur eine Teilmenge der Datenbank darstellen. Demnach sind die nicht-relevanten Bilder der Datenbank unter Berücksichtigung der Suchintention eines Benutzers in der Regel eher unwichtig. Gängige Klassifikationsverfahren wie die Diskriminanzanalyse [Dud73] oder Support Vektor Maschinen [Vap95] behandeln relevante und nicht-relevante Beispiele gleichberechtigt und setzen voraus, dass diese Elemente die zugrundeliegenden Verteilungen adäquat repräsentieren. Dieses ist jedoch gerade bei genauerer Betrachtung der irrelevanten Stichprobenelemente häufig nicht der Fall, da sie gewöhnlich nicht nur eine Klasse, sondern eine beliebige Menge von nicht-relevanten Klassen repräsentieren. Die binäre Trainingsmenge erfordert daher eine asymmetrische Handhabung, in der die relevanten Elemente

[15] Die folgenden Ausführungen beziehen sich ausschließlich auf ganze Bilder. Die vorgestellten Techniken sind jedoch auch auf Bildregionen übertragbar.

eine kompakte Klasse bilden und die nicht-relevanten Elemente eine gute Separierung bezüglich dieser Klasse aufweisen sollten [Zho01].

2.4.2 Varianten der adaptiven Bildersuche

Nachdem in den vorangegangenen Abschnitten die Grundlagen der adaptiven Bildersuche, basierend auf Mensch-Maschine Interaktion, erläutert wurden, werden in den nächsten Absätzen einige Verfahren zur Systemadaption vorgestellt. Dabei handelt es sich ausschließlich um Techniken, die das Kurzzeitlernen (engl. *Short-Term Learning*) innerhalb eines Suchprozesses (online) realisieren. Für eine Veranschaulichung unterschiedlicher Verfahren des Langzeitlernens (engl. *Long-Term Learning*) als auch eine weiterführende Betrachtung verschiedener Online-Lernvarianten sei an dieser Stelle auf die Arbeiten von Minka und Picard [Min96], Koskela und Laaksonen [Kos03] sowie Huang et al. [Hua02, Zho03a] verwiesen.

Die zuerst entwickelten Lernalgorithmen zur Bildersuche basieren auf den im Information Retrieval entwickelten Methoden der Termgewichtung und des Relevance Feedback [Sal89]. Ihre Grundlage bilden heuristische Verfahren sowie die Absicht, die Merkmale zu verstärken, die sowohl eine gute Gruppierung der positiven Beispiele erzielen als auch die negativen Beispiele von diesem Cluster separieren [Pic96, Rui98, Por99]. Rui et al. [Rui98] waren unter den Ersten, die diese Techniken auf die inhaltsbasierte Bildersuche übertrugen. Der Adaptionsprozess besteht dabei, neben einer Verfeinerung der Anfrage durch Anwendung des Verfahrens nach Rocchio (sog. *Query Refinement* bzw. *Query Vector Movement* [Roc71]), aus einer Neugewichtung der Achsen der ursprünglichen Merkmalsräume und einer Neugewichtung der verwendeten Bildrepräsentanten. „Systematischere“ Arbeiten [Ish98, Rui00] verzichten wiederum auf eine heuristische Adaption der internen Systemparameter und formulieren die inhaltsbasierte Bildersuche als Optimierungsproblem, dessen Ziel die Minimierung der Abstände der positiven Beispiele in den verschiedenen Merkmalsräumen ist. Dagegen integrieren Kherfi et al. [Khe02] auch negativ bewertete Bilder in ihr Verfahren zur Systemadaption. Mathematisch werden dabei die Intra- als auch die Intervarianzen der Merkmalsvektoren der negativ und positiv bewerteten Bilder optimiert.

Andere Forscher kombinieren Relevance Feedback und selbstorganisierende Karten (engl. *Self Organizing Map*, SOM). Laaksonen et al. [Laa00, Laa01] verwenden in einer Baumstruktur organisierte SOMs, um die gespeicherten Bilder in den verschiedenen Merkmalsräumen zu indizieren. Relevante und nicht-relevante Beispiele werden als positive und negative Impulse auf die Karten abgebildet. Durch anschließende Tiefpassfilterung werden die Kartenbereiche verstärkt, in denen sich positive Elemente konzentrieren. Umgekehrt werden die Bereiche, in denen negativ bewertete Bilder angesiedelt sind, geschwächt. Somit lassen sich die Merkmalskarten bestimmen, die die Suchintention eines Benutzers am besten repräsentieren. Bilder, die in den positiven

Bereichen der Karten angesiedelt sind und dazu einem Anwender noch nicht präsentiert wurden, sind gute Kandidaten, um im nächsten Suchschritt angezeigt zu werden.

Im Gegensatz zu den meisten Bildsuchsystemen, die auf verschiedenen hochdimensionalen Repräsentanten von Merkmalen wie Farbe, Textur und Form basieren (vgl. Abschnitt 2.2), verwenden Tieu und Viola [Tie00] mehr als 45000 „hoch selektive“, einfache Merkmale. Das Lernen der Klassifikationsfunktion in dem korrespondierenden Merkmalsraum erfolgt durch Anwendung eines Boosting-Verfahrens [Fre99]. Für jedes Merkmal wird separat ein schwacher Zwei-Klassen-Klassifikator trainiert, wobei sowohl die relevanten Stichprobenelemente als auch die zufällig bestimmten nichtrelevanten Elemente als gaußverteilt angenommen werden. Der angestrebte starke Klassifikator resultiert schließlich aus einer gewichteten Linearkombination der besten schwachen Klassifikatoren.

Einige weitere Verfahren verwenden wiederum Support Vektor Maschinen (SVM) [Vap95], um die Suchintention eines Benutzers zu lernen. Hong et al. [Hon00] formulieren den Lernprozess als Zweiklassenproblem. Die nicht-relevanten und relevanten Stichprobenelemente werden in einen hochdimensionalen Merkmalsraum transformiert, in dem sie linear separierbar sind und durch eine Hyperebene getrennt werden. Der Abstand eines relevanten Elements zu der Trennfläche bildet ein Maß für das Gewicht, mit dem es in den Adaptionsvorgang einfließt. Je größer der Abstand ist, desto stärker ist das Gewicht eines Stichprobenelements. Im Gegensatz zum erstgenannten SVM-Verfahren verzichten Chen et al. [Che01] auf eine Zweiklassendarstellung und verwenden stattdessen eine Einklassen-SVM zur adaptiven Bildersuche. Ziel dieses Ansatzes ist die Bestimmung der kleinsten Hyperkugel in einem hochdimensionalen Merkmalsraum, die nahezu alle Merkmalsvektoren der als relevant bewerteten Bilder beinhaltet. Die verschiedenen Bilder lassen sich schließlich auf der Grundlage einer kernbasierten Entscheidungsfunktion sortieren.

Entgegen der in der inhaltsbasierten Suche üblichen Vorgehensweise, Bilder einer bestimmten semantischen Kategorie zu suchen (vgl. auch Abschnitt 2.1), basiert das von Cox et al. [Cox98, Cox00] vorgestellte Verfahren auf dem Prinzip der Zielsuche. Die im Rahmen dieses Verfahrens gegebene Bewertung der Ergebnisbilder wird als relative Bewertung interpretiert, d.h. ein als relevant bewertetes Bild gilt dem gesuchten Bild als ähnlicher als die übrigen Bilder der Datenbank. Ausgehend von der Benutzeraktion wird schließlich das zugrundeliegende Wahrscheinlichkeitsmodell entsprechend der Bayes’schen Regeln modifiziert.

Das von Zhou und Huang [Zho01, Zho03b] vorgestellte Verfahren basiert auf der asymmetrischen Modellierung der relevanten und nicht-relevanten Beispiele. Im Gegensatz zu den meisten Verfahren, die den Online-Lernprozess als Ein- bzw. Zweiklassenproblem begreifen, wird in dieser Arbeit ein $(1 + x)$-Klassenproblem formuliert. Ausgehend von einer einfachen Modifikation der „mehrfachen“ Diskriminanzanalyse (*Multiple Discriminant Analysis*, MDA, vgl. z.B. [Dud73, S.118]) wird eine Abbildung der ursprünglichen Datenvektoren in einen Merkmalsraum bestimmt, in

dem sowohl die Merkmalsvektoren der relevanten Bilder eine kompakte Klasse bilden als auch die Vektoren der nicht-relevanten Beispiele möglichst weit von dieser Klasse entfernt sind. Zusätzlich wird demonstriert, dass durch die Erweiterung dieser Technik auf nicht-lineare Abbildungen eine Steigerung der Suchleistung erzielt werden kann.

Wie beschrieben ist der größte Nachteil der adaptiven Bildersuche der geringe Umfang der klassifizierten Stichprobe. Ein mit dieser Menge trainiertes Lernverfahren besitzt oftmals eine geringe Generalisierungsfähigkeit und modelliert die gesuchte Bildähnlichkeitsfunktion daher nur vage. Der von Wu et al. [Wu00] entwickelte D-EM Algorithmus[16] integriert die unklassifizierten Bilder ebenso in den Lernprozess wie die vom Anwender bewerteten Bilder. In diesem Verfahren werden durch die Kombination der Diskriminanzanalyse und EM Iteration zwei wesentliche Ziele erreicht. Zum einen werden die Modellparameter für die Zweiklassen-Annahme (relevant oder nicht-relevant) geschätzt und zum anderen wird der Unterraum bestimmt, in dem die Klassen der erweiterten Trainingsmenge am besten separiert sind. Aufbauend auf der resultierenden Abbildungsmatrix sowie den entsprechenden Klassenrepräsentationen in dem korrespondierenden Merkmalsraum wird schließlich ein Suchergebnis generiert. Die Ergebnisse der zugehörigen experimentellen Untersuchungen demonstrieren die Leistungsfähigkeit des Verfahrens, dessen Berechnung allerdings sehr aufwendig ist.

2.5 Beispiele inhaltsbasierter Bildsuchsysteme

Ebenso umfangreich wie die in den letzten Abschnitten beschriebenen Verfahren zur Merkmalsextraktion, Ähnlichkeitsbestimmung und Systemadaption ist die Anzahl der auf diesen Techniken basierenden Bildsuchsysteme. In den nächsten Abschnitten werden stellvertretend einige dieser Systeme vorgestellt.

2.5.1 QBIC

Das am IBM-Almaden-Forschungszentrum entwickelte Bildsuchsystem QBIC (*Query By Image Content*) war das erste kommerziell verfügbare inhaltsbasierte System [Nib93, Fal94, Fli95]. Seine Architektur und Techniken dienten häufig als Vorlage für spätere Systeme und haben großen Einfluss auf die Entwicklung der inhaltsbasierten Bildersuche gehabt. QBIC unterstützt sowohl die klassische Beispielanfrage als auch die Anfrage durch Skizzieren oder durch Spezifikation gesuchter Farben und Texturen (vgl. Abschnitt 2.1). Zur Bildrepräsentation dienen verschiedene Farb-, Textur- und Formmerkmale [Nib93]. Neben den in verschiedenen Farbräumen berechneten

[16] Diskriminanz Expectation Maximation Algorithmus

Durchschnittsfarben eines Bildes bzw. Objekts werden Farbhistogramme als Farbcharakteristika verwendet. Der Vergleich zweier Histogramme erfolgt durch die Verwendung des generalisierten euklidischen Abstands, dessen Gewichtsmatrix die Ähnlichkeiten der verschiedenen Farben beinhaltet (vgl. Abschnitt 2.3.3). Texturen innerhalb eines Bildes werden durch eine modifizierte Variante der von Tamura [Tam78] vorgestellten Merkmale Rauheit, Kontrast und Ausrichtung (engl. *Coarseness*, *Contrast* und *Directionality*) beschrieben [Nib93]. Die Formen von Bildobjekten werden durch die skalaren Attribute Fläche, Zirkularität, Exzentrizität, Hauptachsenorientierung sowie einen Satz von algebraischen Momenten repräsentiert. Zum Vergleich der gespeicherten Textur- und Formrepräsentanten wird der gewichtete euklidische Abstand verwendet.

Eine Skalierbarkeit auf große Datenmengen wird durch den Einsatz eines Indizierungsmechanismus erreicht. Dabei werden die zu speichernden Bildrepräsentanten zunächst in einem Vorverarbeitungsschritt mittels einer Karhunen-Loève-Transformation (vgl. z.B. [Nie83, S. 108ff.]) dimensionsreduziert. R*-Bäume [Bec90] dienen schließlich als multidimensionale Indexstrukturen und ermöglichen den schnellen Zugriff auf die Bilddaten [Fal94].

Ein entscheidender Nachteil des QBIC-Systems ist, dass es die automatische Verfeinerung interner Systemparameter auf der Grundlage von benutzergegebenen Bildbewertungen nicht unterstützt.

2.5.2 MARS

MARS (*Multimedia Analysis and Retrieval System*) ist ein Bildsuchsystem, das an der Universität von Illinois in Urbana-Champaign und an der Universität von Kalifornien in Irvine entwickelt wurde [Hua96, Ort97, Ser98]. Es stellt ein interdisziplinäres Forschungsprojekt dar, in dem Techniken aus den Bereichen des Computersehens, Datenbankmanagements und Information Retrieval miteinander kombiniert werden.

Die Basisbestandteile von MARS sind verschiedene Farb-, Farblayout-, Textur- und Formcharakteristika [Hua96, Rui98]. Die Repräsentation von lokalen Bildinhalten basiert auf der Gruppierung von Bildpunkten im Farb-Textur-Merkmalsraum. Als Gruppierungsverfahren wird der *k*-means Algorithmus verwendet. In einem abschließenden Verarbeitungsschritt werden mit einem speziellen aus der Physik (Anziehungskraft) motivierten Algorithmus Regionen miteinander verschmolzen, sodass die verschiedenen Bildobjekte extrahiert werden können [Hua96].

Die wichtigste Eigenschaft von MARS ist, dass ein Benutzers in den inhaltsbasierten Suchprozess einbezogen wird. Dabei wird auf eine manuelle Einstellung von internen Systemparametern verzichtet und stattdessen die aus dem Information Retrieval stammende Technik der Mensch-Maschine Interaktion durch Relevance Feedback auf die

merkmalsgetriebene Bildersuche übertragen [Rui97, Por99]. Zielsetzung ist es nicht, den besten Bildrepräsentanten zu finden, sondern zu lernen, wie die verwendeten Bildcharakteristika gewichtet und die internen Systemparameter, wie Ähnlichkeitsfunktionen und Anfragebeschreibungen, adaptiert werden müssen, um die semantischen Konzepte eines Benutzers formal beschreiben zu können [Rui98].

2.5.3 PicSOM

Das PicSOM-System wurde an der Universität Helsinki entwickelt [Bra00, Laa00, Laa01]. Seine Hauptcharakteristika sind eine Webschnittstelle, eine hierarchische Organisation der zu speichernden Bilder in selbstorganisierenden Karten (SOMs) sowie die Integration des vom Benutzer gegebenen Relevance Feedback. Analog zum WEBSOM System [Koh00], das Textdokumente in einem zweidimensionalen Gitter so organisiert, dass ähnliche Dokumente benachbart sind, werden in PicSOM Bilder auf der Basis ihrer extrahierten Charakteristika in selbstorganisierenden Karten strukturiert. Dabei wird für jeden Bildrepräsentanten eine Kaskade von SOMs (sogenannte TS-SOMs, engl. *Tree Structured SOMs*) erzeugt. Die eingesetzten Bildmerkmale umfassen verschiedene Farb-, Textur- und Formcharakteristika [Bra00, Laa00].

Die Generierung der Ergebnisliste basiert auf dem vom Benutzer gegebenen Relevance Feedback. Die als relevant bewerteten Bilder werden als positive Beispiele betrachtet, nicht bewertete gelten automatisch als negative Beispiele. Da jedes Bild in jeder der hierarchisch angeordneten SOMs einem Knoten zugeordnet ist, lassen sich diese Bewertungen als positive und negative Impulse auf die Karten abbilden. Durch einfache Tiefpassfilterung werden die Kartenbereiche verstärkt, in denen eine Konzentration positiver Elemente auftritt. Im Gegensatz dazu werden die Bereiche geschwächt, die sich in der Nähe der negativen Beispiele befinden. Die Bilder der positiven Bereiche, die einem Benutzer noch nicht präsentiert wurden, stellen potentielle Ergebnisbilder dar. Die endgültige Ergebnisliste resultiert schließlich aus der Kombination der Teilergebnisse der verschiedenen TS-SOMs-Ebenen [Laa01].

2.5.4 Weitere Bilddatenbanksysteme

Das Viper-System [Squ99] wurde an der Genfer Universität entwickelt und basiert auf der aus dem Information Retrieval stammenden Technik der invertierten Dateien (engl. *Inverted Files*). Das Bildsuchsystem SIMBA [Sig01] stammt vom Institut für Mustererkennung und Bildverarbeitung der Universität Freiburg. Seine Grundlage zur farb- und texturbasierten Bildersuche bildet die Extraktion lokaler Invarianten und die darauf berechneten Fuzzyhistogramme.

Das NETRA-Bilddatenbanksystem ist an der Universität von Kalifornien, Santa Barbara, im Rahmen des *Alexandria Digital Library Project* entwickelt worden [Ma97b].

Seine Hauptcharakteristika sind die Texturanalyse durch Gaborfilter und die Bildsegmentierung durch den Kantenflussalgorithmus [Man96, Ma97a]. Die in den resultierenden Bildsegmenten extrahierten Farb-, Textur- und Formmerkmale sowie die Positionsinformationen der Regionen bilden die Grundlage des inhaltsbasierten Suchprozesses.

Ebenfalls auf der Bildsegmentierung basiert das Blobworld-System [Car99] der Universität von Kalifornien in Berkeley und das von Wang et al. [Wan01] an der Pennsylvania State Universität entwickelte Bildsuchsystem SIMPLIcity.

Photobook wurde am MIT Media Lab entwickelt [Pen96]. Das Bildsuchsystem besteht aus drei Bestandteilen, die auf Textur-, Form- und Gesichtssuchen spezialisiert sind. In seiner neueren Version *Four Eyes* [Min96] wird außerdem die iterative Verfeinerung durch Relevance Feedback unterstützt.

Virage ist das neben QBIC wohl bekannteste kommerziell erhältliche Bilddatenbanksystem [Bac96]. Es wurde von Virage Incorporation entwickelt und unterstützt die inhaltsbasierte Bildersuche auf der Grundlage von Farbe, Farblayout, Textur und Struktur. Auch die Kombination dieser Charakteristika ist möglich, wobei die Gewichtung der einzelnen Merkmale vom Benutzer eingestellt werden kann. Darüber hinaus bietet Virage für Entwickler eine Schnittstelle zur Integration selbst entwickelter visueller Primitiva (Bildrepräsentanten) an.

2.6 Zusammenfassung

In diesem Kapitel wurden sowohl die grundlegenden Eigenschaften als auch der aktuelle Stand der Forschung der inhaltsbasierten Bildersuche vorgestellt. Dabei wurden neben einer einleitenden Darstellung der verschiedenen Anfrageparadigmen die unterschiedlichen Varianten zur Bildbeschreibung erläutert. Abgesehen von einigen speziellen, anwendungsabhängigen Deskriptoren werden vor allem verschiedene Repräsentanten der Merkmale Farbe, Textur und Form dazu verwendet, den Inhalt eines Bildes zu erfassen. Sie können sowohl global als auch lokal berechnet werden. Während die globale Merkmalsextraktion alle Bildpunkte einbezieht, ist die lokale Berechnung auf die Elemente einer Bildregion beschränkt. Der letztgenannte Ansatz erfordert daher entweder eine Bildsegmentierung oder die Detektion von interessanten Bildpunkten, in deren Umgebung die lokalen Bildbeschreibungen extrahiert werden.

Der Suchprozess eines inhaltsbasierten Bildsuchsystems basiert auf dem Vergleich von inhärenten Bildcharakteristika. Dabei werden die aus der Anfrage resultierenden Bildrepräsentanten mit denen der gespeicherten Bilder verglichen. Dies ist gleichbedeutend mit einer Abstandsberechnung. Je geringer der Abstand zweier Bildrepräsentanten ist, desto ähnlicher sind sich die korrespondierenden Bilder. Abstandsmaße, wie beispiels-

weise die in Abschnitt 2.3 beschriebenen Minkowski-Metriken, der generalisierte euklidische Abstand oder die Earth-Mover's-Distanz, sind daher neben Bildcharakteristika notwendige Bestandteile eines inhaltsbasierten Bildsuchsystems.

Zusätzlich wurde die Notwendigkeit für die Einbeziehung eines Benutzers in den Suchprozess motiviert und das daraus resultierende Verfahren der iterativen Bildersuche skizziert. Dabei wurde besonderen Wert auf die Erläuterung der speziellen Eigenschaften des Systemlernens in der inhaltsbasierten Bildersuche gelegt. Neben der Notwendigkeit für kurze Antwortzeiten stellt vor allem der geringe Umfang der klassifizierten Stichprobe ein Problem für den Lernvorgang dar. Wird außerdem vorausgesetzt, dass die relevanten Bilder eine kompakte Klasse bilden und die nicht-relevanten Bilder verschiedene Klassen repräsentieren, so ist eine asymmetrische Verarbeitung erforderlich. Dies resultiert sowohl aus der unbekannten Anzahl der nicht-relevanten Klassen als auch aus dem in der Regel geringen Umfang der nicht-relevanten Beispiele, die für die korrespondierenden Klassen wenig repräsentativ sind. Des Weiteren wurden verschiedene Verfahren der adaptiven Bildersuche vorgestellt, deren Ziel, ausgehend vom Relevance Feedback eines Benutzers, das Kurzzeitlernen während der Bildersuche ist. Abschließend wurden beispielhaft verschiedene inhaltsbasierte Bildsuchsysteme vorgestellt.

3 Das Bildsuchsystem INDI – Ein Systemansatz

Die in der Einleitung beschriebenen Szenarien aus der Werbeindustrie und des Journalismus sind typische Beispiele für den Einsatz digitaler Bilddatenbanken. Sie bilden die Grundlage eines Arbeitsprozesses und sollten daher einfach zu handhaben sein sowie effiziente Suchtechniken zur Navigation in der gespeicherten Bildsammlung zur Verfügung stellen. Das im Rahmen des BMB+F Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“ entwickelte Bildsuchsystem INDI[1] basiert auf eben diesen Anforderungen [Käm02, Bau03, Käs03a, Käs04]. Seine Hauptcharakteristika sind sowohl die in Abbildung 3.1 dargestellte natürliche Bedienung durch den Einsatz von Sprache und Touchscreen-Gestik als auch die Navigation in der gespeicherten Datenmenge auf der Grundlage inhaltsbasierter Suchtechniken. Durch die Kombination dieser beiden zentralen Eigenschaften grenzt sich das

Abb. 3.1: Natürliche Bildersuche durch multimodale Mensch-Maschine Interaktion

INDI System deutlich von anderen Bildsuchsystemen wie z.B. MARS [Hua96] oder PicSOM [Laa00] ab, deren Fokus hauptsächlich auf dem inhaltsbasierten Suchprozess liegt. Des Weiteren stellt das INDI System ein flexibles Bildsuchsystem dar, das nicht auf einen fixen Satz von Algorithmen beschränkt ist und einfach erweitert werden kann. Neu entwickelte bzw. für eine Bilddomäne spezialisierte Methoden der Bildverarbeitung können dem bestehenden Bildsuchsystem somit einfach zur Verfügung gestellt werden, ohne dass der zentrale Bilddatenbank-Server neu übersetzt werden muss.

[1] Das Akronym INDI steht für „**I***ntelligent* **N***avigation in* **D***igital* **I***mage Databases*“ und wurde aus dem Titel des BMB+F Teilprojekts abgeleitet.

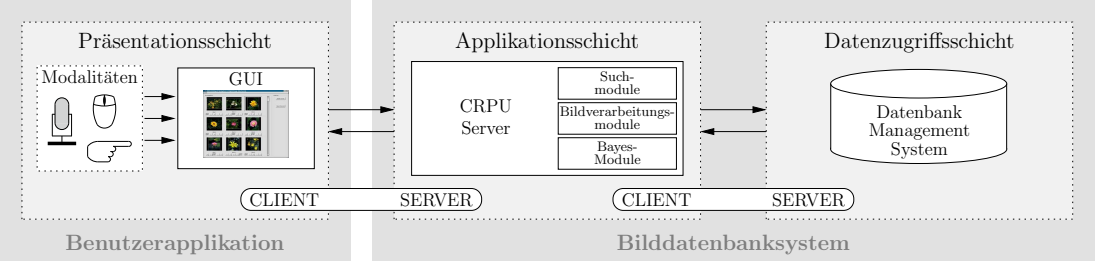


Abb. 3.2: Drei-Schichten-Architektur des Bildsuchsystems INDI: Die für die inhaltsbasierte Bildersuche notwendigen, bildbezogenen Daten werden mittels DBMS in Datenbanken gespeichert und verwaltet (rechts). Auf diese Daten greift die zentrale Einheit des Bildsuchsystems zu (Mitte). Die Haupteigenschaft dieses Servers ist die Bereitstellung von Funktionalitäten, die den Suchprozess realisieren. Die Anfrageformulierung an das INDI System erfolgt durch den multimodal zu bedienenden Datenbank-Client (links).

Systemaufbau

Die Entwicklung eines Bilddatenbanksystems erfordert eine genaue Spezifikation der zu verwaltenden Daten und Funktionalitäten. Dabei gilt es nicht nur, die erforderlichen Datentypen festzulegen, sondern auch ihre Abhängigkeiten und Aufgaben zu erfassen. Neben den Aspekten der Datenhaltung müssen die anwendungsspezifischen Abläufe definiert und die Beziehungen zwischen ihnen spezifiziert werden, um die inhaltsbasierte Suche in einer gespeicherten Datenmenge zu ermöglichen. Da es sich dabei um einen adaptiven Suchprozess handelt, der die Interaktion zwischen Mensch und System erfordert, müssen ebenfalls die Anforderungen an die Benutzerschnittstelle definiert werden. Die angestrebte natürliche Systembedienung erfordert außerdem, dass die speziellen Aspekte der multimodalen Interaktion wie beispielsweise die Kombination von Sprache und Gestik berücksichtigt werden.

Ein übersichtlicher und klar strukturierter Systemaufbau wird durch die Trennung der verschiedenen logischen Systemkomponenten erzielt. Die daraus resultierende Drei-Schichten-Architektur ist in Abbildung 3.2 dargestellt. Die zur Durchführung der inhaltsbasierten Suche elementaren Daten wie beispielsweise die vektoriellen Bildrepräsentationen oder Regionenbeschreibungen werden in einer Datenbank gespeichert und verwaltet. Neben dem Datenbankmanagementsystem (DBMS) wird das INDI System durch den Systemkern, den Bilddatenbank-Server, komplettiert. Diese Konfigurations- und Sucheinheit (engl. *Configuration and Retrieval Processing Unit*, CRPU) stellt hauptsächlich Funktionalitäten zur Verfügung, die die Durchführung der inhaltsbasierten Bildersuchen ermöglichen. Darüber hinaus beinhaltet der Server sowohl die für das Hinzufügen benutzerdefinierter Regionen notwendigen Bildverarbeitungsmodule als auch die für die Auflösung der sprachlichen Referenzierung erforderlichen Bayes-Netze (vgl. Abschnitt 3.3.2). Zur Anfrageformulierung dient die ei-

gentliche Benutzerapplikation, der grafische Datenbank-Client. Dieser ist in der Lage, verschiedene Modalitäten zu verarbeiten und ermöglicht somit eine natürliche und intuitive Bedienung.

In den folgenden Abschnitten werden sowohl die verschiedenen Bestandteile der Daten-, Applikations- und Präsentationsschicht des INDI Systems erläutert als auch ihre Kommunikation beschrieben. Das Ziel der Ausführungen ist es, dem Leser einen Überblick über den Aufbau des Bildsuchsystems zu vermitteln. Ausführlichere Erläuterungen und spezielle Details der Systemarchitektur können in der Arbeit von Pfeiffer [Pfe06] nachgeschlagen werden.

3.1 Datenhaltung und Datenrepräsentation

Die Grundlage jeder Softwareentwicklung bildet ausgehend von einem Anwendungsproblem ein Anforderungsprofil, in dem die Eigenschaften der zu realisierenden Anwendung spezifiziert sind. Aufbauend auf dieser Spezifikation können sowohl die elementaren Daten als auch erforderliche Systemfunktionalitäten definiert werden. In der folgenden Auflistung sind die zentralen Anforderungen dargestellt, die an das Bildsuchsystem INDI gestellt wurden[2]:

1. Entwicklung eines inhaltsbasierten Bildsuchsystems, das ähnliche Bilder durch den Vergleich der verschiedenen low-level Bildcharakteristika findet.
2. Verwendung globaler und lokaler Bildinformationen, wobei auch die Repräsentation von Bildausschnitten und deren Referenzierung möglich sein soll.
3. Das System soll lernfähig sein, sodass eine Adaption an die aktuelle Suchintention eines Anwenders erzielt werden kann.
4. Möglichkeit der natürlichen und intuitiven Systembedienung durch den Einsatz von Sprache und Gestik.
5. Entwicklung eines flexiblen Systems, das einfach konfiguriert und erweitert werden kann.

Basierend auf diesen Anforderungen lassen sich die Daten spezifizieren, die zur Durchführung einer inhaltsbasierten Bildersuche benötigt werden. Da diese Daten permanent verfügbar sein müssen, werden sie durch ein Datenbankmanagementsystem

[2] Ausgehend von der Fragestellung, inwieweit der in dieser Arbeit vorgestellte inhaltsbasierte Suchprozess auf umfangreiche Datenbestände skalierbar ist, sollte das Bildsuchsystem außerdem Mechanismen zur Indizierung hochdimensionaler Daten bereitstellen. Allerdings ist dabei zu beachten, dass diese Anforderung keine generelle Eigenschaft des im Rahmen des BMB+F Teilprojekts zu entwickelnden Bilddatenbanksystems darstellt.

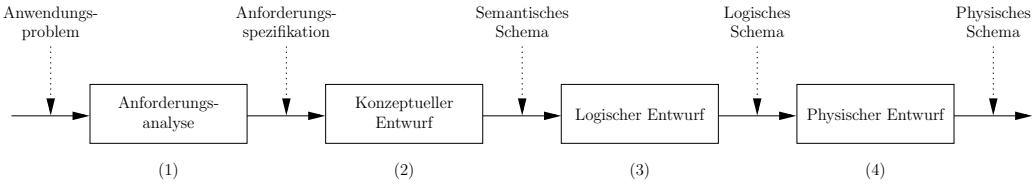


Abb. 3.3: Phasen des Datenbankentwurfs nach Lang und Lockemann [Lan95]: (1) Initial wird aus dem vorliegenden Anwendungsproblem ein Anforderungsprofil erstellt, in dem die notwendigen Daten erfasst und ihre Beziehungen untereinander formuliert werden. (2) Die aus der ersten Phase resultierende Anforderungsspezifikation wird im konzeptuellen Entwurf repräsentiert. (3) Anschließend erfolgt die Abbildung des semantischen Schemas auf das Datenmodell der verwendeten Datenbanksoftware. (4) Abgeschlossen wird der Datenbankentwurf durch die technische Umsetzung, den physischen Entwurf. Aus der anschließenden und erforderlichen Systemvalidierung resultiert gewöhnlich eine iterative Verfeinerung der einzelnen Entwurfsphasen.

in Datenbanken gespeichert. Als DBMS wird das frei verfügbare Datenbankprodukt MySQL (`http://www.mysql.com`) eingesetzt. Dieses hat sich schon in Vorarbeiten [Käs01] durch gute Performance ausgezeichnet. Außerdem unterstützt es sowohl die Speicherung von binären Datentypen als auch die Erweiterung des SQL Sprachumfangs durch die Anbindung von UDFs[3]. Die Datenbankstruktur der INDI Datenschicht resultiert aus dem in Abbildung 3.3 dargestellten Datenbankentwurf und wird in den nächsten Abschnitten inklusive der zu speichernden Daten und deren Repräsentationen näher erläutert.

3.1.1 Datenbankentwurf

Die Grundlage jedes Bilddatenbanksystems bilden die zu speichernden Bilder. Um den Speicherbedarf für die Datenbank und das Datenvolumen bei der Datenbanksicherung so gering wie möglich zu halten, werden die Bilder im Dateisystem abgelegt. In der Datenbank werden lediglich die absoluten Pfade der Bilder gespeichert. An dieser Stelle soll daher auf referentielle Integrität verzichtet werden. Ein Umstand, der durchaus zu vertreten ist, da zur Visualisierung der initialen Bildübersicht und der verschiedenen Ergebnisbilder lediglich verkleinerte Darstellungen[4], sogenannte *Thumbnails*, der zu speichernden Bilder verwendet werden. Die Originalbilder werden während des Datenbankbetriebs nur auf Anfrage der Benutzers zur Detailansicht eines Bildes sowie zur Visualisierung der verschiedenen Bildregionen benötigt.

[3] *User Defineable Function* (UDF): Spezifizierte Schnittstelle zur Entwicklung von Anwenderfunktionen, die den SQL Sprachumfang erweitern und in einer Datenbankanfrage aufgerufen werden können.
[4] Die verkleinerte Darstellung der zu speichernden Bilder motiviert sich aus den Anforderungen an die grafische Benutzerschnittstelle. Diese dient hauptsächlich zur Darstellung der Anfrage- und Ergebnisbilder, von denen möglichst viele auf einer Übersicht angezeigt werden sollen.

Entsprechend der in Kapitel 2 beschriebenen inhaltsbasierten Techniken wird jedes Bild durch einen oder mehrere Bilddeskriptoren repräsentiert. Diese Bildcharakteristika ermöglichen den Vergleich unterschiedlicher Bilder und müssen daher permanent verfügbar sein. Da das System nicht nur ausschließlich auf globale Bildbeschreibungen beschränkt sein soll, ist eine Darstellung von Bildregionen erforderlich (vgl. Anforderung 2 auf S. 43). Für die verschiedenen Bildregionen werden ebenfalls Repräsentanten berechnet, in diesem Fall lokale Repräsentanten, die auch in der Datenbank gespeichert werden müssen. Der Ähnlichkeitsvergleich der Bildcharakteristika erfordert außerdem die Definition von Distanzfunktionen (vgl. Abschnitt 2.3).

Ein wichtiger Bestandteil des in INDI zu realisierenden adaptiven Suchprozesses (vgl. Anforderung 3 auf S. 43) ist die Bewertung der verschiedenen Ergebnisbilder. Damit ein Benutzer auch die in den Bildern enthaltenen Bildregionen identifizieren und bezüglich seiner Suchintention bewerten kann, müssen diese Regionen zunächst extrahiert und anschließend visualisiert werden. Die dafür erforderliche Bildsegmentierung könnte zwar theoretisch zur Laufzeit innerhalb des Suchprozesses durchgeführt werden, allerdings ist diese Vorgehensweise nicht praktikabel. Die Extraktionen von Bildregionen ist gewöhnlich sehr rechenintensiv und sollte daher für jedes Bild nur einmal während der Datenbankinitialisierung erfolgen. Die resultierenden Regionenbeschreibungen werden schließlich in der Datenbank gespeichert, sodass sie permanent verfügbar sind. Die angestrebte natürliche Interaktion zwischen Mensch und System erfordert außerdem die Möglichkeit, Bildregionen auch sprachlich referenzieren zu können. Daher müssen neben elementaren Regionendaten wie beispielsweise der umgebene Kantenzug einer Region auch Regionenattribute wie z.B. Größe und Farbe gespeichert werden.

Die bisher vorgestellten Daten würden zwar als Basis eines inhaltsbasierten Bildsuchsystems ausreichen, die unter 5. geforderte Eigenschaft der Flexibilität wird bisher aber nicht erreicht. Voraussetzung für die einfache Erweiterbarkeit des Systems ist ein modularer Aufbau, sodass sowohl verschiedene Segmentierungsalgorithmen als auch unterschiedlichen Verfahren zur Merkmalsextraktion beliebig ausgetauscht werden können. Zusätzlich sollte eine zentrale Konfigurationstabelle die einfache Auswahl der zu verwendenen Bildrepräsentanten und Abstandsmaße ermöglichen.

Datenmodell

Aus den Anforderungen resultiert das in Abbildung 3.4 dargestellte semantische Schema. Die dabei zu verwaltenden Entitäten können in zwei Kategorien eingeteilt werden. Auf der einen Seite die Einheiten, die zur Initialisierung und Konfiguration des Bilddatenbanksystems benötigt werden (oberer Teil). Auf der anderen Seite die Entitäten, die zur Durchführung und Evaluation der inhaltsbasierten Bildersuche notwendig sind und auf die zur Laufzeit wiederholt zugegriffen wird (unterer Teil).

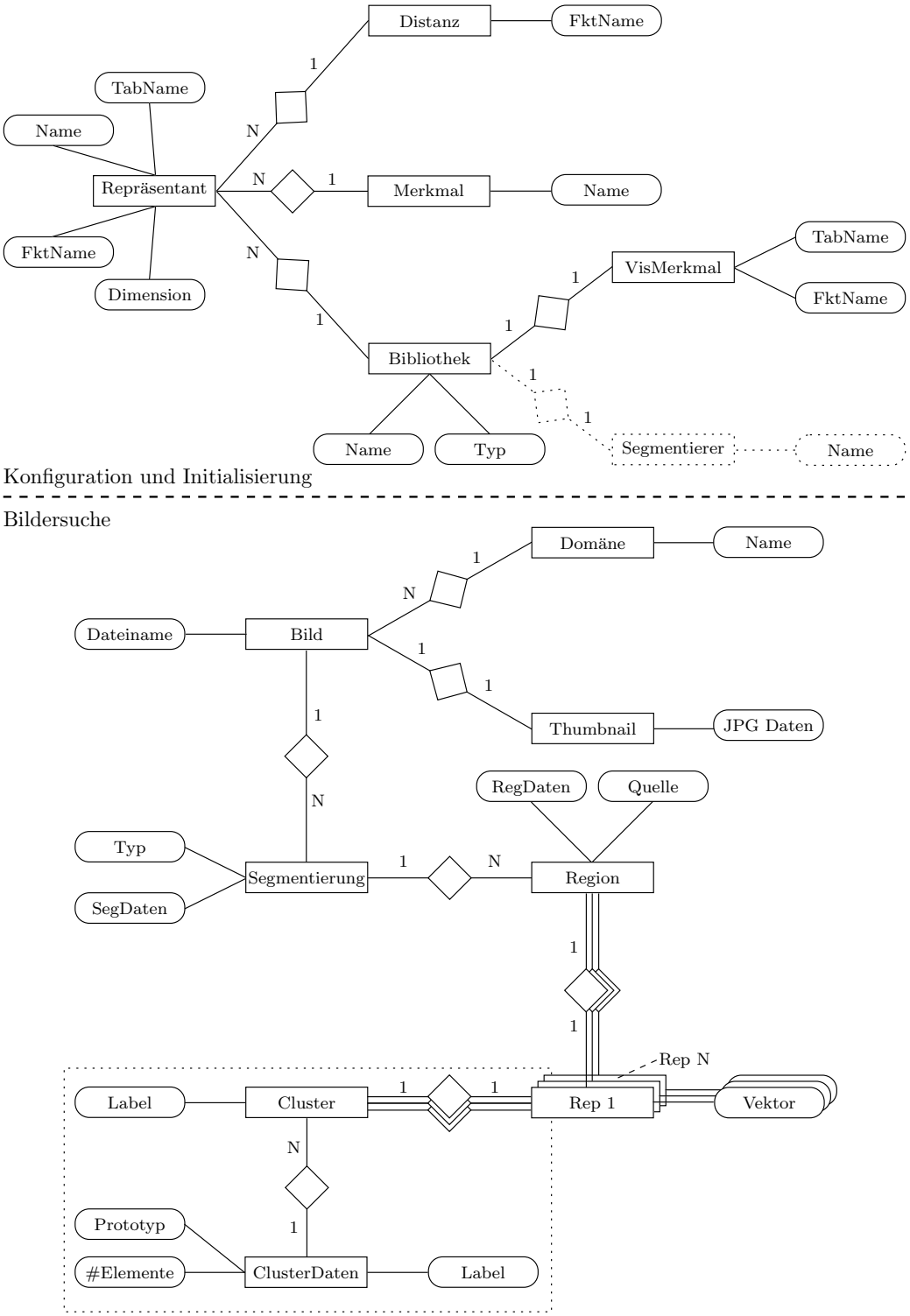


Abb. 3.4: Entity-Relationship-Diagramm: Die Entitätentypen des Bildsuchsystems INDI lassen sich in die Kategorien Konfiguration und Bildersuche unterteilen. Während die Attribute der einen Typen zur Initialisierung der Datenbank und des CRPU Servers dienen (oben), werden die Attribute der anderen Einheiten zur Durchführung der inhaltsbasierten Bildersuche benötigt (unten). Um eine übersichtlichere Darstellung zu erzielen, sind in dem Modell nur die wichtigsten Attribute aufgelistet.

Entitäten der ersten Kategorie sind die Metainformationen der verschiedenen Bildrepräsentanten, die für die inhaltsbasierte Suche eingesetzt werden sollen. Ihre wichtigsten Attribute sind der Name der Tabelle (*TabName*), in der die binären Deskriptoren gespeichert werden, die Bezeichnung des Repräsentanten (*Name*), der Name der Berechnungsfunktion (*FktName*) sowie die Dimension der mathematischen Beschreibung (*Dimension*). Das für den Vergleich der verschiedenen Deskriptoren notwendige Abstandsmaß wird durch den Gegenstandstyp *Distanz* spezifiziert. Unterschiedliche Repräsentanten lassen sich unterschiedlichen Merkmalen bzw. Merkmalsklassen zuordnen. Die entsprechenden Entitäten werden durch den Entitätentyp *Merkmal* charakterisiert. Im aktuellen Bildsuchsystem wird zwischen den Merkmalen Farbe, Textur und Form unterschieden. Aus welcher Programmbibliothek ein Repräsentant stammt wird durch den Gegenstandstyp *Bibliothek* definiert, der durch die Attribute *Name* und *Typ* beschrieben wird. Die zur Berechnung der Thumbnails notwendigen Bibliotheken werden durch den Entitätentyp *VisMerkmal* charakterisiert. Die korrespondierenden Attribute sind der Name der bildverarbeitenden Funktion (*FktName*) sowie der Name der Tabelle (*TabName*), in der die binären, verkleinerten Darstellungen der zu verwaltenden Bilder gespeichert werden. Analog zur Verbindung zwischen den Repräsentanten und den Bibliotheken sollte für die im System verfügbaren Segmentierungsverfahren (Gegenstandstyp *Segmentierer*) ebenfalls eine Beziehung zu den entsprechenden Entitäten des Typs *Bibliothek* bestehen. Allerdings ist der Gegenstandstyp *Segmentierer* in dem aktuellen System bislang nicht realisiert, da er für den Betrieb des Bildsuchsystems nicht zwingend notwendig ist. Deshalb wird seine Beziehung zum Entitätentyp *Bibliothek* in dem dargestellten ER-Diagramm nur angedeutet.

Die zu speichernden Bilder (Gegenstandstyp *Bild*) bilden den Mittelpunkt der für die inhaltsbasierte Bildersuche notwendigen Daten. Zur Evaluation der verschiedenen Systemkomponenten werden sie einer Bilddomäne zugeordnet. Außerdem existiert für jedes Bild eine Entität vom Typ *Thumbnail*, das als JPEG in der Datenbank gespeichert wird. Die Entitäten des Typs *Segmentierung* repräsentieren die verschiedenen Segmentierungsergebnisse der gespeicherten Bilder. Sie werden sowohl durch den Typ (*Typ*) als auch durch die vom Segmentierungsverfahren abhängigen spezifischen Daten (*SegDaten*) beschrieben. Letztere dienen zur Extraktion der aus der Bildsegmentierung resultierenden Bildregionen.

Die Attribute des Entitätentyps *Region* sind die *Quelle* und die binär gespeicherte Regionenstruktur *RegDaten*, in der neben dem Polygonzug der Kantenpunkte auch die verschiedenen Regioneneigenschaften wie Farbe oder Größe kodiert sind. Das Attribut *Quelle* dient zur Unterscheidung zwischen automatisch generierten und manuell hinzugefügten Bildregionen. Für jede Bildregion existieren entsprechend der Anzahl der verfügbaren Algorithmen zur Merkmalsextraktion unterschiedliche Entitäten vom Typ *Rep 1* bis *Rep N*, deren Attribut *Vektor* den binär abgespeicherten Merkmalsvektor

```
typedef struct t_SegInfo
{
  ...
  fnSegment *    ptSegment;       /* Zentrale Funktion zur Bildsegmentierung   */
  fnGetRegions * tGetRegFromSeg;  /* Extrahiert Regionen aus der Segmentierung */
  ...
} T_SegInfo;
```

Abb. 3.5: Wichtigste Attribute der INDI Segmentierungsstruktur. Mit der oberen Funktion wird ein Bild segmentiert und ein entsprechendes Segmentierungsergebnis erzeugt. Aus diesem Ergebnis werden mit der unteren Funktion die korrespondierenden Bildregionen extrahiert.

repräsentiert[5]. Die Eigenschaften dieser Gegenstandstypen werden durch die Entitäten des Typs *Repräsentant* beschrieben. Zusätzlich existieren mit den Gegenstandstypen *Cluster* und *ClusterDaten* zwei Komponenten, deren Entitäten die Grundlage für die in Kapitel 5 beschriebene zweistufige Bildersuche bilden. Das umgebene Rechteck symbolisiert dabei, dass es sich bei diesen Gegenstandstypen um Komponenten handelt, die hauptsächlich für experimentelle Zwecke in das Bildsuchsystem integriert wurden und bislang keinen festen Bestandteil des INDI Systems darstellen.

3.1.2 Datenrepräsentation

Die Zielsetzung eines flexiblen Bildsuchsystems, dessen essentiellen Komponenten wie Segmentierungsalgorithmen, Verfahren zur Merkmalsextraktion oder Distanzfunktionen austauschbar und erweiterbar sein sollen, erfordert die Definition eindeutig spezifizierter Schnittstellen und einen modularen Aufbau. Neu entwickelte Algorithmen müssen lediglich diese Spezifikationen erfüllen und können somit dem bestehenden System einfach als neue Module zur Verfügung gestellt werden.

Segmentierung

Ziel der Bildsegmentierung ist die Extraktion von Bildregionen, die unter Berücksichtigung gewisser Kriterien eine bestimmte Eigenschaft aufweisen (vgl. Abschnitt 2.2.5). Die Grundlage ihrer systeminternen Verarbeitung bildet eine eindeutige Repräsentation. Um eine größt mögliche Repräsentationsvielfalt zu erzielen, werden im INDI System Regionen durch Polygonzüge dargestellt. Zusätzlich beinhaltet die Regionenstruktur sowohl die Koordinaten der umgebenen Box (engl. *Bounding Box*) als auch

[5] Wieviele Gegenstandstypen *Rep 1* bis *Rep N* existieren hängt von der Anzahl der Entitäten des Gegenstandstyps *Repräsentant* ab. Dies bedeutet nichts weiter, als dass für jedes Verfahren zur Merkmalsextraktion eine Tabelle existiert, in der die entsprechenden Merkmalsvektoren der zu verwaltenden Bilder bzw. Bildregionen gespeichert werden.

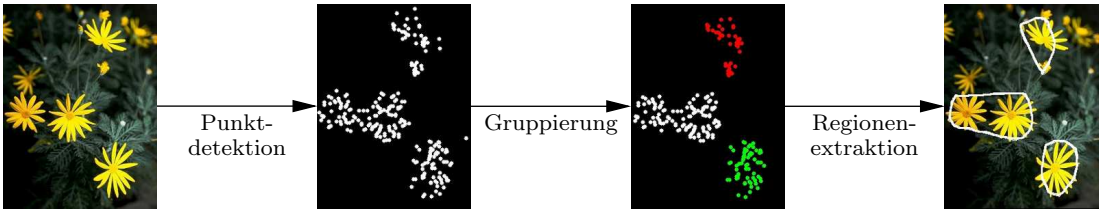


Abb. 3.6: Einzelne Phasen der ROI Bestimmung. Im ersten Schritt werden durch den erweiterten Harris-Punktdetektor [Gou01] die Fokuspunkte des zu verarbeitenden Bildes bestimmt. Diese werden im nächsten Schritt mittels SVC im Ortsraum gruppiert. Die resultierenden Punktwolken repräsentieren letztendlich die interessanten Regionen.

eine Binärmaske, in der die Pixel der Bounding Box durch Einsen und Nullen repräsentiert sind (vgl. Abschnitt 2.2.3). Die Klassifikation der Bildpunkte hängt davon ab, ob sie Element der Bildregion sind oder nicht. Komplettiert wird die Struktur durch eine Reihe von Attributen wie z.B. Farbe, Intensität, Größe, Exzentrizität oder Kompaktheit, die die Grundlage des sprachlichen Referenzierens bilden.

Die Segmentierungsverfahren des Bildsuchsystems INDI werden als Programmbibliotheken realisiert, die während der Initialisierung der Datenbank dynamisch geladen werden. Jedes dieser Module muss eine einheitliche Datenstruktur bereitstellen, deren wichtigsten Attribute in Abbildung 3.5 dargestellt sind.

Bildsegmentierung in INDI

Die in der inhaltsbasierten Bildersuche eingesetzten Segmentierungsverfahren sollten sowohl die Detektion der interessantesten Bildbereiche ermöglichen als auch einfach zu parametrisieren sein. Verfahren, die zur Übersegmentierung neigen und ein Bild in viele kleine Regionen unterteilen, würden den benötigten Speicherbedarf und den Aufwand des Suchprozesses dramatisch ansteigen lassen. Auch die Verfahren, die häufig nur durch manuelle Unterstützung zufriedenstellende Ergebnisse liefern, sind nicht geeignet. Unter Berücksichtigung der zu verwaltenden Datenmengen in einem Bilddatenbanksystem ist der Arbeitsaufwand für die manuelle Parametrisierung der Segmentierungsverfahren nicht zu erfüllen.

Als Konsequenz der aufgeführten Argumente wird im INDI System auf den Einsatz von Bildsegmentierungsverfahren wie z.B. die *Mean Shift*- oder *Color Structure Code*-Segmentierung (vgl. [Com97] und [Reh98]) verzichtet. Stattdessen wurde ein Verfahren entwickelt, das zur Detektion der interessantesten Bildbereiche (engl. *Regions-Of-Interest*, ROI) dient. Die Grundlage dieses Ansatzes, dessen einzelnen Verarbeitungsschritte in Abbildung 3.6 dargestellt sind, bilden die Fokuspunkte eines Bildes. Dabei wird in einem ersten Verarbeitungsschritt der auf die drei Farbkanäle des RGB-Farbraumes generalisierte Harris-Punktdetektor [Gou01] zur Detektion der interessan-

testen Bildpunkte eingesetzt. Häufungsgebiete der Fokuspunkte deuten auf Bildbereiche hin, die visuell interessant sind.[6] Deshalb werden die verschiedenen Punkte im nächsten Schritt gruppiert, sodass die interessantesten Bildbereiche bestimmt werden können. Die Gruppierung dieser in Ortskoordinaten repräsentierten Fokuspunkte erfolgt durch das Verfahren des *Support Vector Clustering* (SVC).

Die Grundidee des SVC Verfahrens, das auf den Arbeiten von Tax und Duin [Tax99] sowie Ben-Hur et al. [Ben02] basiert, ist die kompakte Repräsentation der Ursprungsdaten in einem hochdimensionalen Merkmalsraum. Dabei wird die Hyperkugel gesucht, die alle transformierten Datenpunkte einhüllt und deren Radius gleichzeitig so gering wie möglich ist. Nach der Lösung dieses Optimierungsproblems wird die Hyperkugel in den ursprünglichen Datenraum zurück transformiert. Die daraus resultierenden Konturen repräsentieren schließlich die gesuchten Cluster. Punkte, die von derselben Kontur eingeschlossen werden, gehören zum selben Cluster. Ein Vorteil dieses Verfahrens ist, dass es beliebige Clusterformen erlaubt und nicht auf sphärische Strukturen beschränkt ist. Außerdem erfordert es lediglich die Einstellung von zwei Parametern, wobei der eine die Anzahl der Support Vektoren kontrolliert und der andere zur Lockerung des Optimierungsproblems dient (*Hard* oder *Soft Margin*). Jeder resultierende Cluster repräsentiert letztendlich eine interessante Bildregion.

Merkmalsextraktion

Ebenso wie die Segmentierungsverfahren sind die Algorithmen zur Berechnung der Bildrepräsentanten als dynamisch ladbare Programmbibliotheken realisiert. Diese Bibliotheken werden sowohl während des Initialisierungsprozesses als auch beim Start des Bilddatenbank-Servers geladen. Letzteres ist notwendig, da das System die Integration benutzerdefinierter Bildregionen unterstützt und für diese natürlich auch die verschiedenen Deskriptoren berechnet werden müssen. Die dynamische Handhabung ist nur durch eine klar spezifizierte Schnittstelle möglich. Deshalb stellt jede Merkmalsbibliothek eine eigene Informationsstruktur zur Verfügung. Diese beinhaltet die für die Berechnung und Verwaltung der Bildrepräsentanten erforderlichen Informationen, wie:

- die Dimension des Merkmalsvektors
- der Funktionsname der Berechnungsroutine
- der Name des Repräsentanten (z.B. Farbhistogramm)

[6] Dabei ist zu beachten, dass die Bereiche unter Berücksichtigung eines bestimmten Kriteriums als visuell interessant betrachtet werden. Im konkreten Fall des Harris-Punktdetektors bedeutet dies, dass in den Bereichen verstärkt Krümmungen (in Form von Kanten und Ecken) des Bildsignals auftreten und daher den Fokus eines Betrachters auf diese Bildbereiche lenken (vgl. auch Abschnitt 2.2.4).

- die Klasse des Repräsentanten (z.B. Farbe, Textur oder Form)
- die systemweite eindeutige Identifikationsnummer
- die Identifikationsnummer des zu verwendenen Abstandsmaßes[7]

Jede Berechnungsroutine einer Bibliothek erhält als Übergabeparameter eine Bildregion[8] und liefert als Resultat den entsprechenden Merkmalsvektor. Da verschiedene Bildrepräsentanten oftmals dieselbe Berechnungsgrundlage besitzen, wie z.B. denselben Farbraum, ist es sinnvoll, dass eine Merkmalsbibliothek die Berechnung mehrerer verwandter Repräsentanten unterstützt.

Architekturunabhängige Repräsentation

Bei dem Bildsuchsystem INDI handelt es sich um eine verteilte Anwendung, bei der Client- und Server-Applikation auf unterschiedlichen Architekturen laufen können. Da verschiedene Systemkomponenten, wie z.B. die Merkmalsvektoren, die Regionenstrukturen oder die Thumbnails der Bilder, binär in der Datenbank gespeichert und von unterschiedlichen Systemkomponenten verarbeitet werden, ist eine architekturunabhängige Datenrepräsentation erforderlich. Deshalb wurde die für das Bielefelder *Distributed Application's Communication System* (DACS) entwickelte Netzwerkdatenrepräsentation (NDR) (vgl. [Jun98, Kapitel 4]) um notwendige Datentypen erweitert und in das Bildsuchsystem integriert. Vor dem Speichern werden die Binärdaten in diese Repräsentation konvertiert bzw. eingepackt. Wird zur Laufzeit auf sie zugegriffen, müssen sie vor der eigentlichen Verarbeitung wieder ausgepackt werden. Die für den Ein- und Auspackvorgang notwendigen Funktionen werden jeweils automatisch vom NDR Compiler generiert (vgl. [Pfe06]).

3.1.3 SQL Erweiterung und Abstandsberechnung

Das eingesetzte DBMS bietet mit seiner UDF Schnittstelle die Möglichkeit, den SQL Sprachumfang zu erweitern, sodass sich für die inhaltsbasierte Suche notwendige Prozesse vom Bilddatenbank-Server auf den Server des Datenbank-Backends auslagern lassen. Obwohl damit die strickte Trennung zwischen Applikations- und Datenschicht (vgl. Abbildung 3.2) aufgehoben wird, können Funktionalitäten des verwendeten Datenbankprodukts genutzt und die Systemperformance gesteigert werden.

[7] Dabei handelt es sich lediglich um die Standardeinstellung des Bildsuchsystems. Da das System konfigurierbar ist, können die Abstandsmaße eines Bildrepräsentanten durchaus variiert werden.
[8] Ganze Bilder werden in INDI auch als Regionen repräsentiert, sodass die verfügbaren Bildrepräsentanten auch für sie berechnet werden können und keine spezielle Verarbeitung erforderlich ist.

Da die verschiedenen Bildrepräsentanten in der Datenbank gespeichert werden, bietet es sich an, diese in der Datenschicht mit den Anfragevektoren zu vergleichen. Schließlich ist es effizienter, nur einige wenige Merkmalsvektoren vom INDI Server an den Server des Backends zu übertragen, als alle bzw. viele Merkmalsvektoren in umgekehrter Richtung. Ideal wäre dabei eine Übertragung der Anfragevektoren durch eine SQL Anfrage, die gleichzeitig die Abstandsberechnung zu den gespeicherten Bildrepräsentanten veranlasst und als Antwort die verschiedenen Abstandswerte oder eine entsprechend den Distanzen sortierte Bildliste zurückliefert. Um dies zu erreichen, wird der SQL Sprachumfang des MySQL Servers um verschiedene Funktionen zur Distanzberechnung, wie z.B. der generalisierte euklidische Abstand oder der Histogrammschnitt, erweitert. Der Abstand zweier Merkmalsvektoren lässt sich demnach durch die SQL Anfrage

```
SELECT DistanzFunktion('<AVektor>','<RepVektor>','Dim',
                       '<GewichtsVektor>','GewichtsDim')
```

berechnen. Dabei repräsentiert `DistanzFunktion` eine als UDF[9] implementierte Distanzfunktion, die als Argumente die zu vergleichenden Merkmalsvektoren `AVektor` und `RepVektor` erhält. Der dritte Parameter besteht aus der Dimension `Dim` der zu vergleichenden Repräsentanten. Um eine Gewichtung der unterschiedlichen Merkmalskomponenten zu ermöglichen, wird ein entsprechender Gewichtsvektor `GewichtsVektor` übergeben. Seine Dimension wird durch `GewichtsDim` spezifiziert. Für Abstandsmaße, die eine Gewichtsmatrix zur Distanzberechnung benötigen, wie z.B. der generalisierte euklidische Abstand, wird statt eines Vektors einfach eine Matrix übergeben. Die Dimension `GewichtsDim` entspricht dann dem Quadrat der Vektordimension `Dim`.

Die für eine gegebene Anfrage relevanten Bilder der Datenbank lassen sich durch eine einfache Erweiterung der beschriebenen SQL Anfrage bestimmen. Dies wird nun am Beispiel der in Abbildung 3.7 dargestellten Datenbanktabellen erläutert. Es wird angenommen, dass jedes zu speichernde Bild durch einen Bilddeskriptor repräsentiert wird, der in der Spalte `Vektor` der Datenbanktabelle `tblRep` gespeichert ist. In derselben Tabelle wird außerdem eine Identifikationsnummer `ID` abgelegt, sodass ein Merkmalsvektor dem korrespondierenden Bild zugeordnet werden kann. Sei nun ein Anfragevektor `AVektor` gegeben, dann lassen sich die Bilder, die den geringsten euklidischen Abstand (`EDistanz`) zu der Anfrage besitzen, durch folgende Datenbankanfrage bestimmen:

tblBilder:

ID	Dateiname
1	flowers0032.jpg
2	autoracing024.jpg
3	golf067.jpg

tblRep:

ID	Vektor
1	(0.1,0.4,0.6,0.8)
2	(0.2,0.2,0.1,0.4)
3	(0.5,0.7,0.9,0.1)

Abb. 3.7: Beispieltabellen

[9] Auf die technischen Besonderheiten dieser produktspezifischen Schnittstelle wird an dieser Stelle nicht weiter eingegangen. Nähere Informationen dazu finden sich in der Onlinedokumentation des Datenbankprodukts MySQL (`http://www.mysql.com`).

```
SELECT tblRep.ID, EDistanz('<AVektor>',tblRep.Vektor,'Dim',
                           '<GewichtsVektor>','GewichtsDim')
                           AS Distanz
                           FROM tblRep ORDER BY Distanz
```

Das Ergebnis der Anfrage ist eine sortierte Liste von Bild-IDs und Distanzen, die entsprechend ihres Abstandes (`Distanz`) zum Anfragevektor sortiert sind. Damit wurde demonstriert, wie durch die einfache Formulierung einer SQL Anfrage und die Verwendung der von MySQL angebotenen UDF Schnittstelle der Abstand von den gespeicherten Bildern zu einer gegebenen Anfrage berechnet werden kann. Auf eine aufwendige Übertragung von Merkmalsvektoren aus der Daten- zur Applikationsschicht kann somit verzichtet werden. Dieser Ansatz erfordert lediglich, dass die Anfrage- und Gewichtsvektoren so vorverarbeitet werden, dass sie in eine SQL Anfrage integriert werden können.

3.1.4 Datenbankinitialisierung

Die Inbetriebnahme des Bilddatenbanksystems erfordert die Initialisierung der notwendigen Datenbank. Zielsetzung dieses Prozesses ist die Bereitstellung der benötigten Konfigurations- und Suchdaten, sodass die Voraussetzungen für die inhaltsbasierte Bildersuche geschaffen sind. Die Initialisierung lässt sich in zwei Phasen unterteilen: Konfiguration und Population.

Konfiguration

In der Konfigurationsphase müssen die notwendigen Datenbanktabellen angelegt und die verschiedenen Konfigurationsdaten gespeichert werden. Die dafür entwickelte Konfigurationsapplikation erhält als Übergabeparameter lediglich den gewünschten Datenbanknamen sowie eine Textdatei, in der neben den Befehlen zum Anlegen der Datenbanktabellen und zur Integration der Distanzfunktionen auch die verschiedenen Bibliotheken zur Bildsegmentierung und Merkmalsextraktion spezifiziert sind. Die resultierende Datenbankstruktur enthält schließlich die Informationen über die verfügbaren Segmentierungs- sowie Merkmalsberechnungsalgorithmen, die in der folgenden Population verwendet werden.

Population

Nach der initialen Konfiguration der Datenbank müssen sowohl die Bildregionen extrahiert als auch die verschiedenen Bildrepräsentanten der zu speichernden Bilder berechnet und in der Datenbank gespeichert werden. Da die Bilder im Dateisystem und

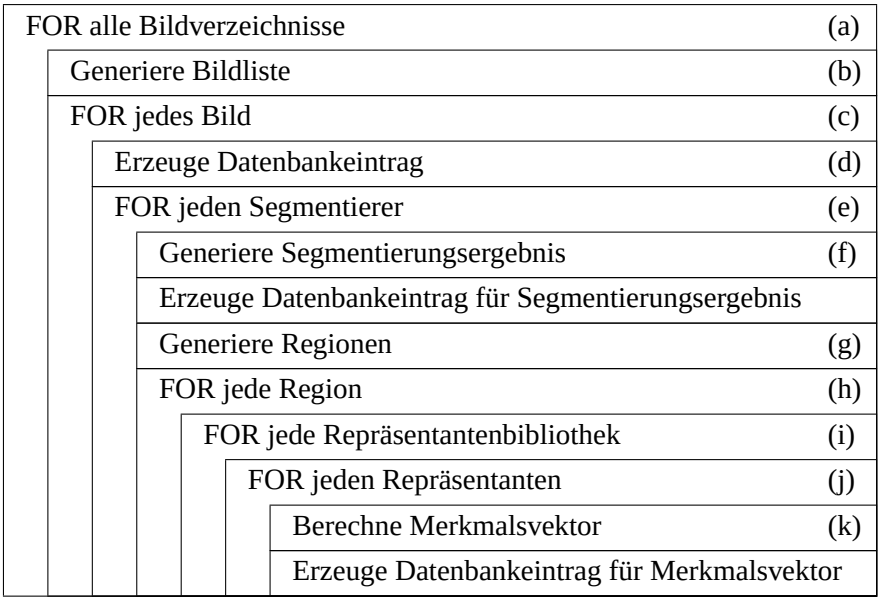


Abb. 3.8: Ablauf der Datenbankpopulation

nicht in der Datenbank abgelegt werden, muss der Populationsapplikation mitgeteilt werden, wo sich die zu speichernden Bilder im Dateisystem befinden. Dieses geschieht mit einer Textdatei in der die Verzeichnisse angegeben werden, in denen die zu speichernden Bilder gesammelt sind. Für jedes dieser Bildverzeichnisse wird der in Abbildung 3.8 dargestellte Populationsprozess durchlaufen. Im ersten Schritt wird für jedes Bildverzeichnis (a) die Liste der zu speichernden Bilder generiert (b). Danach wird für jedes Bild dieser Liste (c) wie folgt fortgefahren: Zuerst wird ein Datenbankeintrag erzeugt (d), in dem neben dem Dateinamen auch Bildattribute wie Höhe, Breite und Domäne gespeichert werden. Anschließend werden die verfügbaren Segmentierungsfunktionen angewendet (e). Es wird ein Segmentierungsergebnis generiert und ein Datenbankeintrag mit den spezifischen Segmentierungsdaten angelegt (f). Darauf aufbauend werden die verschiedenen Bildregionen extrahiert (g). Im nächsten Schritt wird für jede Region des Bildes (h) die Liste der verfügbaren Repräsentantenbibliotheken abgearbeitet (i). Für jeden Repräsentanten einer Bibliothek wird die vektorielle Repräsentation berechnet und ein Datensatz erzeugt (k). Damit existiert für jede Bildregion eine Menge von Repräsentanten (Merkmalsvektoren), die die Grundlage der inhaltsbasierten Bildersuche bilden.

Optimierung der Merkmalsrepräsentanten

Zusätzlich zum eigentlichen Initialisierungsprozess wurden noch verschiedene Optimierungswerkzeuge entwickelt. Diese können sowohl zur Normierung als auch zur Dimensionsreduktion der gespeicherten Merkmalsvektoren verwendet werden. Des Weiteren ermöglichen sie auch die Verkettung einzelner Bildrepräsentanten.

Inbetriebnahme des Bildatenbanksystems

Mit der Initialisierung der Datenbank wird die Grundlage für die Inbetriebnahme des Bildsuchsystems gelegt. Alle zur Durchführung der inhaltsbasierten Bildersuche notwendigen Konfigurationsdaten, Bildrepräsentanten und Abstandsfunktionen sind nun verfügbar. Die für den Betrieb des Bildsuchsystems notwendigen Metainformationen wie z.B. die Verbindungsdaten des DBMS Servers (Host, Benutzer, Passwort) oder die zu verwendene Datenbank müssen in einer Konfigurationsdatei eingetragen werden. Des Weiteren beinhaltet die Datei noch spezielle Parameter des Suchprozesses wie beispielsweise die bevorzugte Strategie zur Kombination von Einzelergebnissen (vgl. Abschnitt 4.2).

3.2 Bilddatenbank-Server

Das Kernstück des Bildsuchsystems INDI bildet der Applikations-Server. Diese Konfigurations- und Sucheinheit stellt sowohl Funktionalitäten zur Verwaltung der eingehenden Datenbank-Clients als auch zur Durchführung der inhaltsbasierten Bildersuchen bereit. Die dabei benötigten administrativen und ausführenden Funktionalitäten werden von unterschiedlichen Threads realisiert.

Damit überhaupt eine Kommunikation zwischen einem Datenbank-Client und dem Server stattfinden kann, muss zunächst eine Verbindung aufgebaut und anschließend verwaltet werden. Diese Aufgabe übernimmt ein eigenständiger Thread, der außerdem zwischen Single- und Multi-Client unterscheidet. Der Single-Client, der meistens die Standardbetriebsart darstellt, baut einmal eine Verbindung zum INDI Server auf und hält diese solange bis das Durchsuchen der gespeicherten Datenmenge abgeschlossen ist und die Applikation beendet wird. Alle Anfragen des Clients werden über diese Verbindung verschickt und sequentiell abgearbeitet. Ein Multi-Client dagegen kommuniziert mit dem Server-Prozess über mehrere Verbindungen, da er nicht in der Lage ist, eine Verbindung permanent zu halten. Dieses gilt beispielsweise für einen Webbrowser. Damit eine angefangene Bildersuche, die in der Regel iterativ ist, jedoch fortgesetzt werden kann, ist eine Identifikation des Multi-Clients notwendig, sodass bereits existierende Suchdaten dem Client zugeordnet werden können. Da das INDI System einen adaptiven, schrittweisen Suchprozess realisiert, in dem die internen Systemparameter basierend auf den Benutzerbewertungen kontinuierlich verfeinert werden (vgl. Abschnitt 2.4), müssen bei einem Multi-Client ständig neue Anfrage-Threads erzeugt und aufgegeben werden. Um den Aufwand für diese Threadverwaltung so gering wie möglich zu halten, werden erzeugte und nicht mehr benötigte Threads daher suspendiert. Sie lassen sich bei Bedarf durch den administrativen Thread, der sie verwaltet, reaktivieren.

Die Unterteilung der verschiedenen Systemprozesse auf mehrere Threads hat außerdem einen entscheidenden Vorteil. Da sich die Threads einen Adressraum teilen,

können sie auf dieselben Daten zugreifen. Der Bilddatenbank-Server unterscheidet die folgenden elementaren Datenstrukturen:

Server-Daten: Informationen, die vom Server global zur Verfügung gestellt werden und von verschiedenen Threads genutzt werden können. Dies sind z.B. Informationen über geladene Programmbibliotheken oder eine Auflistung der suspendierten Threads.

Session-Daten: Das sind die essentiellen Daten, die zur Durchführung einer Bildersuche, die auch als Session bzw. Such-Session bezeichnet wird, benötigt werden. Sie müssen für jeden Client gehalten werden. Dabei handelt es sich unter anderem um das aktuelle Beispielbild, verwendete Merkmalsrepräsentanten, unterschiedliche Gewichtsparameter oder das vom Benutzer gegebene Relevance Feedback.

Client-Daten: Neben den Referenzen auf die bereits beschriebenen Datenstrukturen werden in den Client-Daten die für die TCP/IP Verbindung notwendigen Informationen gespeichert.

3.2.1 Schnittstelle und Kommunikation

Die Kommunikation zwischen Client und Server ist paketorientiert und erfolgt nach dem Prinzip von Frage und Antwort (engl. *Request and Reply*). Der Client hüllt die Anfragen in entsprechende Paketstrukturen und sendet diese an den Server. Dieser analysiert den Paketkopf, entpackt das gesendete Paket, bearbeitet die Anfrage und sendet seinerseits die verpackte Antwort an den Client. Welche Funktionen der Server ausführen muss geht aus dem jeweiligen Kopf der eingehenden Pakete hervor. Im wesentlichen sind dies z.B. das Starten einer Suchiteration, die Anfrage nach einer neuen initialen Bildmenge oder die Anfrage nach einer verkleinerten Darstellung eines bestimmten Bildes. Damit die Kommunikation auch architekturunabhängig erfolgen kann, werden die zu versendenden Daten in die bereits erwähnte Netzwerkdatenrepräsentation konvertiert. Eine Auflistung aller verfügbaren Pakete und die ausführliche Beschreibung ihrer Inhalte findet sich in der Arbeit von Pfeiffer [Pfe06].

3.2.2 Funktionen der inhaltsbasierten Bildersuche

Die wichtigsten Elemente des INDI Servers sind die Module zur Durchführung der inhaltsbasierten Bildersuche. Diese müssen die essentiellen Suchfunktionalitäten zur Verfügung stellen, wobei jedoch eine möglichst große Flexibilität erreicht werden soll. Als Übergabeparameter erhalten die Suchmodule lediglich die vom Bilddatenbank-Server gehaltenen Session-Daten, sodass bereits durchgeführte Suchschritte einer Bildersuche fortgesetzt werden können.

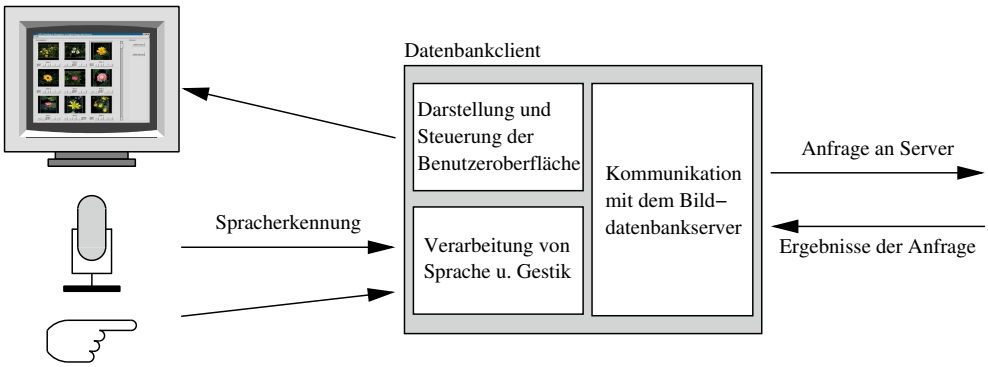


Abb. 3.9: Aufbau des Datenbank-Clients. Neben der Darstellung der grafischen Benutzeroberfläche und der Steuerung ihrer Interaktionselemente verarbeitet und synchronisiert der Client die Modalitäten Sprache und Gestik. Für die Suche relevante Daten werden von ihm einerseits an den Bilddatenbank-Server gesendet. Andererseits fordert er Informationen an, die zur Visualisierung der Ergebnisse benötigte werden.

Die größte Flexibilität wird durch eine hierarchische Anordnung der verschiedenen Funktionalitäten erreicht. Für jede Suche wird ein eigener Suchstrang gestartet. So kann beispielsweise gleichzeitig nach mehreren Regionen oder Bildern gesucht werden. Ein Gesamtsuchergebnis würde sich dann aus der Kombination der Einzelergebnisse ergeben. Jeder Suchstrang wiederum ist ebenfalls hierarchisch organisiert, da in jedem der verfügbaren Merkmalsräume separat nach ähnlichen Bildern gesucht wird. Die Ergebnisse der verschiedenen Merkmalsräume werden schließlich zu einem Gesamtergebnis zusammengefasst.

Neben seiner Flexibilität hat der hierarchische Ansatz noch einen weiteren wichtigen Vorteil. Verschiedene Funktionalitäten lassen sich parallelisieren und können daher in eigenen Threads realisiert werden. Die inhaltsbasierte Bildersuche lässt sich daher auf einem Mehrprozessorrechner beschleunigen und die Systemantwortzeiten dementsprechend verkürzen, ohne dass die verschiedenen Suchfunktionalitäten modifiziert werden müssen. Weitere Details und eine ausführliche Formalisierung der für das INDI System entwickelten inhaltsbasierten Suchmechanismen werden in Kapitel 4 vorgestellt.

3.3 Datenbank-Client

Der Datenbank-Client ist die eigentliche Benutzerapplikation. Seine Aufgabe ist sowohl die Darstellung der grafischen Benutzeroberfläche als auch die Verarbeitung der Aktionen ihrer Interaktionselemente wie z.B. das Drücken einer Schaltfläche (engl. *Button*) oder das Ziehen der Bildlaufleiste (engl. *Scroll Bar*). Zusätzlich verarbeitet und synchronisiert der Client die natürlichen Modalitäten Sprache und Gestik.

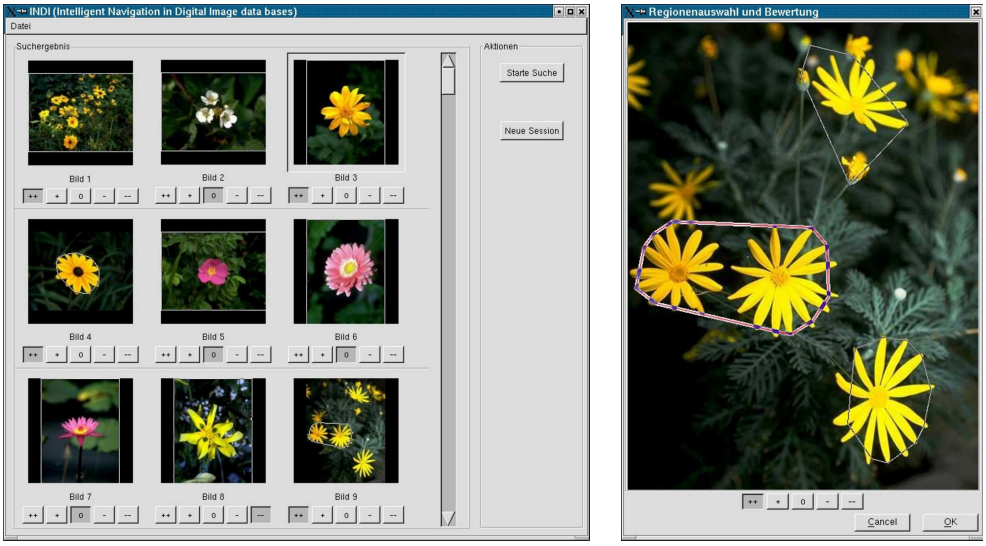


Abb. 3.10: Grafische Oberfläche der Benutzerapplikation. Die Hauptansicht der grafischen Benutzerschnittstelle (links) dient zur Auswahl des initialen Beispielbildes bzw. der initialen Beispielregion, zur Darstellung der Ergebnisbilder sowie zur Bewertung der Suchergebnisse. Regionen eines Bildes lassen sich in dem Regioneneditor (rechts) auswählen und bewerten. Zusätzlich ermöglicht er die Integration von benutzerdefinierten Bildausschnitten.

Durch die Kommunikation mit dem Bilddatenbank-Server sendet er einerseits die zur Durchführung der inhaltsbasierten Bildersuche notwendigen Anfragedaten und andererseits fordert er vom Server Daten an, die zur Visualisierung der Suchergebnisse benötigt werden. Abbildung 3.9 skizziert die wichtigsten Bestandteile des Datenbank-Clients. In den folgenden Abschnitten werden neben der grafischen Benutzerschnittstelle (engl. *Graphical User Interface*, GUI) die verfügbaren Modalitäten der Systembedienung beschrieben und ihre Kombination erläutert.

3.3.1 Grafische Benutzerschnittstelle

Bindeglied zwischen Anwender und Bilddatenbanksystem ist die grafische Benutzerschnittstelle. Diese dient zur Anfrageformulierung, Darstellung der Suchergebnisse und Bewertung der Bilder bzw. Bildregionen. Abbildung 3.10 stellt die verschiedenen Komponenten dieser Benutzerapplikation dar. Zentrale Elemente sind die Bilder der Datenbank. Ausgehend vom Anfrageparadigma der Beispielsuche (vgl. Abschnitt 2.1) wird initial eine zufällig generierte Bildmenge präsentiert. Ansonsten werden die jeweiligen Ergebnisbilder einer Suchanfrage dargestellt. Das Beispielbild einer Suche ist durch ein umgebenes Rechteck gekennzeichnet und auf einer Übersicht sind jeweils neun Bilder in reduzierter Auflösung dargestellt. Weitere Bilder können durch Verschieben der Bildlaufleiste angeschaut werden. Bilder bzw. markierte Bildregio-

nen können durch die verschiedenen Knöpfe, die sich unter den Bildern befinden, bewertet werden. Die Symbole auf den Knöpfen repräsentieren die Bewertungsabstufungen sehr-relevant (++), relevant (+), neutral (0), nicht-relevant (-) und gar-nicht-relevant (--). Die Auswahl einer Beispielregion und die Bewertung mehrerer Regionen eines Bildes wird durch den Regioneneditor ermöglicht (vgl. Abbildung 3.10, rechts). Dieser vergrößert das ausgewählte Bild und zeigt die verfügbaren Bildregionen an. Zusätzlich kann ein Benutzer auch neue Bildregionen einzeichnen und somit der Datenbank hinzufügen. Ist in der initialen Bildübersicht kein Bild enthalten, das der Suchintention eines Anwenders entspricht, dann kann mit dem „Neue Session"-Button eine neue zufällige Bildmenge generiert werden. Nach der Auswahl des initialen Beispielbildes bzw. der initialen Beispielregion sowie der Bewertung eines Suchergebnisses wird der nächste Suchschritt mit dem „Starte Suche"-Button gestartet.

3.3.2 Sprache und sprachliches Referenzieren

Um eine Applikation einer breiten Anwendergruppe zugänglich zu machen, ist es notwendig, einfach zu bedienende Benutzerschnittstellen zu entwerfen. Systeme sollten intuitiv zu bedienen sein, damit auch Anwender ohne spezielle Fachkenntnisse in der Lage sind, sie zu nutzen. Diese Anforderungen an die Benutzerschnittstelle motivieren den Einsatz von natürlichen Modalitäten, die kombiniert die multimodale Mensch-Maschine Interaktion ermöglichen und einen vielversprechenden Ansatz zur einfachen und intuitiven Systembedienung darstellen.

Eine der in INDI integrierten Interaktionsmodalitäten ist die Sprache. Sie ermöglicht die Bedienung der grafischen Oberfläche, sodass auf den Einsatz eines Zeigegerätes, wie z.B. die Maus, fast vollständig verzichtet werden kann.[10] Zentrales Element der sprachverarbeitenden Module des Bildsuchsystems ist der von Fink [Fin99] entwickelte Spracherkenner, der schon in anderen Systemen erfolgreich integriert wurde (vgl. z.B. [Bau02] oder [Löm02]). Seine Hauptidee ist die Kombination von Wissenrepräsentation und schrittweiser Sprachverarbeitung, sodass die Spracherkennung durch linguistisches und domänenspezifisches Wissen beeinflusst wird. Die Grundlage des statistischen Erkenners bilden Hidden-Markov-Modelle. Da in den meisten Systemen die zu erwartenden Äußerungen a priori bekannt sind, bietet es sich an, das statistische Sprachmodell um eine kontextfreie Grammatik für die zu verarbeitende Domäne zu erweitern. Durch die Integration der grammatikalischen Restriktionen während des Erkennungsprozesses kann die Erkennungsleistung im Vergleich zum reinen Spracherkenner gesteigert werden [Wac98].

Da das Bildsuchsystem auch ohne den Spracherkenner funktionsfähig sein soll, läuft dieser als eigenständige Applikation. Die Kommunikation zwischen Erkenner und den

[10] Lediglich zum Einzeichnen einer neuen Bildregion ist die Unterstützung durch ein Zeigegerät, wie z.B. die Maus, notwendig.

Befehl	GUI Aktion
„Wähle Bild 5 als Beispielbild aus“	Das Bild mit der Nummer 5 wird als Beispielbild gekennzeichnet
„Starte die Suche“	Der „Starte Suche“-Knopf wird gedrückt
„Zeige weitere Bilder“	Die Laufleiste scrollt nach unten zur nächsten Bildübersicht
„Die große, gelbe Region ist sehr gut“	Die große, gelbe Region wird im Regioneneditor ausgewählt und der korrespondierende Bewertungsknopf (++) gedrückt

Tabelle 3.1: Beispiele für sprachliche Äußerungen zur Bedienung der grafischen Benutzeroberfläche. Die Oberflächenaktionen der in der linken Spalte formulierten Sprachbefehle sind in der rechten Tabellenspalte dargestellt.

sprachverarbeitenden Modulen des Datenbank-Clients erfolgt durch eine sogenannte *Named Pipe*, die auch als FIFO (*First In First Out*) bezeichnet wird. Diese vom Betriebssystem zur Verfügung gestellte Funktionalität der Interprozesskommunikation ermöglicht den Austausch von Informationen, wobei der eine Prozess lesend und der andere schreibend auf das FIFO zugreift. Die vom Spracherkenner erkannte Wortkette wird in das FIFO eingetragen, wo sie schließlich vom sprachverarbeitenden Clientmodul abgeholt, semantisch analysiert und auf die entsprechenden Aktionen der Benutzeroberfläche abgebildet wird. Dieser Vorgang basiert sowohl auf einem fixen Lexikon als auch auf einer zuvor festgelegten Grammatik. Einen Ausschnitt der im Bildsuchsystem INDI verfügbaren sprachlichen Äußerungen sowie die korrespondierenden Aktionen der Benutzeroberfläche sind in Tabelle 3.1 dargestellt.

Sprachliches Referenzieren

Da das INDI System neben ganzen Bildern auch die Suche nach ähnlichen Bildregionen unterstützt, bietet es sich für eine natürliche Bedienung an, diese auch sprachlich referenzieren zu können. Ein Anwender verwendet dabei gewöhnlich klassische Regionenattribute wie beispielsweise Farbe, Form, Intensität, Größe oder Position. Ein Beispiel für eine derartige Äußerung ist in der letzten Zeile von Tabelle 3.1 gegeben. Die betreffende Region wird dabei durch die Adjektive *groß* und *gelb* referenziert. Zur Verarbeitung der sprachlichen Äußerung und Identifikation der betreffenden Bildregion muss das System Sprachsignal und visuelle Charakteristika der im Bild enthaltenen Regionen miteinander verknüpfen. Die meisten Verfahren (vgl. z.B. [Bro98] oder [Tak98]) basieren dabei auf der Annahme, dass die Sprachsignale fehlerfrei sind und die sprachlichen Beschreibungen eindeutig in visuelle Bildbeschreibungen überführt werden können und umgekehrt. Im Allgemeinen ist dies jedoch nicht der Fall. Die visuelle Bedeutung von Adjektiven wie *groß* oder *gelb* ist in Abhängigkeit von der subjektiven Wahrnehmung eines Anwenders schon an sich unscharf (engl.

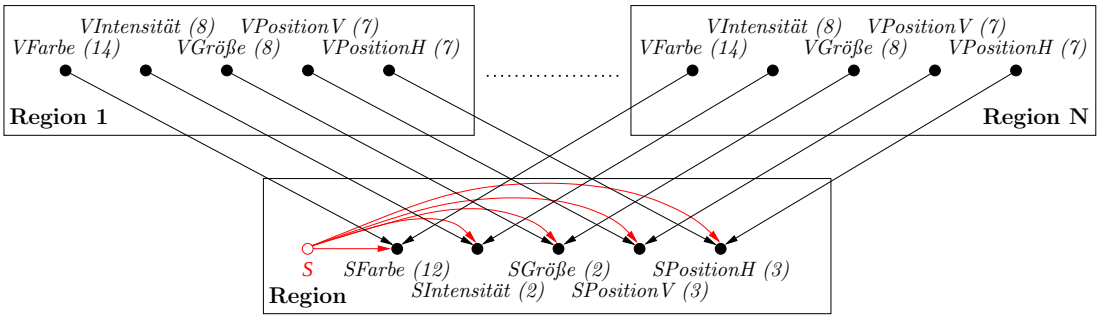
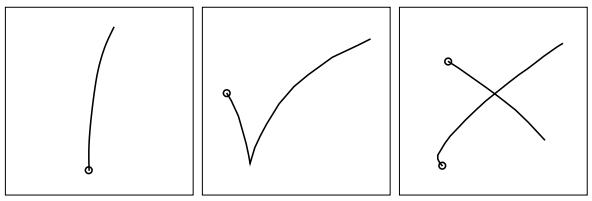


Abb. 3.11: Bayes-Netzwerk zur Beschreibung der Bildregionen. Die Knoten der sprachlichen und visuellen Attribute sind durch die Präfixe *S* und *V* gekennzeichnet. Die Zahlen in den Klammern repräsentieren die Anzahl der Zustände der entsprechenden Zufallsvariable.

fuzzy). Mit welchen Termen ein Anwender Bildinhalte beschreibt, hängt von verschiedenen Einflußfaktoren ab. Neben dem Szenenkontext sind es vor allem Gewohnheiten und der Gemütszustand, die diesen Vorgang beeinflussen. Es bietet sich daher an, die Fusion von Sprache und visuellen Bildinhalten als probabilistischen Prozess zu modellieren.

Ausgangspunkt der Modellierung ist das von Wachsmuth und Sagerer [Wac02] vorgestellte und auf Bayes-Netzwerken basierende Verfahren zur Integration von Sprach- und Bildverstehenskomponenten. Darauf aufbauend können die Beziehungen zwischen sprachlicher und visueller Regionenbeschreibung durch das in Abbildung 3.11 dargestellte Bayes-Netz beschrieben werden. Die Zahlen in den Klammern repräsentieren die Anzahl der Zustände der verschiedenen Zufallsvariablen. *SIntensität* ist beispielsweise ein zweidimensionaler Vektor $(\mathit{hell}, \mathit{dunkel})$. Die Darstellung veranschaulicht eine $1 : N$ Beziehung. Dies bedeutet, dass für eine sprachliche Referenzierung eine Instanz existiert, während ein Bild N Regionen beinhalten kann und daher entsprechend viele Regioneninstanzen vorhanden sind. Welche dieser Regionen vom Anwender referenziert wurde, kann ausgehend von den beobachteten Evidenzen durch Inferenz im Bayes-Netz bestimmt werden. Die dafür notwendigen Übergangstabellen (engl. *Conditional Probability Tables*) der verschiedenen visuellen und sprachlichen Attribute wurden per Hand erstellt. Für den Inferenzvorgang nimmt die Auswahlvariable S eine zentrale Rolle ein. Ihr Zustand $S \in \{1, 2, \ldots, N\}$ ist durch die Anzahl der Regionen bestimmt. Die für den Inferenzvorgang benötigten bedingten Wahrscheinlichkeiten der Sprachknoten sind in Abhängigkeit von der Belegung der Auswahlvariable definiert. So gilt beispielsweise für die bedingte Wahrscheinlichkeit $P(\mathit{SFarbe}|\mathit{VFarbe}_1, \ldots, \mathit{VFarbe}_N, S)$ des Knotens *SFarbe*:

$$P(\mathit{SFarbe}|\mathit{VFarbe}_1, \ldots, \mathit{VFarbe}_N, S) = \begin{cases} P(\mathit{SFarbe}|\mathit{VFarbe}_1), & \text{falls } S = 1 \\ \vdots & \\ P(\mathit{SFarbe}|\mathit{VFarbe}_N), & \text{falls } S = N \end{cases}$$



(a) Vergrößern (b) Auswählen (c) Auswählen

Abb. 3.12: Varianten der Touchscreen-Gesten nach Käster et al. [Käs03a]. Ein auf der Benutzeroberfläche dargestelltes Bild lässt sich durch einen von unten nach oben gezogenen Strich vergrößern (a). Die Auswahl eines Beispielbildes ist sowohl durch einen Haken (b) als auch durch ein Kreuz möglich (c). Für die aus der Bewegung resultierenden Trajektorien werden Merkmale berechnet, die die Klassifikation der ausgeübten Geste ermöglichen.

Die Definition der bedingten Wahrscheinlichkeiten der übrigen Sprachknoten erfolgt analog. Das Ziel des Inferenzvorgangs ist es, die Region zu bestimmen, die auf der Grundlage der gegebenen Evidenzen e die Verbundwahrscheinlichkeit des Netzes maximiert. Dementsprechend definiert die maximale a posteriori Hypothese der Variable S die wahrscheinlichste Beziehung zwischen den verbal erwähnten Regionenattributen und den visuell detektierten Bildregionen:

$$r^* = \underset{r \in \{1,2,\ldots,N\}}{\operatorname{argmax}} P(S = r|e)$$

3.3.3 Touchscreen-Gesten

Ergänzend zur Sprache kann das Bildsuchsystem auch durch Gesten bedient werden. Die Integration dieser Modalität erfolgt unter Zuhilfenahme eines Touchscreens. Dieser Ansatz hat den Vorteil, dass Benutzergesten einfach auf Mausaktionen abgebildet werden können und daher keine externe Applikation notwendig ist, die auf die Erkennung der Gesten spezialisiert ist. Die benötigten Gestenfunktionalitäten werden einfach in den Datenbank-Client integriert.

Das INDI System unterstützt die in Abbildung 3.12 dargestellten Gesten, die sowohl zur Auswahl des initialen Beispielbildes als auch zur Vergrößerung der verkleinerten Bilddarstellungen dienen. Der Benutzer muss zur Auswahl oder Vergrößerung lediglich die entsprechende Geste mit einem Finger auf dem betreffenden Bild ausführen. Dabei wird die Trajektorie der Fingerbewegung aufgezeichnet, die die Grundlage der Gestenklassifikation bildet. Zur Klassifikation der Geste wird ein Polynomklassifikator verwendet. Dieser arbeitet jedoch nicht direkt auf der aufgezeichneten Trajektorie, sondern auf Merkmalen, die aus dem Kantenzug extrahiert werden. Im ersten Schritt wird die zu verarbeitende Trajektorie

durch Translation und Rotation α so normiert, dass der Startpunkt im Koordinatenursprung und der Endpunkt auf der *x*-Achse liegen. Abbildung 3.13 veranschaulicht diesen Prozess am Beispiel einer Auswahlgeste. Die umgebende Box ermöglicht eine Lageuntersuchung bezüglich der *x*-Achse. Dabei wird durch $v_P = y_{\max}/(y_{\max} - y_{\min})$ der prozentuale Anteil der normierten Geste bestimmt, der oberhalb der x-Achse liegt. Als weiteres Merkmal wird die Länge v_L der Trajektorie verwendet. Da diese jedoch größeninvariant sein muss, wird sie durch die Länge der Diagonalen der umgebenen Box normiert. Zusätzlich zu den beschriebenen Charakteristika wird die *x*- und *y*-Koordinate des Gestenschwerpunktes $S = (S_x, S_y)$ des Kantenzuges berechnet. Der resultierende Merkmalsvektor ist schließlich durch

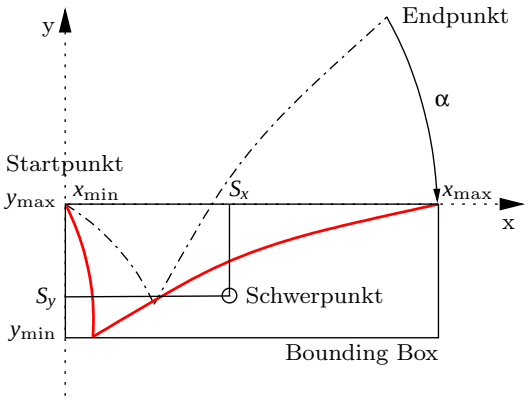


Abb. 3.13: Merkmalsextraktion

$$\mathbf{r}_{\text{Geste}} = (\alpha, v_P, v_L, S_x, S_y)^T$$

definiert und lässt sich durch den Polynomklassifikator klassifizieren. Der Einsatz eines Polynomklassifikators setzt allerdings voraus, dass dieser mit einer klassifizierten Stichprobe trainiert wurde. Zur Generierung der Trainingsmenge wurde ein eigens entwickelter Gesteneditor eingesetzt, auf dem verschiedene Anwender nach Vorlage die verschiedenen Touchscreen-Gesten ausgeführt haben. Mit dem trainierten Klassifikator können zu verarbeitende Gesten klassifiziert und auf die verschiedenen Oberflächenaktionen abgebildet werden. Nicht korrekt ausgeführte Gesten werden auf der Basis eines Schwellwertes detektiert und anschließend zurückgewiesen. Das System verharrt somit im aktuellen Zustand und die Geste muss wiederholt werden.

3.3.4 Kombination von Sprache und Gestik

Ergänzend zum separaten oder abwechselnden Einsatz der verfügbaren Modalitäten Sprache und Gestik unterstützt das Bildsuchsystem INDI auch die Kombination dieser Modalitäten. So lassen sich Bilder und Bildregionen beispielsweise auch durch das Demonstrativpronomen „*dieses*“ wie in „*Wähle dieses Bild als Beispielbild aus*“ referenzieren. Dabei führt der Anwender parallel zur sprachlichen Äußerung eine Zeigegeste aus, mit der er das betreffende Bild eindeutig identifiziert. Für die eingesetzte Touchscreen-Lösung entspricht eine Zeigegeste dem Berühren (engl. *Touching*) des Bildes. Die asynchron eintreffenden Ereignisse (engl. *Events*) der Sprachäußerung und der Gestik sind jeweils mit einem Zeitstempel versehen, der schließlich die Synchronisation der beiden Events und die Abbildung auf die entsprechende Oberflächenak-

tion ermöglicht. Dabei werden die eintreffenden Ereignisse nur dann als zusammengehörend interpretiert, wenn sie im selben Zeitfenster liegen.

3.4 Zusammenfassung

In diesem Kapitel wurde der Aufbau und die Funktionsweise des im Rahmen des LOKI Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“ entwickelten Bildsuchsystems INDI beschrieben. Hauptmerkmale dieser verteilten Anwendung sind die rein inhaltsbasierte, adaptive Bildersuche sowie die natürliche Bedienung durch den Einsatz von Sprache und Touchscreen-Gesten. Zusätzlich ist der modulare Systemaufbau eine wichtige Grundlage für die Verwendung von neu entwickelten Algorithmen zur Bildsegmentierung und Merkmalsextraktion. Ergänzend dazu können durch die Anpassung der Einträge der zentralen Konfigurationstabelle sowohl die für die Bildersuche eingesetzten Repräsentanten als auch die verwendeten Abstandsmaße variiert werden.

Ausgehend von der Formulierung der Systemanforderungen wurden sowohl das Datenmodell als auch die Datenrepräsentationen des Bildsuchsystems vorgestellt. Obwohl mit der Erweiterung des SQL Sprachumfangs die strikte Trennung zwischen Applikations- und Datenschicht aufgehoben wird, ermöglicht die Ausnutzung der speziellen DBMS Funktionalitäten eine effizientere Durchführung der inhaltsbasierten Bildersuche.

Nach der Erläuterung der Datenbankinitialisierung durch Konfiguration und Population wurden die Aufgaben des Bilddatenbank-Servers vorgestellt. Seine Haupteigenschaft ist die Bereitstellung von Funktionalitäten zur Durchführung der inhaltsbasierten Bildersuche. Anschließend wurden die Eigenschaften des Datenbank-Clients, der die eigentliche Benutzerapplikation darstellt, näher erläutert. Seine Aufgaben sind neben der Visualisierung der grafischen Benutzerschnittstelle sowohl die Verarbeitung der GUI Aktionen als auch die Synchronisation der Modalitäten Sprache und Gestik. Die für die Bildersuche notwendigen Daten werden von ihm an den Bilddatenbank-Server gesendet. Umgekehrt fordert er die zur Darstellung der Suchergebnisse notwendige Komponenten vom Server an.

Außerdem wurden die für die natürliche Interaktion notwendigen Modalitäten Sprache und Gestik beschrieben. Da das Bildsuchsystem auch die Verwendung von Bildregionen unterstützt, bietet es sich an, diese auch sprachlich referenzieren zu können. Zur Identifikation der Regionen dienen verschiedene Attribute wie beispielsweise Größe und Farbe, die mit den strukturellen Daten einer Region in der Datenbank gespeichert werden. Die starke subjektive Prägung des Referenzierungsprozesses erfordert allerdings eine probabilistische Verarbeitung. Die Fusion von Sprache und visuellen Bildinhalten erfolgt daher mit Hilfe von Bayes-Netzen. Neben der Sprache dient die

Gestik auf einem Touchscreen zur natürlichen Interaktion mit dem INDI System. Die verfügbaren Gesten ermöglichen sowohl die Auswahl eines Beispielbildes als auch die Vergrößerung eines Elements der Bildübersicht. Darüber hinaus unterstützt das System auch die rein multimodale Interaktion, sodass ein Kommando nur durch die gemeinsame Anwendung von Sprache und Gestik ausgeführt werden kann.

4 Systemlernen durch Mensch-Maschine Interaktion

Nachdem das vorherige Kapitel den Systemaufbau und die einzelnen Funktionalitäten des Bildsuchsystems INDI vorgestellt hat, konzentriert sich das aktuelle Kapitel auf die Erläuterung des Suchschemas, das die zentrale Einheit des Bildsuchsystems darstellt. Aufbauend auf der formalen Beschreibung des Suchprozesses werden sowohl verschiedene Varianten beschrieben, inhaltsbasiert in der gespeicherten Bildsammlung zu suchen, als auch die auf der Mensch-Maschine Interaktion basierenden Methoden der Systemadaption erläutert. Der adaptive Suchprozess basiert auf den Arbeiten von Ishikawa et al. [Ish98] sowie Rui und Huang [Rui98, Rui00], grenzt sich jedoch durch verschiedene Erweiterungen und Verbesserungen deutlich von diesen Ansätzen ab. Des Weiteren wird ein Verfahren zur Kombination von überwachtem und unüberwachtem Lernen vorgestellt. Diesen Ansatz zeichnet aus, dass ergänzend zu den vom Benutzer klassifizierten Bildern auch unklassifizierte Bilder der Datenbank in den Adaptionsprozess integriert werden. Die verschiedenen in INDI verfügbaren Suchansätze und Techniken zur adaptiven Bildersuche werden schließlich ausführlich evaluiert und die Ergebnisse diskutiert.

4.1 Formale Bildbeschreibung

Die Hauptidee der inhaltsbasierten Bildersuche ist die Navigation in einem Bilddatenbestand auf der Grundlage automatisch extrahierter formaler Bildbeschreibungen. Die Qualität eines Suchergebnisses ist eng mit der Qualität der extrahierten Charakteristika verknüpft. Je besser diese die wesentlichen Inhalte eines Bildes beschreiben, desto besser können die relevanten Bilder gruppiert und von den nicht-relevanten Bildern separiert werden. Gewöhnlich wird für ein Bild nicht nur ein Bildrepräsentant erfasst, sondern ein Satz von verschiedenen Repräsentanten. Jeder dieser Deskriptoren beschreibt den Bildinhalt aus einer anderen Sichtweise (vgl. Abschnitt 2.2). Da sie die zentralen Bestandteile der inhaltsbasierten Suche sind, müssen sie permanent verfügbar sein und werden daher in der Datenbank gespeichert. Die Art und Weise wie die verschiedenen Deskriptoren systemintern repräsentiert und davon abhängig gespeichert werden, hat grundlegenden Einfluss auf die Struktur des Suchprozesses und insbesondere auf den Vergleich zweier Bilder. Für die Verwaltung mehrerer Repräsentanten eines Bildes existieren zwei Varianten:

1. Kombinierter Bildrepräsentant: Für jedes Bild wird genau ein Bildrepräsentant gespeichert, der aus der Kombination der einzelnen Bildrepräsentanten resultiert (→ verketteter Merkmalsvektor).

2. Separate Bildrepräsentanten: Bei diesem Ansatz wird auf die Kombination der verschiedenen Charakteristika verzichtet. Stattdessen wird jeder Repräsentant eines Bildes separat in der Datenbank gespeichert.

Die Unterschiede beider Ansätze werden durch die Betrachtung der daraus resultierenden Bildvergleiche deutlich. Während bei der ersten Variante die Ähnlichkeit zweier Bilder direkt aus dem Vergleich der beiden kombinierten Bildrepräsentanten resultiert, erfordert die zweite Variante den schrittweisen Vergleich zweier Bilder. Zuerst werden die korrespondierenden Bildrepräsentanten miteinander verglichen. Anschließend werden die Einzelergebnisse zu einem Gesamtergebnis zusammengefasst. Ein auf diesem Ansatz basierender Vergleichsprozess wird daher im weiteren Verlauf dieser Arbeit als „hierarchischer“ Bildvergleich bezeichnet. Im Gegensatz dazu wird der entsprechende Vergleichsprozess der kombinierten Bildrepräsentanten als „flacher“ Bildvergleich bezeichnet.

Obwohl die erste Variante keinen speziellen Verarbeitungsschritt wie das Zusammenfassen der Einzelergebnisse erfordert, müssen bestimmte Aspekte berücksichtigt und einige Restriktionen akzeptiert werden. Da die Dynamikbereiche der Komponenten unterschiedlicher Merkmalsvektoren variieren können, müssen diese gegebenenfalls vor dem Zusammenfassen normiert werden. Außerdem verbietet dieser Ansatz die Auswahl verschiedener Abstandsmaße sowie eine (dynamisch) unterschiedliche Gewichtung der einzelnen Bildrepräsentanten [Cel99]. Ein weiterer Nachteil ist sicherlich auch, dass der neue Merkmalsraum durch die Kombination einzelner Bildrepräsentanten entsprechend mehr Dimensionen besitzt als die ursprünglichen separaten Vektorräume. Dies ist speziell für den Einsatz von statistischen Lernverfahren problematisch, da sich damit in der Regel auch die Anzahl der zu schätzenden Parameter erhöht. Eine robuste Parameterschätzung kann dann nur durch eine entsprechend umfangreiche Trainingsmenge erzielt werden. Diese ist aber gerade in der inhaltsbasierten Bildersuche nicht gegeben (vgl. Abschnitt 2.4.1).

Im Gegensatz zur kombinierten Repräsentation erweist sich die separate Speicherung der verschiedenen Bildcharakteristika als flexibler. Es können sowohl die verschiedenen Repräsentanten unterschiedlich gewichtet als auch unterschiedliche Abstandsmaße verwendet werden. Da die gespeicherten Bilder allerdings aufgrund der separaten Repräsentation in jedem Merkmalsraum mit den Repräsentanten der jeweiligen Anfrage verglichen werden, existiert für jedes Bild in jedem der verfügbaren Merkmalsräume ein Teilergebnis. Um die Relevanz eines Bildes in Bezug auf die Anfrage zu bestimmen, müssen diese Teilergebnisse zu einem Gesamtergebnis kombiniert werden. Auch wenn der letztgenannte Aspekt einen speziellen Verarbeitungsschritt

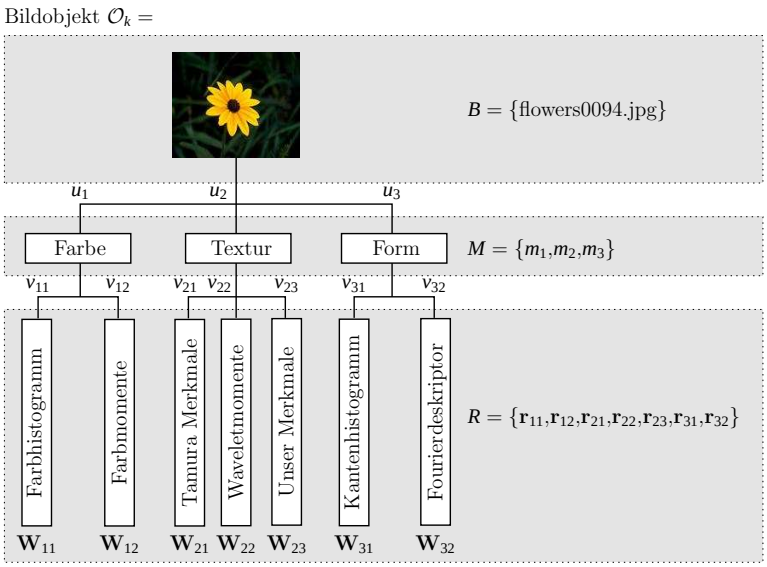


Abb. 4.1: Beispiel eines Bildobjekts, das durch die Merkmale Farbe, Textur und Form sowie die Repräsentanten dieser Merkmalsklassen beschrieben wird. Die Parameter $U = \{u_i\}$, $V = \{v_{ij}\}$ und $W = \{\mathbf{W}_{ij}\}$ ermöglichen eine dynamische Gewichtung der Merkmalsklassen, Repräsentanten und Komponenten der Repräsentanten.

erfordert, ist es gerade die größere Flexibilität im Vergleich zum kombinierten Ansatz, die für die zweite Repräsentationsart spricht. Die in den folgenden Abschnitten beschriebene Modellierung basiert daher auf der separaten Speicherung der verschiedenen Bildcharakteristika.

Um die inhaltsbasierte Bildersuche formal beschreiben zu können, wird analog zu Rui und Huang [Rui98] eine Bildklasse $\mathcal{O}$ eingeführt. Jedes der K zu speichernden Bilder[1] lässt sich demnach durch ein Bildobjekt $\mathcal{O}_k$, $k = 1, 2, \ldots, K$, der Klasse $\mathcal{O} = (B, M, R)$ beschreiben, deren Attribute für ein Beispiel in Abbildung 4.1 dargestellt sind und die wie folgt definiert werden:

- B symbolisiert die digitale Darstellung des Bildes, z.B. ein Bild im JPEG Format.
- $M = \{m_i\}$, mit $i = 1, 2, \ldots, I$, repräsentiert eine Menge von Merkmalen (auch als Merkmalsklassen bezeichnet), wie z.B. Farbe, Textur oder Form.
- R ist die Vereinigungsmenge aller Bildrepräsentanten R_i der verschiedenen Merkmale m_i, z.B. sind $\mathbf{r}_{i1} =$ *Farbhistogramm* oder $\mathbf{r}_{i2} =$ *Farbmomente* Repräsentanten der Merkmalsklasse $m_i =$ *Farbe* (vgl. Abschnitt 2.2.1). Die Vereinigungsmenge R lässt sich demnach wie folgt darstellen:

$$R = R_1 \cup R_2 \cup \cdots \cup R_I, \text{ mit } R_i = \{\mathbf{r}_{ij}\} \text{ für } i = 1, 2, \ldots, I \text{ und } j = 1, 2, \ldots, J_i$$

[1] Um eine übersichtlichere und einfachere Darstellung zu erzielen wird an dieser Stelle auf die zusätzliche Betrachtung von Bildregionen verzichtet. Die formulierte Modellierung ist jedoch ohne Einschränkungen auf Regionen übertragbar.

Dabei symbolisiert J_i die Anzahl der Repräsentanten der Merkmalsklasse m_i. Im weiteren Verlauf der vorliegenden Arbeit werden Bildinhalte ausschließlich durch Merkmalsvektoren beschrieben, sodass $\mathbf{r}_{ij}$ ein Vektor der Dimension N_{ij} ist, $\mathbf{r}_{ij} = (r_{ij_1}, r_{ij_2}, \ldots, r_{ij_{N_{ij}}})^T$. Beispielsweise ist $\mathbf{r}_{ij} = \textit{Farbhistogramm} \in \mathbb{R}^{256}$ die vektorielle Darstellung eines Farbhistogramms, in dem die Häufigkeiten für das Auftreten der 256 Farben kodiert sind.

Die bisher demonstrierte Modellierung bietet zwar eine Grundlage, um den Suchprozess formal zu erfassen, jedoch fehlen ihr Komponenten, die eine Adaption an die Suchintention eines Benutzers ermöglichen. Das bisher starre Modell wird daher um einen Satz von Gewichten $\mathcal{G} = (U, V, W)$ ergänzt. Diese werden abhängig vom Relevance Feedback eines Benutzers gelernt und steuern den Beitrag der verschiedenen Bildobjektkomponenten zum Suchergebnis. $U = \{u_i\}$ bestimmt den Einfluss der unterschiedlichen Merkmalsklassen, $V = \{v_{ij}\}$ gewichtet die verschiedenen Repräsentanten der einzelnen Merkmale und $W = \{\mathbf{W}_{ij}\}$ ermöglicht die Gewichtung der Komponenten r_{ij_n} der Merkmalsvektoren $\mathbf{r}_{ij}$.

4.2 Varianten der Bildersuche

Eine formale Darstellung der inhaltsbasierten Bildersuche erfordert neben der mathematischen Beschreibung der Bildinhalte einen Satz von Abstandsfunktionen, um die Repräsentanten der gespeicherten Bilder mit den aus der Anfrage resultierenden Charakteristika vergleichen zu können (vgl. Abschnitt 2.3). Die Spezifikation eines Suchmodells $\mathcal{S}$ basiert daher neben einer Menge von Bildobjekten $O_\mathcal{S} = \{\mathcal{O}_1, \ldots, \mathcal{O}_K, \mathcal{Q}\}$ und einem Satz von Gewichten $\mathcal{G} = (U, V, W)$ auf einem Satz von Distanzfunktionen $D = \{d_{ij}\}$, sodass für $\mathcal{S}$ gilt:

$$\mathcal{S} = (O_\mathcal{S}, \mathcal{G}, D) = (\{\mathcal{O}_1, \ldots, \mathcal{O}_K, \mathcal{Q}\}, (U, V, W), \{d_{ij}\}),$$

mit $i = 1, 2, \ldots, I$ und $j = 1, 2, \ldots, J_i$. Dabei ist zu beachten, dass das aus dem Query-By-Example Ansatz resultierende Beispielbild ebenfalls auf ein Bildobjekt, dem sogenannten Beispiel- oder auch Anfrageobjekt $\mathcal{Q}$, abgebildet wird. Die auf der Repräsentantenebene korrespondierenden mathematischen Deskriptoren werden durch die Merkmalsvektoren $\mathbf{q}_{ij}$ symbolisiert. Die Relevanz eines Bildobjekts in Bezug auf eine Suchanfrage, die durch das Beispielobjekt repräsentiert wird, lässt sich durch die in Abbildung 4.2 veranschaulichte hierarchische Distanzberechnung ermitteln. Die Grundlage bilden die Distanzen der Merkmalsvektoren der Repräsentantenebene. Diese werden *bottom-up* bzw. hierarchisch auf der Merkmals- und Objektebene zu einem Gesamtdistanzwert zusammengefasst. Je kleiner der Wert ist desto größer ist die Relevanz des korrespondierenden Bildobjekts in Bezug auf die Suchanfrage bzw. desto stärker ähnelt das gespeicherte Bildobjekt dem aktuellen Beispielobjekt.

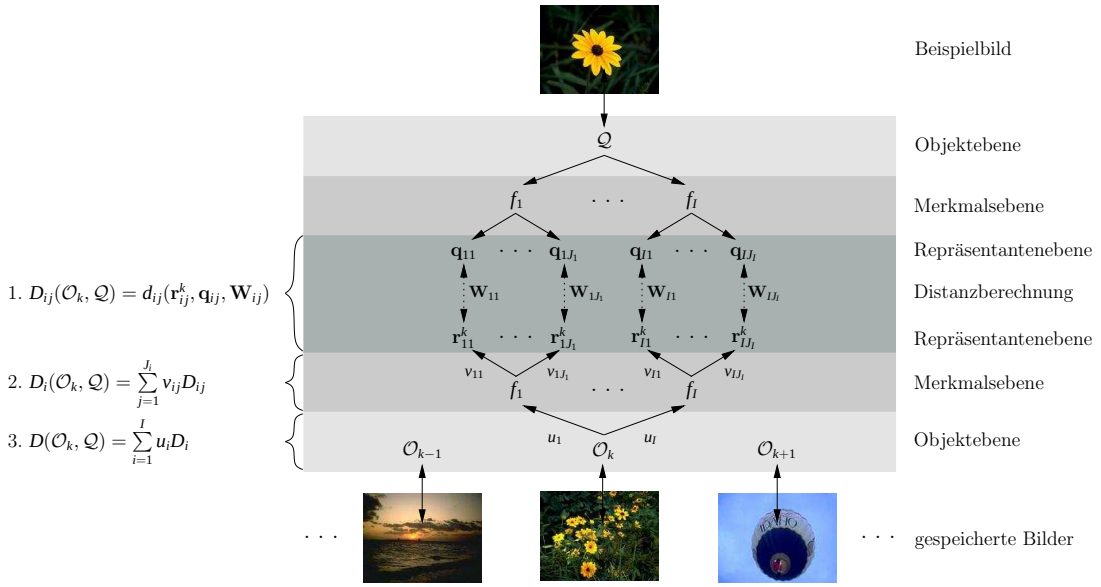


Abb. 4.2: Hierarchische Abstandsberechnung von Bild- und Beispielobjekt (basierend auf [Rui98]). Aufbauend auf der Distanzberechnung der Merkmalsvektoren der Repräsentantenebene werden die Distanzen $D_{ij}(\mathcal{O}_k, \mathcal{Q})$ schrittweise auf der Merkmals- und Objektebene zu einem Gesamtdistanzwert $D(\mathcal{O}_k, \mathcal{Q})$ kombiniert.

In den folgenden Ausführungen werden die verschiedenen Zwischenergebnisse der Repräsentanten- und Merkmalsebene auf der Grundlage einer gewichteten Linearkombination zu einem Gesamtergebnis zusammengefasst.[2] Alternative Varianten zur Kombination der Teilergebnisse, wie z.B. die Beschränkung auf den minimalen oder maximalen Abstandswert eines Bildobjekts, wären ebenfalls möglich, werden aber im Rahmen dieser Arbeit nicht weiter betrachtet.

4.2.1 Distanzbasierte Bildersuche

Nachdem mit der Definition eines Suchmodells eine formale Grundlage geschaffen wurde, werden in den nächsten Abschnitten die Details des Suchprozesses erläutert. Ausgehend von der hierarchischen Distanzberechnung gliedert sich dieser in Abbildung 4.3 skizzierte Prozess in die Phasen Initialisierung, Abstandsbestimmung, Sortierung der Bildobjekte und Generierung der Ergebnisliste.

Bevor die gespeicherten Bildobjekte mit dem Anfrageobjekt verglichen werden können, müssen die Gewichte der verschiedenen Ebenen des Objektmodells initiali-

[2] Dabei wird vorerst, analog zur eingeführten Bildklasse, an der ursprünglichen Einteilung in Repräsentanten-, Merkmals- und Objektebene festgehalten. Wie in Kapitel 4.3 gezeigt wird, kann das Suchmodell ausgehend von der Linearkombination der Zwischenergebnisse vereinfacht werden.

siert werden. An dieser Stelle besteht die Möglichkeit, eine von der Wahl des Beispielobjekts abhängige Gewichtung vorzunehmen. Das dafür notwendige Wissen kann gegebenenfalls aus vorherigen Bildersuchen und einem damit verbundenen Langzeitlernprozess resultieren. Da der Fokus dieser Arbeit allerdings auf dem Kurzzeitlernen und nicht Langzeitlernen liegt, existiert in dem aktuellen Bildsuchsystem keine entsprechende Information, die eine solche individuelle Gewichtung ermöglicht. Deshalb wird eine initial gleichmäßige Gewichtung aller Komponenten verwendet:

$$u_i = \frac{1}{I}, \quad v_{ij} = \frac{1}{J_i} \quad \text{und} \quad \mathbf{W}_{ij} = \mathbf{E} = \begin{bmatrix} 1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & 1 \end{bmatrix} = [\delta_{mn}]_{N_{ij},N_{ij}}, \tag{4.1}$$

Für die verschiedenen Gewichte gelten dementsprechend die folgenden Eigenschaften:

$$\sum_{i=1}^{I} u_i = 1, \quad \sum_{j=1}^{J_i} v_i j = 1 \quad \text{und} \quad \det(\mathbf{W}_{ij}) = 1, \quad \forall i, j$$

Der inhaltsbasierte Suchprozess wird von der Zielsetzung motiviert, die gespeicherten Bilder zu bestimmen, die bei gegebener Suchanfrage die größte Relevanz besitzen. Zu diesem Zweck wird für jedes Bildobjekt $\mathcal{O}_k$ der Abstand zum Beispielobjekt $\mathcal{Q}$ berechnet (vgl. Abbildung 4.2). Ausgangspunkt der hierarchischen Distanzberechnung ist der repräsentantenabhängige Vergleich des Bild- und Beispielobjekts. Demnach ist die Ähnlichkeit der beiden Objekte auf der Grundlage des *ij*-ten Bildrepräsentanten durch

$$D_{ij}(\mathcal{O}_k, \mathcal{Q}) = d_{ij}(\mathbf{r}_{ij}^k, \mathbf{q}_{ij}, \mathbf{W}_{ij}) \tag{4.2}$$

definiert. Dass es sich bei $\mathbf{r}_{ij}^k$ um den *ij*-ten Repräsentanten des Bildobjekts $\mathcal{O}_k$ handelt, wird durch den hochgestellten Index k signalisiert. Im nächsten Schritt werden die verschiedenen Distanzen in Abhängigkeit von der *i*-ten Merkmalsklasse zu einem Distanzwert zusammengefasst:

$$D_i(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J_i} v_{ij}\, D_{ij}(\mathcal{O}_k, \mathcal{Q}) \tag{4.3}$$

Der für die Relevanz eines Bildobjekts $\mathcal{O}_k$ repräsentative Gesamtdistanzwert D resultiert schließlich aus der gewichteten Linearkombination der Distanzwerte der Merkmalsebene:

$$D(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i\, D_i(\mathcal{O}_k, \mathcal{Q}) \tag{4.4}$$

Da für jedes gespeicherte Bildobjekt $\mathcal{O}_k$ ein Distanzwert $D(\mathcal{O}_k, \mathcal{Q})$ existiert, können die K Bildobjekte entsprechend ihrer Relevanz für die Suchanfrage sortiert werden:

$$SL(\mathcal{Q}) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^K\},$$

1. Initialisiere die Gewichte $U = \{u_i\}, V = \{v_{ij}\}$ und $W = \{\mathbf{W}_{ij}\}$
2. Bestimme für jedes gespeicherte Bildobjekt $\mathcal{O}_k$ den Abstand zum gegebenen Beispielobjekt $\mathcal{Q}$ wie folgt:
   a) Berechne in den verschiedenen Merkmalsräumen $\mathbb{R}^{N_{ij}}$ die Distanzen zwischen den Repräsentanten $\mathbf{r}_{ij}^k$ des Bildobjekts $\mathcal{O}_k$ und den Repräsentanten $\mathbf{q}_{ij}$ des Beispielobjekts $\mathcal{Q}$:
   $$D_{ij}(\mathcal{O}_k, \mathcal{Q}) = d_{ij}(\mathbf{r}_{ij}^k, \mathbf{q}_{ij}, \mathbf{W}_{ij})$$
   b) Kombiniere für jedes Merkmal m_i die Ergebnisse der Repräsentantenebene zu einem Abstandswert $D_i(\mathcal{O}_k, \mathcal{Q})$:
   $$D_i(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J_i} v_{ij}\, D_{ij}(\mathcal{O}_k, \mathcal{Q})$$
   c) Bestimme den Gesamtabstand des Bildobjekts O_k zum Anfrageobjekt Q durch Kombination der Ergebnisse der Merkmalsebene:
   $$D(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i\, D_i(\mathcal{O}_k, \mathcal{Q})$$
3. Sortiere die K gespeicherten Bildobjekte entsprechend ihrer Distanzwerte $D(\mathcal{O}_k, \mathcal{Q})$ in aufsteigender Ordnung.
4. Bilde die Ergebnisliste $RL(Q)$ aus den L Bildobjekten, die den geringsten Abstand zu $\mathcal{Q}$ besitzen:
   $$RL(Q) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^L\},$$
   mit $D(\mathcal{O}_k^1, \mathcal{Q}) \leq D(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D(\mathcal{O}_k^L, \mathcal{Q})$ und $L \ll K$.

Abb. 4.3: Schema der distanzbasierten Bildersuche

mit $D(\mathcal{O}_k^1, \mathcal{Q}) \leq D(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D(\mathcal{O}_k^K, \mathcal{Q})$. Die endgültige Ergebnisliste besteht lediglich aus den L Bildobjekten, die den geringsten Abstand zum Beispielobjekt besitzen:

$$RL(Q) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^L\},$$

mit $D(\mathcal{O}_k^1, \mathcal{Q}) \leq D(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D(\mathcal{O}_k^L, \mathcal{Q})$. In der Praxis sollte die Menge der Ergebnisbilder überschaubar sein, sodass ein Anwender einfach in der Ergebnismen-

ge navigieren kann. Dementsprechend ist der Umfang dieser Menge gewöhnlich viel geringer als die Anzahl der gespeicherten Bilder, $L \ll K$. Da das vorgestellte Suchschema ausschließlich auf dem sukzessiven Zusammenfassen der verschiedenen Distanzwerte basiert, wird dieser Ansatz im weiteren Verlauf der Arbeit als „distanzbasierte“ Bildersuche bezeichnet.

4.2.2 Rangbasierte Bildersuche

Bei genauerer Betrachtung der distanzbasierten Bildersuche fällt auf, dass ein wesentlicher Bestandteil dieses Suchschemas die Linearkombination der verschiedenen Distanzen der Repräsentantenebene ist (vgl. Gleichung (4.3)). Dabei ist zu beachten, dass die in den verschiedenen Merkmalsräumen berechneten Abstände zwischen den Repräsentanten der Bildobjekte und des Beispielobjekts eine unterschiedliche Dynamik besitzen können. Ist dies der Fall, so wird die Linearkombination der verschiedenen Distanzwerte von den Merkmalsräumen dominiert, in denen die Abstände eine größere Dynamik besitzen und daher die Distanzen in den anderen Merkmalsräumen überschatten. Um eine derartige und initial ungewollte Dominanz verschiedener Repräsentanten zu verhindern, ist es erforderlich, die verschiedenen Distanzwerte eines Bildobjekts vor deren Kombination zu normieren. [Rui98, Cel99, Zhu00].

Ein Suchansatz, der keine derartige Normierung erfordert, ist die sogenannte „rangbasierte“ Bildersuche, deren einzelnen Schritte in Abbildung 4.4 skizziert sind. Die Grundlage des Schemas bildet ebenso wie bei der distanzbasierten Bildersuche die Initialisierung der Gewichte (vgl. Gleichung (4.1)), sowie die Berechnung der Abstände zwischen den Repräsentanten der Bildobjekte und des Beispielobjekts (vgl. Gleichung (4.2)). Im weiteren Verlauf wird jedoch auf eine hierarchische Berechnung einer Gesamtdistanz $D(\mathcal{O}_k, \mathcal{Q})$ verzichtet. Stattdessen werden die K Bildobjekte entsprechend ihres Distanzwertes $D_{ij}(\mathcal{O}_k, \mathcal{Q})$ für den ij-ten Repräsentanten aufsteigend sortiert:

$$RL_{ij}(\mathcal{Q}) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^K\},$$

mit $D_{ij}(\mathcal{O}_k^1, \mathcal{Q}) \leq D_{ij}(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D_{ij}(\mathcal{O}_k^K, \mathcal{Q})$. Durch die Sortierung lässt sich jedem Bildobjekt $\mathcal{O}_k$ ein Rang $P_{ij}(\mathcal{O}_k, \mathcal{Q})$ in der entsprechenden Liste RL_{ij} zuordnen:

$$\mathcal{O}_k(RL_{ij}) \mapsto P_{ij}(\mathcal{O}_k, \mathcal{Q}) \in [1, 2, \ldots, K]$$

Die Relevanz eines Bildobjekts bei gegebener Suchanfrage wird daher nicht mehr durch einen Distanzwert, sondern durch den korrespondierenden Rang repräsentiert. Analog zur hierarchischen Abstandsberechnung wird der Gesamtrang eines Bildobjekts durch das sukzessive Zusammenfassen der Einzelränge ermittelt. Dazu werden die Ränge der Repräsentantenebene für jede Merkmalsklasse m_i durch eine gewichtete Summation zu einem Zwischenrang P_i kombiniert:

$$P_i(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J_i} v_{ij}\, P_{ij}(\mathcal{O}_k, \mathcal{Q}) \tag{4.5}$$

1. Initialisiere die Gewichte $U = \{u_i\}, V = \{v_{ij}\}$ und $W = \{\mathbf{W}_{ij}\}$
2. Bestimme für jedes gespeicherte Bildobjekt $\mathcal{O}_k$ die Ähnlichkeit zum gegebenen Beispielobjekt $\mathcal{Q}$ wie folgt:
   a) Berechne in den verschiedenen Merkmalsräumen $\mathbb{R}^{N_{ij}}$ die Distanzen zwischen den Repräsentanten $\mathbf{r}_{ij}^k$ des Bildobjekts $\mathcal{O}_k$ und den Repräsentanten $\mathbf{q}_{ij}$ des Beispielobjekts $\mathcal{Q}$:
   $$D_{ij}(\mathcal{O}_k, \mathcal{Q}) = d_{ij}(\mathbf{r}_{ij}^k, \mathbf{q}_{ij}, \mathbf{W}_{ij})$$
   b) Sortiere die Bildobjekte $\mathcal{O}_k$ in den verschiedenen Merkmalsräumen entsprechend ihrer Distanzwerte $D_{ij}(\mathcal{O}_k, \mathcal{Q})$ in aufsteigender Ordnung:
   $$RL_{ij}(\mathcal{Q}) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^K\},$$
   mit $D_{ij}(\mathcal{O}_k^1, \mathcal{Q}) \leq D_{ij}(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D_{ij}(\mathcal{O}_k^K, \mathcal{Q})$
   c) Verwende zur weiteren Verarbeitung statt des berechneten Distanzwertes $D_{ij}(\mathcal{O}_k, \mathcal{Q})$ den Rang $P_{ij}(\mathcal{O}_k, \mathcal{Q}) \in [1, 2, \ldots, K]$, den das Bildobjekt $\mathcal{O}_k$ in der Ergebnisliste $RL_{ij}(\mathcal{Q})$ einnimmt.
   d) Kombiniere für jedes Merkmal m_i die Ergebnisse der Repräsentantenebene zu einem Zwischenrang $P_i(\mathcal{O}_k, \mathcal{Q})$:
   $$P_i(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J_i} v_{ij}\, P_{ij}(\mathcal{O}_k, \mathcal{Q})$$
   e) Bestimme den Gesamtrang $P(\mathcal{O}_k, \mathcal{Q})$ eines Bildobjekts $\mathcal{O}_k$ durch Kombination der Zwischenränge $P_i(\mathcal{O}_k, \mathcal{Q})$:
   $$P(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i\, P_i(\mathcal{O}_k, \mathcal{Q})$$
3. Sortiere die K gespeicherten Bildobjekte entsprechend ihres Ranges $P(\mathcal{O}_k, \mathcal{Q})$ in aufsteigender Ordnung.
4. Bilde die Ergebnisliste $RL(Q)$ aus den L Bildobjekten, die den geringsten Gesamtrang besitzen:
   $$RL(Q) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^L\},$$
   mit $P(\mathcal{O}_k^1, \mathcal{Q}) \leq P(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq P(\mathcal{O}_k^L, \mathcal{Q})$ und $L \ll K$.

Abb. 4.4: Schema der rangbasierten Bildersuche

Der Gesamtrang P eines Bildobjekts $\mathcal{O}_k$ folgt schließlich aus der Linearkombination der verschiedenen Zwischenränge P_i:

$$P(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i \, P_i(\mathcal{O}_k, \mathcal{Q}) \tag{4.6}$$

Analog zum distanzbasierten Verfahren können die K Bildobjekte entsprechend ihres Gesamtranges P sortiert werden. Die daraus resultierende Ergebnisliste RL beinhaltet schließlich die L Objekte, deren Gesamtränge am geringsten sind und die dementsprechend die größte Ähnlichkeit zum Beispielobjekt aufweisen:

$$RL(Q) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^L\},$$

mit $P(\mathcal{O}_k^1, \mathcal{Q}) \leq P(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq P(\mathcal{O}_k^L, \mathcal{Q})$ und $L \ll K$. Die bisher vorgestellten Suchansätze ermöglichen ausgehend von einer initialen Auswahl eines Beispielbildes und der Initialisierung der Gewichte die Suche in der gespeicherten Datenmenge. Die Bildersuche wurde dabei vorerst auf einen Suchschritt beschränkt und verzichtete bisher auf die Integration des aus der Mensch-Maschine Interaktion resultierenden Relevance Feedback. Die Erweiterung dieses einfach strukturierten Suchprozesses zu einem iterativen Prozess und die verschiedenen Methoden des Systemlernens bilden den Schwerpunkt des nächsten Abschnitts.

4.3 Lernen durch Benutzerfeedback

Das Lernen der semantischen Konzepte eines Benutzers während der Bildersuche ist für die Konstruktion leistungsfähiger Bildsuchsysteme ein wichtiger Bestandteil (vgl. Abschnitt 2.4). Das eingeführte Suchmodell $\mathcal{S}$ bietet mit seiner hierarchischen Struktur gleich mehrere Möglichkeiten, das Bildsuchsystem so zu adaptieren, dass eine möglichst gute Anpassung an die Suchintention eines Benutzers erzielt werden kann. Bevor jedoch auf die Einzelheiten der Systemadaption eingegangen wird, kann ausgehend von der Einschränkung auf lineare Funktionen zur Kombination der Zwischenergebnisse die Modellierung ein wenig vereinfacht werden.[3]

Die in den vorherigen Abschnitten beschriebenen Suchschemata beschränken sich bei der Zusammenfassung der verschiedenen Teilergebnisse auf deren Linearkombination.

[3] Auf eine derartige Vereinfachung sollte genau dann verzichtet werden, wenn es gewünscht ist, dass ein Anwender (in der Regel ein „Experte“) explizit die Gewichtung der verwendeten Merkmalsklassen vorgibt. In dieser Arbeit wird allerdings davon ausgegangen, dass diese Gewichtung der Merkmalsklassen innerhalb der iterativen adaptiven Bildersuche implizit mitgelernt wird. Deshalb kann für die folgenden Ausführungen auf eine Unterscheidung von Merkmals- und Objektebene verzichtet werden

Dementsprechend gilt für die Kombination von Gleichung (4.3) und (4.4) sowie Gleichung (4.5) und (4.6):

$$D(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i \sum_{j=1}^{J_i} v_{ij}\, D_{ij}(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} \sum_{j=1}^{J_i} u_i v_{ij}\, D_{ij}(\mathcal{O}_k, \mathcal{Q})$$

und

$$P(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} u_i \sum_{j=1}^{J_i} v_{ij}\, P_{ij}(\mathcal{O}_k, \mathcal{Q}) = \sum_{i=1}^{I} \sum_{j=1}^{J_i} u_i v_{ij}\, P_{ij}(\mathcal{O}_k, \mathcal{Q})$$

Aufgrund der Linearität können die Merkmals- und Objektebene zu einer Ebene zusammengefasst werden. Dadurch vereinfacht sich die Berechnung der Gesamtdistanz D und des Gesamtranges P eines Bildobjekts $\mathcal{O}_k$ zu

$$D(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J} v_j D_j(\mathcal{O}_k, \mathcal{Q}) \tag{4.7}$$

und

$$P(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J} v_j P_j(\mathcal{O}_k, \mathcal{Q}) \tag{4.8}$$

Dabei ist allerdings zu beachten, dass die verschiedenen Merkmalsrepräsentanten j im Gegensatz zur ursprünglichen Modellierung in Abschnitt 4.2 unabhängig von der Merkmalsklasse sind. Die Gesamtanzahl J aller Deskriptoren eines Bildes ergibt sich aus der Summation der merkmalsabhängigen Repräsentantenanzahl J_i, $J = \sum_{i=1}^{I} J_i$. Das Gewicht, mit dem ein Repräsentant j in das Gesamtergebnis eingeht, wird durch v_j symbolisiert. Im Folgenden kann auf eine merkmalsabhängige Indizierung der verschiedenen Merkmalsräume, Repräsentanten und Abstandsfunktionen verzichtet werden. Dementsprechend gilt für den Abstand zwischen einem Bildobjekt $\mathcal{O}_k$ und dem Beispielobjekt $\mathcal{Q}$ im Merkmalsraum $\mathbb{R}^{N_j}$:

$$D_j(\mathcal{O}_k, \mathcal{Q}) = d_j(\mathbf{r}_j^k, \mathbf{q}_j, \mathbf{W}_j) \tag{4.9}$$

Durch die Vereinfachung ist das Suchmodell $\mathcal{S}$ schließlich wie folgt definiert:

$$\begin{aligned} \mathcal{S} = (O_{\mathcal{S}}, \mathcal{G}, D) &= (\{\mathcal{O}_1, \ldots, \mathcal{O}_K, \mathcal{Q}\}, (V, W), \{d_j\}) \\ &= (\{\mathcal{O}_1, \ldots, \mathcal{O}_K, \mathcal{Q}\}, (\{v_j\}, \{\mathbf{W}_j\}), \{d_j\}), \text{ mit } j = 1, \ldots, J \end{aligned}$$

Ausgehend von diesem vereinfachten Suchmodell existieren verschiedene Möglichkeiten, das System an die Suchintention eines Anwenders zu adaptieren. Eine davon ist das Lernen der „idealen“ bzw. „optimalen“ Anfrage. Bei der initialen Auswahl des Beispielbildes handelt es sich um eine erste Anfrageformulierung, die zwar mit der Suchintention eines Benutzers korreliert, aber mit großer Wahrscheinlichkeit noch nicht ideal ist. Eine Annäherung an die ideale Anfrage kann durch die Adaption der

Merkmalsvektoren $\mathbf{q}_j$ des Anfrageobjekts $\mathcal{Q}$ erzielt werden. In der Literatur wird dieser Vorgang als Anfrageverfeinerung (engl. *Query Refinement* [Roc71]) bezeichnet.

Da die semantischen Konzepte eines Benutzers in den ursprünglichen Merkmalsräumen der verschiedenen Bildcharakteristika oft nur vage repräsentiert werden, ist eine Transformation der Bildrepräsentanten erforderlich. In den neuen Merkmalsräumen sollten die Merkmalsvektoren der relevanten Bilder von denen der nicht-relevanten Bilder separiert sein und somit eine bessere Modellierung der Benutzerkonzepte ermöglichen. Eine derartige Transformation kann durch die Adaption der Gewichte $W = \{\mathbf{W}_j\}$ der Repräsentantenebene erzielt werden. Im einfachsten Fall handelt es sich dabei um eine Neugewichtung der Koordinatenachsen des Ursprungsraumes.

Während sich die meisten Systeme aufgrund ihrer kombinierten Bildrepräsentation und des damit verbundenen flachen Bildvergleichs (vgl. Abschnitt 4.1) auf die Adaption der Parameter $\mathbf{q}_j$ und $\mathbf{W}_j$ der Repräsentantenebene beschränken, ermöglicht der hierarchische Ansatz zusätzlich die Adaption der Repräsentantengewichte $V = \{v_j\}$. Dadurch lassen sich die Bildcharakteristika verstärken, die die semantischen Konzepte eines Benutzers am besten beschreiben. Umgekehrt kann das Gewicht derjenigen Charakteristika reduziert werden, die weniger zum Auffinden der relevanten Bilder geeignet sind.

Unter der Voraussetzung, dass eines der vorgestellten Suchschemata (distanz- oder rangbasiert) zur Durchführung eines Suchschritts ausgewählt wurde, kann der adaptive Suchprozess des INDI Systems wie folgt skizziert werden:

1. Initialisierung der Gewichte $V = \{v_j\}$ und $W = \{\mathbf{W}_j\}$.
2. Das System generiert auf der Basis eines initial selektierten Beispielbildes $\mathcal{Q}$ ein erstes Suchergebnis $RL(\mathcal{Q})$.
3. Der Benutzer bewertet die Bilder der Ergebnisliste $RL(\mathcal{Q})$ entsprechend seiner Suchintention entweder als sehr-relevant (++), relevant (+), neutral (0), nicht-relevant (-) oder gar-nicht-relevant (--).
4. Ausgehend von der Benutzerbewertung lernt das System sowohl die Gewichte $V = \{v_j\}$ und $W = \{\mathbf{W}_j\}$ als auch die Repräsentanten $\mathbf{q}_j$ des Beispielobjekts, die sogenannten Anfragevektoren. Mit den gelernten Parametern wird ein weiterer Suchschritt durchgeführt und eine neue Ergebnisliste bestimmt.
5. Falls der Benutzer mit dem neuen Suchergebnis zufrieden ist oder die Suche abbricht, wird der Suchprozess beendet. Ansonsten wird mit Schritt 3 fortgefahren.

Durch die Integration und Verarbeitung des vom Benutzer gegebenen Relevance Feedback wird der Suchvorgang zu einem iterativen Prozess erweitert. Im Bewertungsschritt werden die Bilder der Ergebnisliste entsprechend ihrer Relevanz für die bestehende Suchintention bewertet, sodass für jedes Element $O_l \in RL(\mathcal{Q})$ der Ergebnisliste

eine Bewertung π_l existiert. Dabei muss ein Benutzer nicht alle Ergebnisbilder bewerten. Unbewertete Bilder erhalten das Etikett „neutral“ und haben keinen Einfluss auf den anschließenden Adaptionsschritt. Die durch einen Benutzer gegebenen linguistischen Terme bzw. deren korrespondierende Symbole werden zur weiteren Verarbeitung auf eine numerische Repräsentation abgebildet:

$$\pi_l = \begin{cases} 2 & = & \text{sehr-relevant (++)} \\ 1 & = & \text{relevant (+)} \\ 0 & = & \text{neutral (0)} \\ -1 & = & \text{nicht-relevant (-)} \\ -2 & = & \text{gar-nicht-relevant (--)} \end{cases}$$

Die Einteilung in fünf Abstufungen erfolgt in Analogie zu Rui und Huang [Rui98]. Lediglich die Schrittweite zwischen der sehr-relevanten und relevanten Bewertung sowie zwischen der nicht-relevanten und gar-nicht-relevanten Bewertung wurde verringert. Dies ist dadurch motiviert, dass innerhalb der Bildersuche die exakte Wahl der Bildbewertung („relevant oder sehr-relevant“ bzw. „nicht-relevant oder gar-nicht-relevant“) oftmals nur äußerst schwer zu treffen ist. Deshalb werden diese Bewertungen zwar unterschieden, die Schrittweite von einer Abstufung zur nächsten wird jedoch über alle Bewertungen konstant gehalten. Auch auf eine feinere Abstufung der Bewertungen wurde verzichtet. Sie wäre zwar ebenfalls möglich und für den Adaptionsvorgang eventuell vorteilhaft, würde aber die Interaktion für einen Benutzer komplizierter gestalten. Umgekehrt würde eine Reduzierung auf weniger Bewertungsstufen einen Benutzer gegebenenfalls zu sehr einschränken. Die letztgenannten Hypothesen wurden allerdings nicht weiter untersucht und sollten daher gegebenenfalls in zukünftigen Arbeiten genauer analysiert werden.

4.3.1 Systemadaption durch Distanzminimierung

Eine Anpassung an die Suchintention eines Benutzers erfordert die Adaption der Anfragevektoren $\mathbf{q}_j$, der Komponentengewichte $W = \{\mathbf{W}_j\}$ sowie der Repräsentantengewichte $V = \{v_j\}$. Die Grundlage des in INDI realisierten Adaptionsmechanismus bildet das von Rui und Huang [Rui00, Rui01] vorgestellte Verfahren. Diese Weiterentwicklung des von Ishikawa et al. [Ish98] präsentierten Adaptionsschemas basiert auf der Distanzminimierung. Dabei werden die Parameter des Suchschemas so gelernt, dass die Abstände der Beispielobjekte zum idealen Anfrageobjekt minimiert werden.

Ausgangspunkt eines Lernschritts sind die vom Benutzer gegebenen Bewertungen sowie die korrespondierenden Bilder bzw. deren Repräsentanten in den verschiedenen Merkmalsräumen. In den folgenden Ausführungen wird davon ausgegangen, dass H Bilder der Ergebnisliste $RL(\mathcal{Q})$ als relevant oder sehr-relevant bewertet wurden. Dann repräsentiert $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_H)^T$ die verschiedenen Bewertungen, wobei π_h die Bewertung des Bildobjekts $\mathcal{O}_h \in RL(\mathcal{Q})$ symbolisiert und die Bedingung $\pi_h > 0$ erfüllt.

Die Repräsentanten $\mathbf{r}_j^h$ der als positiv[4] bewerteten Bilder bilden in jedem Merkmalsraum $\mathbb{R}^{N_j}$ eine Trainingsmenge $X_j = \{\mathbf{r}_j^1, \mathbf{r}_j^2, \ldots, \mathbf{r}_j^H\}$. Der Adaptionsschritt basiert demnach ausschließlich auf den positiven Beispielen. Wie in Kapitel 2.4.1 beschrieben wurde, erfordern die negativen Beispiele eine besondere Handhabung, sodass die Fokusierung auf positiv bewertete Bilder sinnvoll erscheint. In Abschnitt 4.3.2 wird allerdings eine Erweiterung des Adaptionsschemas vorgestellt, in der neben den positiven auch negative Elemente zum Systemlernen verwendet werden.

Der Gesamtabstand zwischen einem Stichprobenobjekt $\mathcal{O}_h$ und dem Beispielobjekt $\mathcal{Q}$ ist nach Gleichung (4.7) und (4.9) als

$$D(\mathcal{O}_h, \mathcal{Q}) = \sum_{j=1}^{J} v_j D_j(\mathcal{O}_h, \mathcal{Q}) = \sum_{j=1}^{J} v_j \underbrace{d_j(\mathbf{r}_j^h, \mathbf{q}_j, \mathbf{W}_j)}_{d_j^h} \tag{4.10}$$

definiert. Die vektorielle Schreibweise ermöglicht eine kompaktere Darstellung dieses Gesamtdistanzwertes:

$$f_h = \mathbf{v}^T \mathbf{d}_h \tag{4.11}$$

Dabei bezeichnet f_h den Abstand zwischen dem Bildobjekt $\mathcal{O}_h$ und dem Beispielobjekt $\mathcal{Q}$. $v^T = (v_1, v_2, \ldots, v_J)$ repräsentiert die Gewichte der verschiedenen Repräsentanten. Der Vektor $\mathbf{d}_h = (d_1^h, d_2^h, \ldots, d_J^h)^T$ besteht aus den in den verschiedenen Merkmalsräumen $\mathbb{R}^{N_j}$ berechneten Distanzen der Bildrepräsentanten $\mathbf{r}_j^h$ und der Anfragevektoren $\mathbf{q}_j$. Als Abstandsmaß wird in dieser Arbeit der quadrierte generalisierte euklidische Abstand verwendet. Für den Abstand eines Bildobjekts $\mathcal{O}_h$ und dem Anfrageobjekt $\mathcal{Q}$ gilt daher in Abhängigkeit vom j-ten Bildrepräsentanten:

$$d_j^h = (\mathbf{r}_j^h - \mathbf{q}_j)^T \, \mathbf{W}_j \, (\mathbf{r}_j^h - \mathbf{q}_j)$$

Ziel ist es nun, die Parameter $\mathbf{q}_j$, $W = \{\mathbf{W}_j\}$ und $V = \{v_j\}$ so zu adaptieren, dass die Abstände zwischen dem idealen Anfrageobjekt $\mathcal{Q}^*$ und den positiv bewerteten Bildobjekten $\{\mathcal{O}_h | h = 1, 2, \ldots, H\}$ minimiert werden. Motiviert durch diese Zielsetzung kann das folgende Optimierungsproblem formuliert werden [Rui00, Rui01]:

$$\min(F), \text{ mit } F = \boldsymbol{\pi}^T \, \mathbf{f} = \sum_{h=1}^{H} \pi_h \, f_h$$

Dabei beinhaltet $\mathbf{f} = (f_1, f_2, \ldots, f_H)^T$ die Distanzen der als relevant oder sehr-relevant bewerteten Bilder. Ihre Berechnung basiert auf der in Gleichung (4.11) formulierten

[4] Unter positivem Feedback wird die Bewertung eines Bildes als relevant oder sehr-relevant verstanden. Im Gegensatz dazu wird die Bildbewertung durch die Etiketten nicht-relevant und gar-nicht-relevant als negatives Feedback bezeichnet.

Vorschrift. Wie in Anhang A demonstriert wird, besitzt diese Gleichung ohne die Formulierung gewisser Restriktionen lediglich die triviale Lösung für die Parameter $\{v_j\}$ und $\{\mathbf{W}_j\}$. Deshalb gelten die folgenden Nebenbedingungen:

$$\sum_{j=1}^{J} \frac{1}{v_j} = 1 \quad \text{und} \quad \det(\mathbf{W}_j) = 1$$

Dieses Problem mit Nebenbedingung lässt sich durch das Verfahren der Lagrangesche Multiplikatoren in ein Optimierungsproblem ohne Nebenbedingung überführen:

$$F_\lambda = \boldsymbol{\pi}^T \, \mathbf{f} - \lambda \left(\sum_{j=1}^{J} \frac{1}{v_j} - 1 \right) - \sum_{j=1}^{J} \lambda_j \, (\det(\mathbf{W}_j) - 1)$$

Zur Lösung des Optimierungsproblems gilt es nun, F_λ zu minimieren. Dazu werden die partiellen Ableitungen nach $\mathbf{q}_j$, $w_{j_{mn}}$ und v_j gebildet:

$$\frac{\partial F_\lambda}{\partial \mathbf{q}_j} = 0 \quad , \frac{\partial F_\lambda}{\partial w_{j_{mn}}} = 0 \quad \text{und} \quad \frac{\partial F_\lambda}{\partial v_j} = 0$$

Dabei beschreibt $w_{j_{mn}}$ eine Komponente der Gewichtsmatrix $\mathbf{W}_j = [w_{j_{mn}}]$, mit $m, n = 1, 2, \ldots, N_j$. An dieser Stelle wird auf eine ausführliche Herleitung der Ergebnisse verzichtet. Eine detaillierte Beschreibung der einzelnen Lösungsschritte kann in den Originalarbeiten [Rui00, Rui01] nachgeschlagen werden. Für die optimalen Anfragevektoren $\{\mathbf{q}_j^*\}$, Gewichtsmatrizen $\{\mathbf{W}_j^*\}$ und Gewichte $\{v_j\}$ der Repräsentanten gilt schließlich:

$$\mathbf{q}_j^* = \frac{\sum_{h=1}^{H} \pi_h \mathbf{r}_j^h}{\sum_{h=1}^{H} \pi_h}, \tag{4.12}$$

$$\mathbf{W}_j^* = (\det(\mathbf{C}_j))^{\frac{1}{N_j}} \mathbf{C}_j^{-1}, \tag{4.13}$$

$$v_j^* = \sum_{i=1}^{J} \sqrt{\frac{g_i}{g_j}} \tag{4.14}$$

Dabei repräsentiert $\mathbf{C}_j$ die gewichtete Kovarianzmatrix der Vektoren der Trainingsmenge X_j. Für einen Eintrag $c_{j_{mn}}$ der Matrix $\mathbf{C}_j$ sowie für die Abstände g_j der Merkmalsvektoren der Stichprobe gilt:

$$c_{j_{mn}} = \frac{\sum_{h=1}^{H} \pi_h \, (r_{j_m}^h - q_{j_m}^*) \, (r_{j_n}^h - q_{j_n}^*)}{\sum_{h=1}^{H} \pi_h} \tag{4.15}$$

und

$$g_j = \sum_{h=1}^{H} \pi_h d_j^h \tag{4.16}$$

Lernen der idealen Anfragevektoren

Aus der partiellen Ableitung nach $\mathbf{q}_j$ resultiert, dass der ideale Anfragevektor $\mathbf{q}_j^*$ eines Repräsentanten j durch die gewichtete Summation der entsprechenden Stichprobenvektoren berechnet wird (vgl. Gleichung (4.12)). Wird die unterschiedliche Gewichtung der Trainingsbeispiele vernachlässigt, so entspricht der ideale Anfragevektor dem Mittelwert der Verteilungsdichte, die durch Elemente der Trainingsmenge X_j beschrieben wird.

Adaption der Komponentengewichte

Die optimale Gewichtsmatrix $\mathbf{W}_j^*$ basiert auf der Kovarianzmatrix $\mathbf{C}_j$ der Trainingsbeispiele $\mathbf{r}_j^h \in X_j$, die entsprechend ihrer korrespondierenden Bewertung π_h gewichtet werden. Die Invertierung der Kovarianzmatrix sowie eine anschließende Normierung durch ihre Determinante und der Dimension des jeweiligen Merkmalsraumes ergeben schließlich die Gewichtsmatrix $\mathbf{W}_j^*$ (vgl. Gleichung (4.13)). Eine genauere Betrachtung des daraus resultierenden Abstandsmaßes ermöglicht eine anschaulichere Interpretation dieses Adaptionsschritts.

Da $\mathbf{W}_j^*$ eine nicht singuläre symmetrische Matrix ist, lässt sie sich durch eine geeignete orthonormale Transformation $\mathbf{T} = \mathbf{\Phi}_j^T$ in Diagonalform bringen (vgl. z.B [Fuk90, S. 29ff] oder [Fin03, S. 144]). Dabei repräsentiert $\mathbf{\Phi}_j$ die sogenannte Eigenvektormatrix, deren Spalten den Eigenvektoren ϕ_{jn} der Gewichtsmatrix $\mathbf{W}_j^*$ entsprechen:

$$\mathbf{\Phi}_j = [\phi_{j1}, \phi_{j2}, \ldots, \phi_{jN_j}]$$

Aus der Verallgemeinerung der Eigenwertgleichung folgt, dass $\mathbf{W}_j^*$ in die Eigenvektor- und Eigenwertmatrix zerlegt werden kann:

$$\mathbf{W}_j^* \mathbf{\Phi}_j = \mathbf{\Phi}_j \mathbf{\Lambda}_j \Leftrightarrow \mathbf{W}_j^* = \mathbf{\Phi}_j \mathbf{\Lambda}_j \mathbf{\Phi}_j^T,$$

wobei $\mathbf{\Lambda}_j$ eine Diagonalmatrix darstellt, deren Diagonalelemente die Eigenwerte von $\mathbf{W}_j^*$ sind. Für den Abstand des Repräsentanten $\mathbf{r}_j^k$ eines Bildobjekts $\mathcal{O}_k$ zum Anfragevektor $\mathbf{q}_j^*$ des idealen Beispielobjekts $\mathcal{Q}^*$ gilt dementsprechend im nächsten Iterationsschritt des Suchprozesses:

$$\begin{aligned} d_j^k &= (\mathbf{r}_j^k - \mathbf{q}_j^*)^T \, \mathbf{W}_j^* \, (\mathbf{r}_j^k - \mathbf{q}_j^*) && (4.17) \\ &= (\mathbf{r}_j^k - \mathbf{q}_j^*)^T \, \mathbf{\Phi}_j \, \mathbf{\Lambda}_j \, \mathbf{\Phi}_j^T \, (\mathbf{r}_j^k - \mathbf{q}_j^*) && (4.18) \\ &= (\mathbf{\Phi}_j^T (\mathbf{r}_j^k - \mathbf{q}_j^*))^T \, \mathbf{\Lambda}_j \, (\mathbf{\Phi}_j^T (\mathbf{r}_j^k - \mathbf{q}_j^*)) && (4.19) \end{aligned}$$

Die Umformungen demonstrieren, dass die Adaption der Gewichtsmatrix $\mathbf{W}_j$ eine Transformation in einen neuen Merkmalsraum bewirkt, dessen Koordinatenachsen durch die Komponenten der Diagonalmatrix $\mathbf{\Lambda}_j$ neu gewichtet werden.

Unter der Annahme, dass die für eine Suche relevanten Bilder in den jeweiligen Merkmalsräumen Häufungsgebiete bilden, wird an dieser Stelle deutlich, dass die Adaption der Anfragevektoren und der Gewichtsmatrix im wesentlichen der Schätzung einer Verteilungsdichte entspricht. Diese modelliert in den jeweiligen Merkmalsräumen die Häufungsgebiete der relevanten Bilder. Die Abstandsberechnung auf der Grundlage des quadratischen generalisierten euklidischen Abstands entspricht daher der Berechnung eines Dichtewertes:

$$p_j(\mathbf{r}_j^k) = \mathcal{N}_{\mathbf{r}_j^k}(\mathbf{q}_j^*, \mathbf{W}_j) = \frac{1}{\sqrt{(2\pi)^{N_j} \det(\mathbf{W}_j^{-1})}} \, e^{-\frac{1}{2}(\mathbf{r}_j^k - \mathbf{q}_j^*)^T \mathbf{W}_j (\mathbf{r}_j^k - \mathbf{q}_j^*)}$$

Allerdings ist dabei zu beachten, dass sich die Abstands- und Dichtewerte reziprok zueinander verhalten, d.h. ein kleiner Abstandswert impliziert einen großen Dichtewert und umgekehrt.

Beschränkung auf diagonale Kovarianzen

Bisher erfolgte ein rein theoretische Betrachtung der optimalen Gewichtsmatrix. Um dieses Verfahren jedoch auch in der Praxis einsetzen zu können, müssen einige statistische und daraus resultierend numerische Aspekte berücksichtigt werden. Die Berechnung der optimalen Gewichtsmatrix erfordert die Invertierung der Kovarianzmatrix (vgl. Gleichung (4.13)). Dieses ist jedoch gerade bei dem oftmals geringen Umfang H der verfügbaren Stichprobe X_j problematisch. Als Konsequenz dieses Datenmangelproblems (engl. *Sparse Data Problem*, vgl. [Fin03, S. 137]) kann es passieren, dass die Matrix singulär ist und daher keine Lösung für $\mathbf{W}_j^*$ existiert. Für die Fortsetzung des Suchprozesses ist die Berechnung der Gewichtsmatrix allerdings essentiell. Daher werden im INDI System in singulären Fällen die Komponenten r_{j_n} eines Bildrepräsentanten als statistisch unabhängig angenommen. Anstatt einer kompletten wird somit eine diagonale Kovarianzmatrix[5] $C_j = [c_{j_{mn}}]$, mit $m, n = 1, 2, \ldots, N_j$, berechnet, deren Komponenten wie folgt definiert sind:

$$c_{j_{mn}} = \begin{cases} \frac{\sum_{h=1}^{H} \pi_h \, (r_{j_m}^h - q_{j_m}^*) \, (r_{j_n}^h - q_{j_n}^*)}{\sum_{h=1}^{H} \pi_h}, & \text{falls } m = n \\ 0 & , \text{sonst} \end{cases}$$

Da die diagonale Kovarianzmatrix in der Regel invertierbar ist und ihre Determinante bestimmt werden kann, lässt sich die optimale Gewichtsmatrix nach Gleichung 4.13

[5] Diese Strategie entspricht im Wesentlichen dem von Rui und Huang [Rui98] vorgeschlagenen heuristischen Ansatz zur Adaption der Gewichtsvektoren. Dabei werden die Komponenten der Merkmalsvektoren, die innerhalb der Trainingsmenge am geringsten streuen, am stärksten gewichtet. In der Mustererkennung stellt die Beschränkung auf diagonale Kovarianzen zur Dichteschätzung allerdings eine gängige Variante dar. Dieser Ansatz wird gewöhnlich immer dann zur Lösung einer Klassifikationsaufgabe angewandt, wenn davon auszugehen ist, dass die zu verarbeitende Trainingsmenge keine robuste Schätzung einer kompletten Kovarianzmatrix zulässt.

berechnen. Der quadrierte generalisierte euklidische Abstand reduziert sich demnach zu einem gewichteten Abstandsmaß:

$$\begin{aligned} d_j &= (\mathbf{r}_j - \mathbf{q}_j^*)^T \, \mathbf{W}_j^* \, (\mathbf{r}_j - \mathbf{q}_j^*) \\ &= (\mathbf{r}_j - \mathbf{q}_j^*)^T \, \left(\prod_{n=1}^{N_j} c_{j_{nn}} \right)^{\frac{1}{N_j}} \, \mathbf{C}_j^{-1} \, (\mathbf{r}_j - \mathbf{q}_j^*) \end{aligned}$$

Anschaulich bedeutet der Adaptionsschritt, dass im nächsten Suchschritt die Komponenten am stärksten gewichtet werden, die innerhalb der Trainingsmenge am wenigsten streuen.

Regularisierung

Alternativ zur Verwendung der diagonalen Kovarianzmatrix kann im INDI System die von Friedman [Fri89] vorgestellte Regularisierung eingesetzt werden. Die Hauptidee dieses Verfahrens ist die Manipulation der Elemente und insbesondere der Diagonalelemente der zu invertierenden Matrix. Dabei werden kleine Werte zu den Diagonaleinträgen der Matrix addiert, sodass sie invertierbar ist:

$$\mathbf{C}_j' = (1 - \gamma)\mathbf{C}_j + \frac{\gamma}{N_j} \mathrm{tr}[\mathbf{C}_j]\mathbf{E}$$

Die Stärke der Regularisierung wird mit dem Parameter $\gamma \in [0, 1]$ kontrolliert und sollte abhängig von der Menge der verfügbaren Stichprobenelemente gewählt werden. Je geringer das Verhältnis der vorhandenen Stichprobenelemente zu den benötigten ist, desto größer sollte γ sein. Allerdings existiert für die Anzahl der benötigten Trainingsbeipiele keine Faustregel, sodass oftmals empirische Untersuchungen die Grundlage für eine derartige Abschätzung bilden. Alternativ bietet sich eine inkrementelle Regularisierung an, bei der solange mit einem langsam anwachsenden γ an der Matrix „gerüttelt“ wird, bis entweder ein Maximalwert[6] $\gamma_{\max}$ überschritten wird oder die Kovarianzmatrix invertierbar ist. Im Wesentlichen entspricht die Regularisierung nach Friedman einem Verrauschen der Trainingsmenge. Anstatt jedoch die Stichprobenelemente zu verrauschen, wird dabei die Kovarianzmatrix manipuliert, die direkt aus der ursprünglichen Stichprobe berechnet wird.

Lernen der Repräsentantengewichte

Der Einfluss eines Repräsentanten *j* im nächsten Iterationsschritt der Bildersuche wird durch die in Gleichung (4.14) formulierte Berechnungsvorschrift bestimmt. Dabei

[6] Die Definition einer maximalen Korrekturschwelle ist deshalb sinnvoll, um die ursprüngliche Matrix nicht zu stark zu verrauschen. Sollte der Fall eintreten, dass der Maximalwert überschritten wird, wird anstatt einer kompletten Kovarianzmatrix die diagonale Kovarianzmatrix zur weiteren Verarbeitung verwendet.

werden in jedem Merkmalsraum $\mathbb{R}^{N_j}$ die Distanzen d_j^h der Trainingsvektoren $\mathbf{r}_j^h$ zu dem jeweiligen optimalen Anfragevektor $\mathbf{q}_j^*$ berechnet und zu einem Gesamtdistanzwert g_j aufsummiert. Schließlich wird der Bildrepräsentant j am stärksten gewichtet, in dessen Merkmalsraum die Trainingsbeispiele am geringsten um den optimalen Anfragevektor streuen. Dies bedeutet, dass die positiven Bilder in dem entsprechenden Merkmalsraum am kompaktesten repräsentiert werden und der Gesamtdistanzwert g_j im Verhältnis zu den übrigen Distanzwerten g_i, mit $i = 1, \ldots, J$ und $i \neq j$, am kleinsten ist. Da dieser Ansatz auf dem Vergleich der Distanzen g_j der Trainingsbeispiele in den unterschiedlichen Merkmalsräumen basiert, erfordert er allerdings, dass die Abstandswerte in den verschiedenen Merkmalsräumen vergleichbar sind. Sind sie es nicht, so wird der Adaptionsschritt von dem Merkmalsraum dominiert, in dem die Merkmalsvektoren ohnehin eine geringe Dynamik besitzen.

Alternativ zu der in Gleichung 4.14 formulierten Berechnungsvorschrift der optimalen Repräsentantengewichte wurde für das INDI System eine rangbasierte Variante dieses Adaptionsschritts entwickelt. Diese verwendet anstelle der Distanz eines Trainingselements $\mathcal{O}_h$ den entsprechenden Rang $P_j(\mathcal{O}_h, \mathcal{Q})$, den das Bildobjekt in der aktuellen sortierten Liste $RL_j(\mathcal{Q})$ des Merkmalsraumes $\mathbb{R}^{N_j}$ einnimmt (vgl. Abschnitt 4.2.2):

$$RL_j(\mathcal{Q}) = \{\mathcal{O}_k^1, \mathcal{O}_k^2, \ldots, \mathcal{O}_k^K\},$$

mit $D_j(\mathcal{O}_k^1, \mathcal{Q}) \leq D_j(\mathcal{O}_k^2, \mathcal{Q}) \leq \cdots \leq D_j(\mathcal{O}_k^K, \mathcal{Q})$. Der Ansatz wird demnach wie folgt modifiziert:

$$v_j^* = \sum_{i=1}^{J} \sqrt{\frac{g_i}{g_j}}, \quad \text{mit} \quad g_j = \sum_{h=1}^{H} \pi_h P_j(\mathcal{O}_h, \mathcal{Q}), \quad \text{und} \quad j = 1, 2, \ldots, J \tag{4.20}$$

Komplettiert wird die Adaption der Repräsentantengewichte durch die Normierung der verschiedenen Quantitäten v_j^*, sodass sie sich zu Eins summieren:

$$v_j^* = \frac{v_j^*}{v_{\text{total}}}, \quad \text{mit} \quad v_{\text{total}} = \sum_{j=1}^{J} v_j^* \tag{4.21}$$

4.3.2 Systemlernen mit negativen Beispielen

Das bisher betrachtete Adaptionsschema berücksichtigt ausschließlich positiv bewertete Bilder. Von einem Benutzer gegebenes negatives Feedback wird nicht in den Adaptionsprozess integriert. Obwohl die inhaltsbasierte Bildersuche von der Zielsetzung motiviert ist, bei gegebener Suchanfrage die Bilder mit der größten Relevanz zu finden, können auch als nicht-relevant bewertete Bilder zum Systemlernen beitragen (vgl. z.B. [Laa00], [Zho01] oder [Khe02]).

Ergänzend zu der im vorherigen Abschnitt vorgestellten Adaptionsmethode der Repräsentantengewichte wurde ein bestehendes heuristisches Verfahren [Rui98]

zum Lernen der idealen Repräsentantengewichte weiterentwickelt, das neben positivem auch negatives Feedback berücksichtigt. Die Grundlage dieses Verfahrens bilden die Teilergebnislisten $RL_j(\mathcal{Q})$, eine Menge von H bewerteten Bildobjekten $\{\mathcal{O}_h | h = 1, 2, \ldots, H\}$ sowie deren korrespondierenden Bewertungen $\pi_h \in [-2, -1, 1, 2]$. Die Berechnungsvorschrift dieses Adaptionsschritts, der auf den Gewichten v_j des vorherigen Suchschritts basiert, lautet:

$$v_j^* = v_j + \alpha \sum_{h=1}^{H} \pi_h \, \Psi(P_j(\mathcal{O}_h, \mathcal{Q})) \tag{4.22}$$

Ein Lernschritt der Adaptionsmethode wird durch die Lernrate $\alpha \in [0, 1]$ begrenzt. $P_j(\mathcal{O}_h, \mathcal{Q})$ repräsentiert den Rang, den das Bildobjekt $\mathcal{O}_h$ in der Teilergebnisliste $RL_j(\mathcal{Q})$ einnimmt und $\Psi : \mathbb{Z}^+ \mapsto \mathbb{R}_0^+$ berechnet das Gewicht, mit dem es aufgrund seines Ranges in den Adaptionsschritt eingeht. Ob dabei das Gewicht des betrachteten Repräsentanten verstärkt oder geschwächt wird, hängt von dem Vorzeichen der entsprechenden Bildbewertung ab. In INDI wird die in Abbildung 4.5 dargestellte sigmoide Funktion als Gewichtsfunktion verwendet, die in Abhängigkeit von der Anzahl der Ergebnisbilder L definiert ist. Je niedriger der Rang eines Bildobjekts in der Ergebnisliste $RL_j(\mathcal{Q})$ ist, desto größer ist der entsprechende Gewichtswert. Anschaulich bedeutet der Adaptionsschritt, dass die Bildrepräsentanten, in deren Merkmalsräumen die als positiv bewerteten Bilder einen niedrigen Rang[7] einnehmen, verstärkt werden. Besitzen negativ bewertete Bilder einen geringen Rang in einer Teilergebnisliste, so wird das Gewicht des entsprechenden Repräsentanten reduziert. Somit lassen sich die Bildcharakteristika bestimmen, die die Suchintention eines Benutzers am besten repräsentieren.

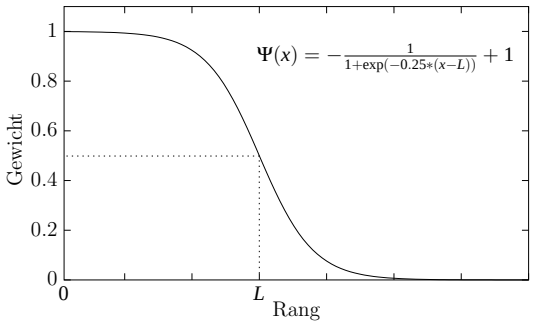


Abb. 4.5: Sigmoide Gewichtsfunktion

Eine Eigenschaft der in Gleichung 4.22 formulierten Berechnungsvorschrift ist, dass ein Repräsentantengewicht v_j^* durchaus negativ werden kann. Ein negatives Gewicht signalisiert, dass der korrespondierende Repräsentant für die aktuelle Suche keine Relevanz besitzt. Deshalb wird dieser für den folgenden Suchschritt deaktiviert, indem der entsprechende Gewichtswert auf Null gesetzt wird, $v_j^* = 0$. Abschließend werden die Repräsentanten so normiert, dass sie sich zu Eins summieren (vgl. Gleichung 4.21). Sollten alle Repräsentantengewichte negativ sein, wird der folgende Suchschritt einfach mit der in Abschnitt 4.2.1 beschriebenen initialen Gewichtung fortgesetzt.

[7] Ein niedriger Rang in einem Merkmalsraum ist gleichbedeutend mit einem geringen Abstand zum jeweiligen Anfragevektor.

4.3.3 Kombination von überwachtem und unüberwachtem Lernen

Das wohl größte Manko eines adaptiven Bildsuchsystems ist der sehr geringe Umfang der klassifizierten Stichprobe. Ausgehend von einer derartig kleinen Trainingsmenge ist eine robuste Parameterschätzung und eine gute Generalisierung nur schwer zu erzielen. Dieses Problem ist allerdings nicht auf den Bereich der inhaltsbasierten Bildersuche beschränkt. Vielmehr existiert diese Problematik in vielen Anwendungsszenarien, in denen eine große Menge von Daten durch einen Klassifikator klassifiziert wird, der auf der Grundlage einiger weniger Stichprobenelemente trainiert wurde. Die Ursache für eine solche unzureichende Trainingsmenge liegt oftmals in den Kosten für das Akquirieren der Beispielelemente. Gewöhnlich ist der Vorgang der Stichprobengenerierung sehr aufwendig und in vielen Bereichen auch nur durch anwendungsspezifisches Expertenwissen zu realisieren. Speziell in der inhaltsbasierten Bildersuche ist die geringe Trainingsmenge eine logische Konsequenz der nur begrenzt zumutbaren Interaktion zwischen Anwender und System (vgl. Kapitel 2.4.1). Ideal wäre daher ein Lernansatz, der ausgehend von einer klassifizierten Trainingsmenge automatisch weitere Trainingsbeispiele akquiriert und in den Lernprozess integriert. Verschiedene Ansätze des maschinellen Lernens verfolgen eine Strategie, bei der während des Trainings neben den klassifizierten auch unklassifizierte Elemente berücksichtigt werden [Blu98, Gol00, See01]. Wu et al. [Wu00] sowie Qian et al. [Qia02] demonstrieren außerdem, dass unklassifizierte Bilder auch in der inhaltsbasierten Bildersuche verwendet werden können, um den Lernprozess zu unterstützen. Die Ergebnisse der experimentellen Untersuchungen dieser Arbeiten belegen, dass dieser Ansatz vielversprechend ist. Motiviert durch die Erfolge der erwähnten Arbeiten wurde deshalb ein Verfahren entwickelt, das zum Lernen neben den klassifizierten auch unklassifizierte Bilder berücksichtigt. Wie dies im Detail erfolgt, wird in den kommenden Abschnitten näher erläutert.

Bei dem bisher beschriebenen Lernverfahren handelt es sich um überwachtes Lernen. Der Benutzer bewertet eine Menge von Bildern entweder als relevant oder nichtrelevant. Ausgehend von dieser Trainingsmenge adaptiert das System seine Parameter und generiert eine neue Ergebnismenge. Rein formal ist die Parameteradaption nichts weiter als das Training eines Klassifikators. Dieser erzeugt im folgenden Suchschritt für jedes Bild der Datenbank ein „weiches“ Klassifikationsergebnis, das ein Maß für die Relevanz eines Bildes bei gegebener Anfrage darstellt. Wie bereits gezeigt, ist der Lernschritt auf der Repräsentantenebene von der Menge und der Qualität der als relevant oder sehr-relevant klassifizierten Beispielvektoren abhängig. Sind diese für die zu beschreibende Datenverteilung wenig repräsentativ, so können die Parameter der Kovarianzmatrix nicht robust geschätzt werden. Insbesondere kann dies zur Folge haben, dass die Kovarianzmatrix nicht invertierbar ist. In diesem Fall wird auf die Berechnung der kompletten Kovarianzmatrix verzichtet und stattdessen die diagonale Matrix verwendet (vgl. Abschnitt 4.3.1). Dies ist gleichbedeutend mit der Beschränkung auf

einen „schlechteren“ Klassifikator[8], da lediglich eine Neugewichtung der Koordinatenachsen des Ursprungsraumes erzielt wird und eine komplexere Transformation der Daten, wie sie in Gleichung 4.17 bis Gleichung 4.19 demonstriert wurde, nicht möglich ist. Es ist daher anzustreben, die Unzulänglichkeiten der klassifizierten Stichprobe auszugleichen, indem die Trainingsmenge durch weitere Beispielelemente ergänzt wird, sodass eine robustere Parameterschätzung möglich ist.[9]

Eine gängige Methode für die Erweiterung der Trainingsmenge und das Erzielen einer besseren Generalisierungsfähigkeit des Klassifikators ist das Verrauschen der Beispielvektoren. Mit der Regularisierung nach Friedman [Fri89] wurde in Abschnitt 4.3.1 bereits ein Verfahren vorgestellt, das genau dies macht. Dabei werden allerdings nicht direkt die Trainingsbeispiele manipuliert, sondern speziell die Diagonalelemente der entsprechenden Kovarianzmatrix. Alternativ zu dieser Variante bietet es sich an, aus den unklassifizierten Elementen, die zu den Beispielvektoren ähnlichsten Repräsentanten zu bestimmen und sie der ursprünglichen Trainingsmenge hinzuzufügen. Motiviert durch diese Zielsetzung ist die Hauptidee des entwickelten Verfahrens die Kombination von überwachtem und unüberwachtem Lernen. Dabei werden ausgehend von einer klassifizierten Trainingsmenge in einem Co-Training[10] Schritt zunächst weitere Trainingsbeispiele in der unklassifizierten Datenmenge bestimmt und der aktuellen Stichprobe hinzugefügt. Aufbauend auf dieser erweiterten Trainingsmenge wird anschließend ein neuer Klassifikator trainiert, der schließlich zur Generierung eines neuen Suchergebnisses dient.

Den Ausgangspunkt des in Abbildung 4.6 skizzierten Algorithmus bilden sowohl eine Menge von positiven Beispielvektoren $X^p = \{\mathbf{r}_1^p, \mathbf{r}_2^p, \ldots, \mathbf{r}_H^p\}$ als auch eine Menge von negativen Mustern $X^n = \{\mathbf{r}_1^n, \mathbf{r}_2^n, \ldots, \mathbf{r}_I^n\}$. Falls keine negativen Bewertungen gegeben wurden, werden einfach zufällig I Merkmalsvektoren der unbewerteten Bilder bestimmt und als negative Beispiele interpretiert. Diese Vorgehensweise erscheint deshalb gerechtfertigt, da anzunehmen ist, dass für eine gegebene Anfrage wesentlich mehr nicht-relevante als relevante Bilder in der Datenbank existieren. Dementsprechend ist die Wahrscheinlichkeit, den Bildrepräsentanten eines nicht-relevanten Bildes zu erhalten, viel größer als die Wahrscheinlichkeit, einen Deskriptor zu erhalten, der den Inhalt eines relevanten Bildes beschreibt. Aufbauend auf der klassifizierten Stichprobe gilt es nun im unüberwachten Klassifikationsschritt, die Merkmalsvektoren der unklassifizierten Bilder zu bestimmen, die den positiven Bildrepräsentanten ähneln. Dies geschieht auf der Grundlage eines Nachbarschaftsklassifikators. Dabei

[8] Diese Hypothese wird wie wir später sehen werden von den experimentellen Untersuchungen belegt.

[9] Inwieweit wirklich eine robustere Parameterschätzung für die zu beschreibende Datenverteilung erzielt wird, bleibt vorerst abzuwarten und muss experimentell untersucht werden. Im ungünstigsten Fall wird die Trainingsmenge durch die neuen Beispielelemente zu stark verrauscht, sodass die Stichprobenerweiterung kontraproduktiv ist und der Klassifikator eher schlechtere statt bessere Ergebnisse liefert.

[10] Co-Training meint hier, dass der überwachte Lernschritt durch einen unüberwachten Verarbeitungsschritt unterstützt wird. Dabei liefert der unüberwachte Klassifikationsschritt neue Trainingsbeispiele für einen erneuten überwachten Lernschritt. Der unüberwachte Klassifikator fungiert somit als Co-Trainer.

Gegeben sei eine Menge von positiven Merkmalsvektoren $X^p = \{\mathbf{r}_1^p, \mathbf{r}_2^p, \ldots, \mathbf{r}_H^p\}$ und eine Menge von negativen Bildbeschreibungen $X^n = \{\mathbf{r}_1^n, \mathbf{r}_2^n, \ldots, \mathbf{r}_I^n\}$.

Bestimme ausgehend von X^p und den korrespondierenden Bildbewertungen $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_H)^T$ gemäß Gleichung 4.13 die optimale Gewichtsmatrix $\mathbf{W}^*$. (a)

FOR alle $\mathbf{r}_h^p \in X^p$

Bestimme den maximalen Abstand $d_{\max}$ zu den Vektoren von X^p und den minimalen Abstand $d_{\min}$ zu den Elementen von X^n:

$$\left.\begin{array}{l} d_{\max} = \max\limits_{\mathbf{r}_k^p \in X^p} d(\mathbf{r}_h^p, \mathbf{r}_k^p) \\ d_{\min} = \min\limits_{\mathbf{r}_k^n \in X^n} d(\mathbf{r}_h^p, \mathbf{r}_k^n) \end{array}\right\} \text{mit } d(\mathbf{r}_k, \mathbf{r}_l) = (\mathbf{r}_k - \mathbf{r}_l)^T \mathbf{W}^* (\mathbf{r}_k - \mathbf{r}_l)$$

Berechne den Nachbarschaftsradius R_h der Stützstelle $\mathbf{r}_h^p$ nach:

$$R_h = \begin{cases} (d_{\min} + d_{\max})/2, & \text{falls } d_{\min} \geq d_{\max} \\ \rho\, d_{\min}, & \text{sonst} \end{cases} \text{, mit } 0 < \rho < 1$$

$\mathbf{Z} = \varnothing$

FOR alle unklassifizierten Merkmalsvektoren $\mathbf{r}$ der Datenbank (b)

Bestimme die am nächsten benachbarte Stützstelle $\mathbf{r}_h^p : h = \underset{i=1,\ldots,H}{\operatorname{argmin}}\, d(\mathbf{r}, \mathbf{r}_i^p)$

IF $d(\mathbf{r}, \mathbf{r}_h^p) \leq R_h$

TRUE	FALSE
Klassifiziere $\mathbf{r}$ als relevant ($Z = Z \cup \{\mathbf{r}\}$) und berechne gemäß Gleichung 4.23 den Zugehörigkeitswert $\pi(\mathbf{r})$.	$\mathbf{r}$ bleibt unklassifiziert

Berechne ausgehend von der Vereinigungsmenge $X^p \cup Z$ gemäß Gleichung 4.24 und 4.25 für den nächsten Suchschritt den idealen Anfragevektor $\mathbf{q}^*$ und die optimale Gewichtsmatrix $\mathbf{W}^*$.

Abb. 4.6: Lernen mit Stichprobenerweiterung durch Co-Training

werden die positiven Merkmalsvektoren als Präzedenzfälle bzw. Stützstellen im Merkmalsraum betrachtet. Die Vektoren der unklassifizierten Bilder werden der Stützstelle zugeordnet, zu der sie den minimalen Abstand besitzen. Das für die Abstandsberechnung notwendige Abstandsmaß basiert auf den positiven Elementen der klassifizierten Stichprobe. In einem überwachten Lernschritt (vgl. Abschnitt (a) in Abbildung 4.6) wird zunächst entsprechend Gleichung 4.13 die optimale Gewichtsmatrix $\mathbf{W}^*$ berechnet. Falls die Kovarianzmatrix der Datenmenge nicht invertierbar ist, wird lediglich eine diagonale Gewichtsmatrix verwendet. Das Abstandsmaß reduziert sich demnach, wie in Abschnitt 4.3.1 beschrieben, von einem generalisierten zu einem gewichteten euklidischen Abstand.

Um zu verhindern, dass ein Element zu einer Stützstelle klassifiziert wird, obwohl sein Abstand sehr groß ist, wird für jede Stützstelle ein Klassenradius eingeführt, sodass der einfache Nachbarschaftsklassifikator zu einem beschränkten Nachbarschaftsklassifikator erweitert wird. Demnach wird ein Vektor $\mathbf{r}$ nur dann einem Element $\mathbf{r}_h^p$ zugeordnet, wenn sein Abstand zu diesem minimal ist und er außerdem innerhalb des Klassengebietes liegt, das durch den korrespondierenden Radius bestimmt ist. Die Berechnung des Klassenradius ist durch die Arbeit von Qian et al. [Qia02] motiviert und basiert sowohl auf der Menge der positiven als auch auf der Menge der negativen Trainingselemente. Zunächst wird für eine Stützstelle $\mathbf{r}_h^p$ der maximale Abstand $d_{\max}$ zu den übrigen positiven Elementen sowie der minimale Abstand $d_{\min}$ zu den Elementen der negativen Stichprobe bestimmt:

$$\left.\begin{array}{l} d_{\max} = \max\limits_{\mathbf{r}_k^p \in X^p} d(\mathbf{r}_h^p, \mathbf{r}_k^p) \\ d_{\min} = \min\limits_{\mathbf{r}_k^n \in X^n} d(\mathbf{r}_h^p, \mathbf{r}_k^n) \end{array}\right\} \text{mit } d(\mathbf{r}_k, \mathbf{r}_l) = (\mathbf{r}_k - \mathbf{r}_l)^T \mathbf{W}^* (\mathbf{r}_k - \mathbf{r}_l)$$

Darauf aufbauend gilt für den Klassenradius R_h einer Stützstelle $\mathbf{r}_h^p$:

$$R_h = \begin{cases} (d_{\min} + d_{\max})/2, & \text{falls } d_{\min} \geq d_{\max} \\ \rho\, d_{\min} \quad , & \text{sonst} \end{cases}$$

Dabei repräsentiert ρ eine Konstante, die die Bedingung $0 < \rho < 1$ erfüllt. Da von nun an für jede Stützstelle ein Klassenradius existiert, sind die Voraussetzungen für den unüberwachtem Klassifikationsschritt (vgl. Abschnitt (b) in Abbildung 4.6) gegeben. Um jedoch zu verhindern, dass eine Fehlklassifikation[11] die Qualität des zu trainierenden Klassifikators negativ beeinflusst, wird auf eine harte Zuordnung verzichtet und stattdessen eine weiche Klassifikation durchgeführt. Der Zugehörigkeitswert $\pi(r)$ eines unklassifizierten Musters $\mathbf{r}$ ist sowohl vom Abstand $d(\mathbf{r}, \mathbf{r}_h^p)$ zu der entsprechenden Stützstelle $\mathbf{r}_h^p$ als auch vom korrespondierenden Radius R_h abhängig:

$$\pi(\mathbf{r}) = e^{-\frac{z^2}{2\sigma^2}}, \tag{4.23}$$

mit $z = d(\mathbf{r}, \mathbf{r}_h^p)$ und $\sigma = R_h/2$. Aus dem unüberwachten Lernschritt resultiert eine Menge Z von neu klassifizierten Merkmalsvektoren. Diese bilden zusammen mit der ursprünglich als positiv klassifizierten Stichprobe X^p die Trainingsmenge für den abschließenden Lernschritt. Während die Repräsentanten $\mathbf{r}_h^p$ der vom Benutzer als positiv bewerteten Bilder mit der entsprechenden Bildbewertung $\pi(\mathbf{r}_h^p) \in \{1, 2\}$ in den Adaptionsschritt eingehen, entspricht das Gewicht eines neuen Trainingsbeispiels $\mathbf{r} \in Z$ dem Zugehörigkeitswert $\pi(\mathbf{r})$, mit dem es zur Stützstelle klassifiziert wurde. Ausgehend von der Vereinigungsmenge $X^p \cup Z$ und den korrespondierenden Gewichten $\pi(\mathbf{r})$,

[11] In dem hier betrachteten Szenario wird unter einer Fehlklassifikation die Tatsache verstanden, dass ein unklassifiziertes Muster als relevant klassifiziert wurde, obwohl das korrespondierende Bild nicht zur gesuchten Bildkategorie gehört.

mit $\mathbf{r} \in X^p \cup Z$, erfolgt die Berechnung des idealen Anfragevektors $\mathbf{q}^*$ sowie der optimalen Gewichtsmatrix $\mathbf{W}^*$ analog zu Gleichung 4.12, 4.13 und 4.15:

$$\mathbf{q}^* = \frac{\sum_{\mathbf{r} \in X^p \cup Z} \pi(\mathbf{r})\, \mathbf{r}}{\sum_{\mathbf{r} \in X^p \cup Z} \pi(\mathbf{r})} \tag{4.24}$$

$$\mathbf{W}^* = (\det(\mathbf{C}))^{\frac{1}{N}} \mathbf{C}^{-1}, \tag{4.25}$$

wobei die Elemente c_{mn} der Kovarianzmatrix $\mathbf{C}$ wie folgt definiert sind:

$$c_{mn} = \frac{\sum_{\mathbf{r} \in X^p \cup Z} \pi(\mathbf{r})(r_m - q_m^*)(r_n - q_n^*)}{\sum_{\mathbf{r} \in X^p \cup Z} \pi(\mathbf{r})} \tag{4.26}$$

4.4 Evaluation

In den vorangegangenen Abschnitten wurde der für das INDI System entwickelte Suchprozess formal beschrieben und verschiedene Varianten des Systemlernens vorgestellt. Um jedoch die Leistungsfähigkeit des Systems zu zeigen und eine Aussage über die Qualität der verschiedenen Ansätze treffen zu können, ist eine Evaluation erforderlich. Diese bildet den Schwerpunkt der folgenden Abschnitte.

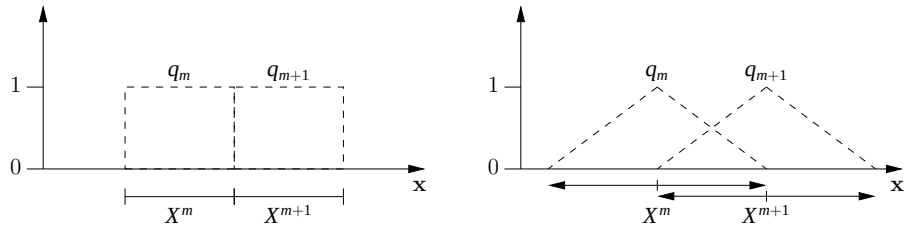
4.4.1 Merkmalsextraktion

Die Evaluation des Bildsuchsystems INDI erfordert die Extraktion inhärenter Bildcharakteristika, die die Grundlage der inhaltsbasierten Suche bilden (vgl. Abschnitt 2.2). Im Folgenden werden die im Rahmen der Evaluation verwendeten Farb-, Textur- und Formrepräsentanten beschrieben.

Farbe

Die in der Regel gebräuchlichste Methode zur Repräsentation der Farben innerhalb eines Bildes stellt eine Wahrscheinlichkeitsverteilung dar. Die Grundlage ihrer Berechnung bilden die Farbwerte eines Bildes, deren Häufigkeit schließlich in der Verteilung kodiert ist. Neben parametrischen Verteilungen, wie z.B. Gauß- oder Mischverteilungen [Dud73, Nie83], sind es vor allem nicht parametrische Verteilungen wie Histogramme, die häufig zur Beschreibung der Farben eines Bildes dienen [Str94, Zha95]. Ihre Vorteile liegen dabei insbesondere in der einfachen Berechnung und der nicht erforderlichen Festlegung auf ein bestimmtes Verteilungsmodell.

Die Grundlage der klassischen Histogramme bildet die Einteilung des verwendeten Farbraumes X (häufig Merkmalsraum der Dimension $D = 3$, wie z.B. die im Anhang B beschriebenen Farbräume RGB, HSI, CIE $L^*u^*v^*$ oder CIE $L^*a^*b^*$) in nicht



(a) Klassisches Histogramm (b) Fuzzyhistogramm

Abb. 4.7: Beispiele für Histogrammzugehörigkeitsfunktionen eines eindimensionalen Merkmalsvektors: (a) klassisches Histogramm (b) Fuzzyhistogramm nach Siggelkow [Sig02].

überlappende Teilräume X^m, wobei $m = 0, 1, \ldots, M-1$ ist und M die Anzahl der Quantisierungsstufen der Farbwerte des Ursprungsraumes repräsentiert. Formal gilt für einen quantisierten Farbraum X:

$$X^m \subset X, \quad \bigcup_{m=0}^{M-1} X^m = X, \text{ mit } X^m \cap X^n = \varnothing, \ \forall m \neq n$$

Für eine gegebene Menge von I Farbwerten $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$ mit $\mathbf{x}_i \in X$ lässt sich die Wahrscheinlichkeit für das Auftreten eines Farbwertes $\mathbf{x}$ der Region X^m wie folgt abschätzen:

$$P_m(\mathbf{x} \in X^m) = \frac{1}{I} \sum_{i=1}^{I} q_m(\mathbf{x}_i),$$

mit

$$q_m : X \to \{0, 1\} \quad \text{und} \quad q_m(\mathbf{x}_i) = \begin{cases} 1, & \text{falls } \mathbf{x}_i \in X^m \\ 0, & \text{sonst} \end{cases}$$

Abbildung 4.7(a) veranschaulicht die Zugehörigkeitsfunktion eines eindimensionalen Merkmalsvektors[12] zu einer Region eines klassischen Histogramms. Dabei ist zu beachten, dass bei einem solchen Histogramm eine eindeutige Zuordnung für den Farbwert eines Pixels zu einer Region existiert. Aus dieser harten oder auch scharfen Zugehörigkeit resultieren Unstetigkeiten an den Regionengrenzen. Eine Lösung dieses Problems bieten die von Swain und Ballard [Swa91] erwähnten und von Siggelkow [Sig02] ausführlich formalisierten Fuzzyhistogramme. Dabei wird auf eine eindeutige Zuordnung eines Farbwertes zu einer Region des Farbraumes verzichtet und stattdessen eine weiche, unscharfe Zugehörigkeit verwendet. Wird von einer gleichmäßigen Aufteilung des Ursprungsraumes in verschiedene Hyperkuben ausgegangen, dann ist der Beitrag eines Farbwertes zu einem Eintrag des Histogramms

[12] z.B. der Intensitätswert eines Farbkanals

umso größer, je näher er dem Zentrum des korrespondierenden Hyperkubus ist. Aufgrund der Fuzzyfizierung liefert ein Farbwert allerdings auch noch einen Beitrag zu den benachbarten Histogrammeinträgen. In Abbildung 4.7(b) ist ein Beispiel für die Zugehörigkeitsfunktionen eines Fuzzyhistogramms gegeben. Die dargestellten Funktionen q_m sind jeweils kontinuierlich, d.h. sie steigen gleichmäßig von Null bis Eins an und nehmen dann wieder linear von Eins bis Null ab. Ihre Funktionswerte erfüllen die Eigenschaft der Positivität, $q_m(\mathbf{x}) \geq 0$. Außerdem überlappen sich die Zugehörigkeitsfunktionen benachbarter Regionen so, dass sich ihre Funktionswerte zu Eins summieren (*Partition of Unity*). Nach Siggelkow [Sig02, S.34f] lässt sich die Berechnung eines Fuzzyhistogramms mit den beschriebenen dreieckigen Zugehörigkeitsfunktionen folgendermaßen herleiten:

Gegeben sei für jede Komponente eines D dimensionalen Merkmalsraumes eine Quantisierung in M_d Regionen, mit $d = 0, 1, \ldots, D-1$. Die Komponenten eines Stichprobenelements $\mathbf{x} \in \mathbb{R}^D$ werden durch

$$\tilde{x}_d = (x_d - x_d^{\min}) \frac{(M_d - 1)}{x_d^{\max} - x_d^{\min}} , \quad x_d \in [x_d^{\min}, x_d^{\max})$$

auf die Anzahl der Quantisierungsstufen M_d normiert, $\tilde{\mathbf{x}} = (\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_{D-1})^T$. Durch die im Zähler durchgeführte Subtraktion (-1) wird berücksichtigt, dass jede Komponente an den Intervallgrenzen jeweils Zugehörigkeitsfunktionen halber Breite besitzt (vgl. Abbildung 4.8(b)). Die zum normierten Stichprobenelement $\tilde{\mathbf{x}}$ benachbarten Zentren der Hyperkuben[13] sind durch

$$\tilde{\mathbf{x}}^{(b_0,\ldots,b_{D-1})} = \sum_{d=0}^{D-1} (\lfloor \tilde{x}_d \rfloor + b_d)\, \mathbf{e}_d \,, \text{ mit } b_d \in \{0,1\} \text{ und } \mathbf{e}_d : d\text{-ter Einheitsvektor},$$

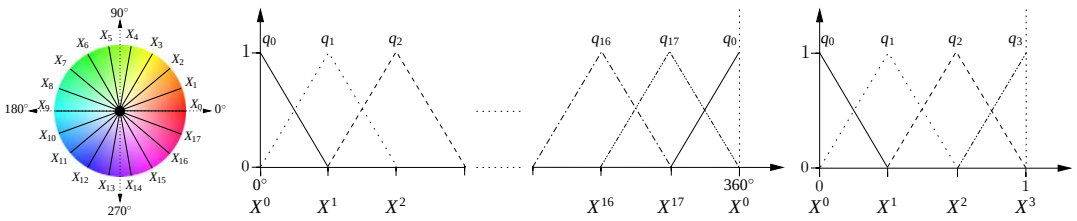
definiert. Der Beitrag $\Delta P_{\tilde{\mathbf{x}}^{(b_0,\ldots,b_{D-1})}}$ eines Stichprobenelements $\tilde{\mathbf{x}}$ zum Histogrammwert des Hyperkubus mit dem Zentrum $\tilde{\mathbf{x}}^{(b_0,\ldots,b_{D-1})}$ folgt schließlich aus

$$\Delta P_{\tilde{\mathbf{x}}^{(b_0,\ldots,b_{D-1})}}(\mathbf{x}) = \frac{1}{N} \prod_{d=0}^{D-1} \left(1 - \left|\tilde{x}_d - \tilde{x}_d^{(b_0,\ldots,b_{D-1})}\right|\right),$$

wobei N die Anzahl der Stichprobenelemente repräsentiert.

Die Grundlage der zur Evaluation verwendeten Fuzzyhistogramme bildet der in Anhang B.2 beschriebene HSI Farbraum. Dieser Farbraum entspricht mit seiner Repräsentation von Farbton (H), Sättigung (S) und Intensität (I) stärker der menschlichen Farbwahrnehmung und Farbinterpretation als der additive RGB Farbraum (vgl. Anhang B). Bei der Histogrammberechnung wird zwischen Farb- und Grautönen unterschiedenen. Bildpunkte, deren Sättigungs- und Intensitätswert größer als 0.03 und

[13] Aus den verschiedenen Kombinationsmöglichkeiten aller $b_d \in \{0,1\},\ d = 0, 1, \ldots, D-1$, ergeben sich 2^D benachbarte Zentren $\tilde{\mathbf{x}}^{(b_0,\ldots,b_{D-1})}$. Für $\tilde{x}_d = \lfloor \tilde{x}_d \rfloor$ sind die nächsten Nachbarn durch $\tilde{x}_d$ und $\tilde{x}_d + 1$ gegeben.



(a) Farbkreis (b) Farbton (c) Grauton

Abb. 4.8: Zentren der Farbkreisregionen und Zugehörigkeitsfunktionen der Fuzzyhistogrammberechnung. Der in (a) dargestellte Farbkreis veranschaulicht die Zentren der verschiedenen Regionen des in 18 Stufen quantisierten Farbtonkanals. Die Zugehörigkeit eines Farbwertes zu diesen Regionen wird durch die in (b) dargestellten Zugehörigkeitsfunktionen bestimmt. Zusammen mit den in (c) abgebildeten Zugehörigkeitsfunktionen der verschiedenen Grauwertregionen bilden sie die Grundlage der Farbton- und Grauwerthistogrammberechnung.

0.05 sind werden als farbig klassifiziert.[14] Umgekehrt werden die Farbwerte der Pixel, die diese Schwellen unterschreiten als Grauton eingestuft. Auf der Basis dieser Klassifikation werden schließlich ein in achtzehn Quantisierungsstufen unterteiltes Farbtonsowie ein in vier Stufen quantisiertes Grautonhistogramm berechnet. Die entsprechenden Zugehörigkeitsfunktionen der Histogramme sowie die Zentren der verschiedenen Regionen des quantisierten Farbtonkanals sind in Abbildung 4.8 dargestellt. Dabei ist zu beachten, dass die Zirkularität des Farbtonkanals durch eine weiche Repräsentation der Zugehörigkeit, wie sie in Abbildung 4.8(b) dargestellt ist, sehr einfach berücksichtigt werden kann. Beide Histogramme werden durch die Anzahl der Bildpunkte normiert. Das endgültige Fuzzyhistogramm und somit der beschreibende Merkmalsvektor eines Bildes resultieren schließlich aus der Verkettung der vektoriellen Einzelhistogramme:

$$\mathbf{r}_1 = \mathbf{r}_{\text{Fuzzyhistogramm}} = \mathbf{r}_{\text{Farbtonhistogramm}} \oplus \mathbf{r}_{\text{Grautonhistogramm}},$$

mit

$$\mathbf{r}_{\text{Farbtonhistogramm}} \in \mathbb{R}^{18},\ \mathbf{r}_{\text{Grautonhistogramm}} \in \mathbb{R}^{4} \text{ und } \mathbf{r}_{\text{Fuzzyhistogramm}} \in \mathbb{R}^{22}$$

Ein weiteres Verfahren zur Beschreibung der Farbcharakteristik eines Bildes sind die von Stricker und Orengo vorgestellten Farbmomente [Str95]. Dabei werden anstatt einer kompletten Farbverteilung die ersten drei Momente Mittelwert, Varianz und Schiefe dazu verwendet die Farben eines Bildes zu repräsentieren (vgl. auch Abschnitt 2.2.1). Die Erweiterung dieses Verfahrens verzichtet auf die Berechnung des

[14] Die Schwellwerte wurden per Hand gesetzt und spiegeln die subjektive Farbempfindung des Autors wider.

dritten Moments und verwendet stattdessen die Kovarianzmatrix der vektoriellen Farbbeschreibungen, da diese mehr Information beinhaltet (vgl. [Str97]). Basierend auf der Farbdarstellung in dem nahezu perzeptuell linearen CIE $L^*u^*v^*$ Farbraum (vgl. Anhang B.3) lässt sich dieser Bildrepräsentant für ein Bild B mit M Bildpunkten durch

$$\boldsymbol{\mu} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_m\,,\ \text{mit}\ \mathbf{x}_m, \boldsymbol{\mu} \in \{L^*, u^*, v^*\}$$

und

$$\mathbf{C} = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{x}_m - \boldsymbol{\mu})(\mathbf{x}_m - \boldsymbol{\mu})^T$$

berechnen. Da die Kovarianzmatrix $C = [c_{ij}]_{3,3}$ symmetrisch ist und sie dementsprechend in dem dreidimensionalen Merkmalsraum durch sechs Komponenten vollständig beschrieben ist, gilt für den resultierenden Merkmalsvektor:

$$\mathbf{r} = (\mu_1, \mu_2, \mu_3, c_{11}, c_{12}, c_{13}, c_{22}, c_{23}, c_{33})^T$$

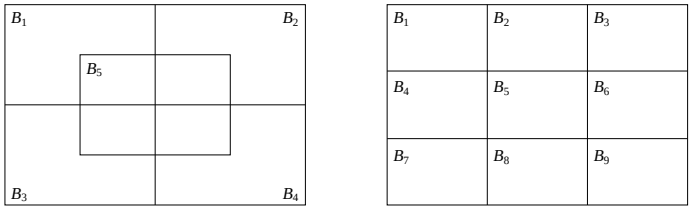
Ergänzend zum bereits beschriebenen Fuzzyhistogramm, in dem zwar das Auftreten der verschiedenen Bildfarben kodiert ist, nicht aber wo diese Farben im Bild lokalisiert sind, soll in der Evaluation ein weiterer Farbdeskriptor verwendet werden, der auch lokale Farbeigenschaften repräsentiert (vgl. Abschnitt 2.2.1). Deshalb wird auf eine globale Berechnung der Farbmomente verzichtet. Stattdessen wird ein Bild entsprechend Abbildung 4.9(a) gerastert. In jedem Bildraster B_r, mit $r = 1, \ldots, 5$, werden sowohl der Mittelwertsvektor als auch die Kovarianzmatrix der Farbvektoren $\mathbf{x} \in \{L^*, u^*, v^*\}$ berechnet. Die daraus resultierenden neundimensionalen Merkmalsvektoren der Bildraster werden in einem abschließenden Verarbeitungsschritt zu einem Gesamtvektor

$$\mathbf{r}_2 = \mathbf{r}_{\text{Farbmomente}} = \mathbf{r}_{B_1} \oplus \mathbf{r}_{B_2} \oplus \mathbf{r}_{B_3} \oplus \mathbf{r}_{B_4} \oplus \mathbf{r}_{B_5}, \quad \mathbf{r}_{\text{Farbmomente}} \in \mathbb{R}^{45}.$$

kombiniert. In der resultierenden vektoriellen Bildbeschreibung ist somit neben der Farbinformation auch kodiert, in welchem Bildbereich die Farben auftreten.

Textur

Für die im Rahmen der Evaluation verwendeten Texturrepräsentanten dient die Arbeit von Wagner [Wag99] als Grundlage. Darin werden achtzehn verschiedene Deskriptoren untersucht und miteinander verglichen. Unter diesen Merkmalen befinden sich z.B. die von Haralick et al. [Har73] vorgestellten und auf Grauwertübergangsmatrizen basierenden Texturcharakteristika sowie die durch Filterung mit Gabor Wavelets extrahierten Texturbeschreibungen (vgl. Abschnitt 2.2.2). Aus den Ergebnissen der von Wagner [Wag99] durchgeführten experimentellen Untersuchungen geht hervor, dass



(a) Farbe (b) Textur

Abb. 4.9: Bildrasterung zur Berechnung der lokalen Farbmomente (a) und Texturmerkmale (b)

die besten Klassifikationsergebnisse mit den von Unser [Uns86] vorgestellten Texturmerkmalen erzielt werden können. Deshalb werden neben den bereits beschriebenen Farbcharakteristika diese Merkmale dazu verwendet, den visuellen Inhalt eines Bildes zu beschreiben. In den folgenden Abschnitten werden die Details der Repräsentantenberechnung näher erläutert.

Die Grundlage des Verfahrens bildet die Berechnung der Summen- und Differenzbilder, wobei die Summe und die Differenz zweier Pixel mit der relativen Verschiebung (d_1, d_2) durch

$$\begin{aligned} s_{k,l} &= p_{k,l} + p_{k+d_1,l+d_2} \\ d_{k,l} &= p_{k,l} - p_{k+d_2,l+d_2} \end{aligned}$$

definiert ist. Dabei repräsentiert $p_{k,l} \in [1, N_G]$ den Grauwert eines Pixels (k, l). Darauf aufbauend gilt für die Summen- und Differenzhistogramme eines Bildes B der Größe M bei einer gegebenen Parametrisierung (d_1, d_2):

$$\begin{aligned} h_s(i; d_1, d_2) &= h_s(i) = \text{card}\{(k, l) \in B | s_{k,l} = i\} \\ h_d(j; d_1, d_2) &= h_d(j) = \text{card}\{(k, l) \in B | d_{k,l} = j\} \end{aligned}$$

Die Summe der einzelnen Histogrammeinträge entspricht der Anzahl der Bildpunkte

$$M = \text{card}(B) = \sum_i h_s(i) = \sum_j h_d(j),$$

sodass die normierten Summen- und Differenzhistogramme durch

$$\begin{aligned} P_s(i) &= h_s(i)/M; \quad (i = 2, \ldots, 2N_g) \\ P_d(j) &= h_d(j)/M; \quad (j = -N_g + 1, \ldots, N_g - 1) \end{aligned}$$

definiert sind und in ihnen die Wahrscheinlichkeit für das Auftreten einer Summe i oder Differenz j kodiert ist. Obwohl die berechneten Histogramme durchaus als Bildrepräsentanten eingesetzt werden könnten, sind sie viel zu hochdimensional und dement-

sprechend für die inhaltsbasierte Bildersuche nicht zweckmäßig. Deshalb werden auf ihrer Grundlage die folgenden statistischen Charakteristika[15] berechnet [Uns86]:

$$\begin{aligned}
\text{Mittelwert:} \quad f_1 &= \frac{1}{2}\sum_i i\, P_s(i) = \mu \\
\text{Varianz:} \quad f_2 &= \frac{1}{2}\left(\sum_i (i-2\mu)^2\, P_s(i) + \sum_j j^2\, P_d(j)\right) \\
\text{Energie:} \quad f_3 &= \sum_i P_s(i)^2 \sum_j P_d(j)^2 \\
\text{Entropie:} \quad f_4 &= -\sum_i P_s(i)\ \log(P_s(i)) - \sum_j P_d(j)\ \log(P_d(j)) \\
\text{Kontrast:} \quad f_5 &= \sum_j j^2\, P_d(j)
\end{aligned}$$

Basierend auf der Parametrisierung (d_1, d_2) wird ein Bild somit durch fünf Charakteristika beschrieben:

$$\mathbf{r}_{d_1,d_2} = (f_1, f_2, f_3, f_4, f_5)^T \in \mathbb{R}^5$$

Analog zur Merkmalsberechnung auf der Basis von Grauwertübergangsmatrizen [Har73] kann die relative Verschiebung $(d_1, d_2)^T$ der Spalten und Zeilen ebenso gut durch einen Abstand d und eine Orientierung θ dargestellt werden:

$$\mathbf{r}_{d,\theta} = (f_1, f_2, f_3, f_4, f_5)^T, \text{ mit } d = \|(d_1, d_2)^T\| \text{ und } \theta = \sphericalangle((d_1, d_2)^T, \mathbf{e}_{\text{Zeile}})$$

Gängige Parametrisierungen sind dabei der Pixelabstand $d = 1$ sowie die Orientierungen $\theta = \{0°, 45°, 90°, 135°\}$, die auch in der vorliegenden Arbeit zur Berechnung der verschiedenen Summen- und Differenzhistogramme verwendet werden. Das zu bearbeitende Graustufenbild wird zusätzlich auf 32 Intensitätswerte quantisiert. Des Weiteren wird auf eine globale Berechnung eines Texturrepräsentanten verzichtet. Stattdessen wird ähnlich wie bei der im vorherigen Abschnitt beschriebenen Berechnung des zweiten Farbrepräsentanten jedes Bild in verschiedene Teilbilder unterteilt. Die entsprechende Bildpartitionierung ist in Abbildung 4.9(b) dargestellt. In jedem Teilbild B_i, $i = 1, \ldots, 9$, werden die Texturdeskriptoren $\mathbf{r}_{1,0°}(B_i), \mathbf{r}_{1,45°}(B_i), \mathbf{r}_{1,90°}(B_i)$ und $\mathbf{r}_{1,135°}(B_i)$ berechnet. Ihre Verkettung ergibt die lokalen Repräsentanten

$$\mathbf{r}_{B_i} = \mathbf{r}^{B_i}_{1,0°} \oplus \mathbf{r}^{B_i}_{1,45°} \oplus \mathbf{r}^{B_i}_{1,90°} \oplus \mathbf{r}^{B_i}_{1,135°},$$

die wiederum zu einem globalen Texturrepräsentanten $\mathbf{r}_{\text{Unser}}$ zusammengefasst werden:

$$\mathbf{r}_3 = \mathbf{r}_{\text{Unser}} = \mathbf{r}_{B_1} \oplus \mathbf{r}_{B_2} \oplus \cdots \oplus \mathbf{r}_{B_9}, \quad \mathbf{r}_{\text{Unser}} \in \mathbb{R}^{180}.$$

Diese Methode der Texturrepräsentation hat den Vorteil, dass in dem globalen Merkmalsvektor neben der reinen Texturinformation auch kodiert ist, wo diese Information im Bild lokalisiert ist.

[15] Diese Charakteristika sind zu den gleichnamigen statistischen Merkmalen äquivalent, die Haralick et al. [Har73] auf der Basis von Grauwertübergangsmatrizen berechnen.

Form

Neben Farbe und Textur sind Formmerkmale eine weitere Möglichkeit, Bildinhalte zu beschreiben und Bilder miteinander zu vergleichen. Während bei einigen dieser Techniken die Objekterkennung eine wichtige Voraussetzung für die Berechnung darstellt, existieren auch andere Verfahren, die formspezifische Charakteristika auf der Basis des ganzen Bildes extrahieren (vgl. Abschnitt 2.2.3). Erstgenannte Verfahren erfordern zur Objektdetektion entweder die Einschränkung der Bilddomäne und die Anwendung spezieller Segmentierungs- sowie Gruppierungsverfahren oder die manuelle Unterstützung durch einen Anwender. Da die allgemeine Anwendbarkeit eine wichtige und notwendige Voraussetzung für die Navigation in der verwendeten heterogenen Bildsammlung ist und da sich auch die manuelle Unterstützung zur Objektdetektion ähnlich problematisch darstellt wie der in Kapitel 1 beschriebene Prozess der Bildverschlagwortung[16], wird im Rahmen der Evaluation auf die Verwendung derartiger Formcharakteristika verzichtet. Stattdessen werden statistische Merkmale benutzt, die auf einem kompletten Bild bzw. einem Teilbild berechnet werden können und keine Bildsegmentierung voraussetzen.

Brandt et al. [Bra99, Bra00] untersuchen in ihren Arbeiten verschiedene solcher statistischer Formcharakteristika. Dabei handelt es sich um Kantenhistogramme, Kantenübergangsmatrizen sowie verschiedene Varianten von Fourierdeskriptoren (einfach, polar und logarithmisch-polar). Aus den experimentellen Untersuchungen dieser Arbeiten geht hervor, dass mit dem einfachen Kantenhistogramm gute Ergebnisse erzielt werden können. Da darüber hinaus das Histogramm mit acht Dimensionen sehr niedrigdimensional ist und dementsprechend wenig Parameter während des Suchprozesses geschätzt werden müssen (vgl. Abschnitt 4.3.1), wird dieser Deskriptor dazu verwendet, die Formen innerhalb eines Bildes zu beschreiben. Die Berechnung des Bildrepräsentanten erfolgt weitestgehend in Analogie zu der Berechnung von Brandt et al. [Bra99, Bra00]:

Im ersten Schritt wird das zu beschreibende Bild in den HSI Farbraum (vgl. Anhang B.2) transformiert, von dem im weiteren Verlauf der Berechnung der Farbtonkanal nicht weiter berücksichtigt wird. Seine Zirkularität würde zur Kantendetektion eine spezielle Handhabung erfordern, auf die an dieser Stelle verzichtet werden soll. Die Merkmalsextraktion erfolgt daher ausschließlich auf der Basis des Sättigungs- und Intensitätskanals. Für jeden dieser Kanäle werden durch Filterung mit den in Abbildung 4.10 dargestellten Sobeloperatoren die Gradienten $\nabla \mathbf{f}(x,y) = (G_x, G_y)^T = (\partial B(x,y)/\partial x, \partial B(x,y)/\partial y)^T$ der verschiedenen Bildpunk-

[16] Ebenso wie die Bildverschlagwortung würde die manuelle Objektdetektion bei den umfangreichen Datenmengen, wie sie gewöhnlich in Bilddatenbanken verwaltet werden, kaum effizient durchführbar sein. Erschwert würde dieser Prozess außerdem durch die subjektive Wahrnehmung und Empfindung darüber, welche Bestandteile eines Bildes eine semantische Einheit bilden und daraus resultierend als Objekt bezeichnet werden (vgl. [Mar01]).

-1	0	1
-2	0	2
-1	0	1

(a) x-Richtung

-1	-2	-1
0	0	0
1	2	1

(b) y-Richtung

Abb. 4.10: Sobeloperatoren nach Gonzales und Woods [Gon02, S. 578]

te (x, y) berechnet. Die zur Detektion der Bildkanten und Extraktion der Formcharakteristika notwendigen Gradientenstärken und Gradientenrichtungen sind durch

$$|\nabla \mathbf{f}(x,y)| = \left| \begin{pmatrix} G_x \\ G_y \end{pmatrix} \right| = \sqrt{G_x^2 + G_y^2}$$

und

$$\nabla \mathrm{f}_{\text{Richtung}}(x,y) = \arctan\left(\frac{G_y}{G_x}\right)$$

definiert. Die Gradientenrichtungen werden außerdem auf die Quantisierungsstufen $0°, 45°, 90°, 135°, 180°, 225°, 270°$ und $315°$ abgebildet, die durch die Indizes $i = 1, 2, \ldots, 8$, repräsentiert werden. Im nächsten Verarbeitungsschritt werden die resultierenden Gradientenstärkebilder durch Anwendung eines Schwellwertes binarisiert, sodass starke Kanten erhalten bleiben und schwache Kanten beseitigt werden. Der Schwellwert ist für alle zu verarbeitende Bilder identisch und beträgt für den Sättigungskanal 35% und für den Intensitätskanal 15% der maximalen Gradientenstärke.

Nach der Binarisierung werden die beiden Binärbilder durch logische ODER-Verknüpfung zu einem Kantenbild B_{Kante} kombiniert, sodass für ein Kantenpixel $B_{\text{Kante}}(x,y) = 1$ und für ein Nicht-Kantenpixel $B_{\text{Kante}}(x,y) = 0$ gilt. Zusätzlich werden die Gradientenrichtungen der Kantenpixel in ein Gradientenrichtungsbild B_{Richtung} eingetragen. Sollten für ein Bildpunkt (x, y) die Gradientenrichtungen des Sättigungs- und Intensitätskanals unterschiedlich sein, wird die Richtung des größten Gradienten ausgewählt. Aufbauend auf der Kantenextraktion lässt sich das Kantenhistogramm H eines Bildes B der Größe $M \times N$ nach

$$H(i) = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} B_{\text{Kante},i}(x,y), \text{ mit } i = 1, \ldots, 8$$

berechnen, wobei $B_{\text{Kante},i}(x,y)$ ein Kantenpixel (x, y) mit der Gradientenrichtung i repräsentiert ($B_{\text{Richtung}}(x,y) = i$). Um außerdem eine Invarianz von der Bildskalierung zu

erzielen, werden die Histogrammeinträge analog zu Jain und Vailaya [Jai96] durch die Anzahl der Kantenpixel normiert:

$$h(i) = \frac{H(i)}{\sum_{i=1}^{8} H(i)}$$

Für den endgültige Bildrepräsentanten gilt schließlich:

$$\mathbf{r}_4 = \mathbf{r}_{\text{Kantenhistogramm}} = (h_1, h_2, \ldots, h_8)^T, \quad \mathbf{r}_{\text{Kantenhistogramm}} \in \mathbb{R}^8$$

Optimierung der Bildrepräsentanten

Ziel des abschließenden Verarbeitungsschritts ist die Optimierung der berechneten Bildbeschreibungen. Dies ist sinnvoll, um einerseits bestehende Korrelation zwischen Vektorkomponenten zu bereinigen und andererseits eine kompakte Repräsentation zu erzielen. In dem Optimierungsschritt erfolgt daher durch die Anwendung der in Anhang C beschriebenen Hauptachsentransformation sowohl eine Dekorrelation der Merkmalsvektoren als auch eine Reduktion ihrer Dimensionalität. Der Grad der Dimensionsreduktion wurde dabei so gewählt, dass jeweils 90% der ursprünglichen Information erhalten bleibt. Dementsprechend werden die Farbdeskriptoren auf jeweils zehn bzw. siebzehn Dimensionen reduziert. Die Texturcharakteristika und Kantenhistogramme werden in einen neunzehn- bzw. fünfdimensionalen Eigenraum transformiert.

Abgeschlossen wird die Optimierung der Bildrepräsentanten durch die Varianznormierung[17] der einzelnen Merkmalskomponenten. Damit wird verhindert, dass die numerischen Dynamikbereiche der Komponenten zu stark variieren und eine Komponente während der Abstandsberechnung die anderen dominiert. Die unabhängige Normierung der Vektorkomponenten erfolgt analog zu der von Aksoy und Haralick [Aks01] beschriebenen Berechnung und verfolgt die Zielsetzung, jede Komponente in das Intervall $[0, 1]$ abzubilden:

Gegeben sei eine Menge von Bildrepräsentanten $R = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K\}$ der Dimension N. Die Elemente einer Dimension werden als Messreihe $R_n = \{r_{1_n}, r_{2_n}, \ldots, r_{K_n}\}$ interpretiert, für die Mittelwert μ_n und Standardabweichung σ_n bestimmt werden. Die normierte Komponente $\tilde{r}_{k_n}$ ist durch

$$\tilde{r}_{k_n} = \frac{r_{k_n} - \mu_n}{\sigma_n}$$

definiert. Unter der Voraussetzung, dass die $\{r_{k_n} | k = 1, 2, \ldots, K\}$ normalverteilt sind, folgt, dass die entsprechenden $\tilde{r}_{k_n}$ zu 68% im Intervall $[-1, 1]$ liegen. Durch die zusätzliche Verschiebung und Skalierung nach

$$\tilde{r}_{k_n} = \frac{(r_{k_n} - \mu_n)/(3\sigma_n) + 1}{2}$$

[17] Die Kombination von Dekorrelation und anschließender Normierung der Varianzanteile wird als „Weissen“ (engl. *Whitening*) bezeichnet (vgl. z.B. [Fin03, S. 145f]).

ist garantiert, dass 99% aller $\tilde{r}_{k_n}$ Elemente des Intervalls $[0, 1]$ sind. Elemente, die sich außerhalb dieses Intervalls befinden, werden einfach auf die benachbarte Intervallgrenze, 0 oder 1, abgebildet.

4.4.2 Evaluation inhaltsbasierter Bildsuchsysteme

Die Evaluation inhaltsbasierter Bildsuchsysteme ist zwar ein wesentlicher Bestandteil des Systementwurfs, allerdings existiert bislang weder ein standardisierter und frei verfügbarer Datensatz noch ein einheitliches und allgemein akzeptiertes Evaluationsschema, sodass ein einfacher Vergleich unterschiedlicher Suchsysteme (externe Evaluation) möglich ist. Motiviert durch die Evaluationen im Information Retrieval, die sich im Rahmen der seit 1992 jährlich stattfindenden *Text REtrieval Conference* (TREC) etabliert haben, existiert zwar mit dem Benchathlon-Netzwerk (`http://www.benchathlon.net`) eine Initiative[18], auch in der inhaltsbasierten Bildersuche eine einheitliche Evaluationsumgebung zu etablieren, allerdings sind für seine Nutzung verschiedene technische Voraussetzungen zu erfüllen. Insbesondere erfordert diese Umgebung die Verwendung der *Multimedia Retrieval Markup Language* (MRML), was dazu führt, das ein Bildsuchsystem seine Struktur bezüglich dieser Anforderung anpassen muss.

Die Grundvoraussetzung für den Vergleich unterschiedlicher Bildsuchsysteme ist sicherlich, dass die zu vergleichenden Systeme auf derselben Bildmenge basieren. Obwohl sich mit dem Bilddatensatz von Corel (`http://www.corel.com`) eine Bildsammlung etabliert hat, ist diese so umfangreich, dass in den verschiedenen Arbeiten häufig unterschiedliche Teilmengen verwendet werden (vgl. z.B. [Wan01] oder [Car02]). Deshalb ist es sehr schwierig, die Performance unterschiedlicher Bildsuchsysteme miteinander zu vergleichen. Doch selbst die Verwendung einer identischen Bildmenge ist noch keine Garantie für den gültigen Vergleich verschiedener Bildsuchsysteme. Müller et al. [Mül02] demonstrieren eindrucksvoll, wie die Suchleistung eines Bilddatenbanksystems lediglich durch eine geeignete Auswahl der Anfragebilder verbessert werden kann.

Des Weiteren erfordert die Evaluation die Formulierung eines Bewertungsschemas sowie die Konstruktion einer Referenzgruppierung, die eine qualitative Beurteilung eines Suchergebnisses ermöglicht. Dabei ist jedoch zu beachten, dass diese Prozesse stark subjektiv geprägt sind. Die entsprechenden Schemata und Gruppierungen können daher in der Regel nur bedingt als allgemeingültig erklärt werden.

Unter Berücksichtigung der bisherigen Ausführungen kann festgehalten werden, dass der Vergleich von unterschiedlichen Bilddatenbanksystemen nur dann sinnvoll ist,

[18] Neben dem Benchathlon-Projekt existieren zwar noch weitere Aktivitäten einen Standard für die Evaluation in der inhaltsbasierten Bildersuche zu formulieren, allerdings wird an dieser Stelle nicht weiter auf diese Projekte eingegangen. Eine Auflistung der Aktivitäten findet sich auf den Webseiten des Benchathlon-Netzwerkes.

wenn die Evaluationsexperimente identisch durchgeführt werden und auf derselben Bildmenge basieren. In dieser Arbeit wird deshalb auf eine externe Evaluation verzichtet. Stattdessen werden die verschiedenen Suchansätze sowie die unterschiedlichen Lernmechanismen des INDI Systems experimentell verglichen (interne Evaluation). Welche Richtlinien für den Entwurf des benötigten Evaluationsschemas zu beachten sind, wird in den folgenden Abschnitten kurz erläutert.

Zunächst muss eine geeignete Bildsammlung ausgewählt werden. Eng verknüpft mit dieser Auswahl ist die Spezifikation eines sogenannten *Groundtruth* (dt. etwa „Grundwahrheit“). Darunter wird für eine Bildmenge eine Zuordnung der Bilder zu einer oder mehreren semantischen Kategorien verstanden. Bilder, die derselben Kategorie zugeordnet sind, werden als ähnlich betrachtet. Somit existiert eine unter semantischen Aspekten erzeugte Referenzgruppierung, die als Grundlage der qualitativen Bewertung eines Suchergebnisses dient.

Die Bewertung wiederum erfordert die Definition von Qualitätsmaßen. In der inhaltsbasierten Bildersuche haben sich die aus dem Information Retrieval stammenden Maße *Precision* (*P*) und *Recall* (*R*) etabliert, die häufig in „Precision vs. Recall“-Graphen dargestellt werden. Sie sind für eine betrachtete Ergebnismenge von L Bildern wie folgt definiert:

$$P(L) = \frac{\text{card}\{\text{gefunden} \cap \text{relevant}\}}{\text{card}\{\text{gefunden}\}} = \frac{L_{rel}}{L}$$

und

$$R(L) = \frac{\text{card}\{\text{gefunden} \cap \text{relevant}\}}{\text{card}\{\text{insgesamt relevant}\}} = \frac{L_{rel}}{N_{rel}}$$

Der Precision-Wert $P(L)$ beschreibt das Verhältnis zwischen der Anzahl L_{rel} der gefundenen relevanten Bilder zu der Anzahl L der gefundenen Bilder. Dagegen ist der Recall-Wert $R(L)$ ein relatives Maß für die Anzahl L_{rel} der gefundenen relevanten Bilder zu der Menge N_{rel} der insgesamt in der Datenbank vorhandenen relevanten Bilder. Beide Maße können ein Maximum von Eins ($P_{\max}(L) = 1.0$ bzw. $R_{\max}(L) = 1.0$) nicht überschreiten. Der maximale Precision-Wert signalisiert, dass für die aktuelle Kategoriesuche alle gefundenen Bilder relevante Bilder sind. Analog dazu signalisiert der maximale Recall-Wert, dass alle relevanten Bilder der Datenbank gefunden wurden. Müller et al. [Mül02] stellen in ihrer Arbeit weitere Maße vor, die ebenfalls die qualitative Bewertung der Leistungsfähigkeit eines Bildsuchsystems ermöglichen. Die meisten von ihnen basieren auf den soeben beschriebenen Qualitätsmaßen:

- $rel_1, \overline{Rank}, \widetilde{Rank}$: rel_1 bezeichnet den Rang, an dem das erste relevante Bild positioniert ist, wobei das initiale Beispielbild nicht berücksichtigt wird. $\overline{Rank}$ und $\widetilde{Rank}$ sind der Durchschnitts- bzw. normierte Durchschnittsrang aller relevanten Bilder, wobei das normierte Qualitätsmaß nach folgender Vorschrift berechnet wird:

$$\widetilde{Rank} = \frac{1}{NN_{\text{rel}}} \left(\sum_{i=1}^{N_{\text{rel}}} R_i - \frac{N_{\text{rel}}(N_{rel}+1)}{2}) \right),$$

R_i repräsentiert den Rang, an dem das *i*-te relevante Bild positioniert ist. N bezeichnet die Anzahl der gespeicherten Bilder und N_{rel} symbolisiert die Anzahl der relevanten Bilder, die für eine gegebene Anfrage in der Datenbank existieren.

- $P(20)$, $P(50)$ und $P(N_{rel})$: Precision nach 20, 50 und N_{rel} Bildern
- $R_P(0.5)$: Recall an der Stelle, an der der Precision-Wert 0.5 erreicht
- $R(100)$: Recall nach 100 Bildern

Im Gegensatz zur einfachen, einstufigen Bildersuche erfordert der interaktive Suchprozess eine entsprechend der Suchintention gegebene Bewertung der Ergebnisbilder. Für die Art und Weise, wie diese Bewertung im Rahmen der Evaluation gegeben wird, existieren zwei verschiedene Ansätze:

Manuelle Bewertung: Die manuelle Bewertung erfordert die Unterstützung durch eine Testperson, die entsprechend ihrer Suchintention die Bilder eines Iterationsergebnisses bewertet. Damit die Ergebnisse repräsentativ sind, reicht es jedoch nicht aus, nur ein Experiment mit einer Testperson durchzuführen, sondern es sind verschiedene Experimente erforderlich, die mit unterschiedlichen Probanden durchgeführt werden (vgl. z.B. [Käs03a] oder [Pfe06]).[19] Dies ist zum einen recht aufwendig und zum anderen sind die Experimente in der Regel nur schwer zu reproduzieren, da aufgrund der subjektiven visuellen Wahrnehmung die Bewertungskriterien eines Anwenders durchaus variieren können. Dieses individuelle Verhalten kann sowohl innerhalb eines Suchprozesses als auch bei der Wiederholung einer Bildersuche auftreten.

Automatische Bewertung: Bei der automatischen Bewertung dient der spezifizierte Groundtruth als Grundlage. Dabei werden diejenigen Bilder als relevant bewertet, die entsprechend der Referenzgruppierung zu derselben Bildklasse wie das Beispielbild gehören. Alle anderen Bilder der Ergebnismenge bleiben entweder unbewertet oder werden als nicht-relevant klassifiziert. Diese Variante hat den Vorteil, dass die experimentellen Ergebnisse reproduzierbar sind, da der komplette Suchprozess automatisiert ist. Ein Nachteil ist jedoch, dass zur Konstruktion der benötigten Referenzgruppierung die Bilder kategorisiert werden müssen. Da diese Einteilung der gespeicherten Bilder in Bildkategorien allerdings einen subjektiv geprägten Prozess darstellt, ist dieser Vorgang personenabhängig.

Des Weiteren erfordert der Entwurf eines Evaluationsschemas die Festlegung auf ein Anfrageparadigma (Ziel- oder Kategoriesuche) sowie die damit verbundene Spezifikation der entsprechenden Ziel- bzw. Beispielbilder.

[19] Auf die statistischen Besonderheiten, die bei personenbasierten Experimenten beachtet werden müssen, soll an dieser Stelle nicht weiter eingegangen werden. Nähere Informationen dazu finden sich in den Arbeiten von Käster et al. [Käs03a] und Pfeiffer [Pfe06].

 Autorennen	 Blumen	 Heißluftballons	 Tiere	 Golf
 Winterszenen	 Unter Wasser	 Sonnenuntergang	 San Francisco	 New York

Abb. 4.11: Bildkategorien der Evaluationsdatenbank

4.4.3 Evaluationsschema

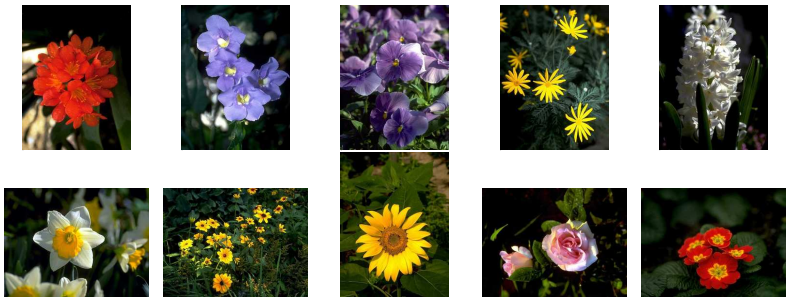
Die Grundlage der Evaluationsexperimente bildet eine Teilmenge der Fotokollektion der „ArtExplosion® 600000 Images“ Bildsammlung der Nova Development Corporation (`http://www.novadevelopment.com`). Dieser Abschnitt umfasst ca. 100000 Bilder, die in unterschiedliche semantische Kategorien eingeteilt sind, z.B. Automobile, Industrie, Landschaften oder Wassersport. Die Evaluationsdatenbank besteht aus 1250 Bildern der in Abbildung 4.11 dargestellten zehn Bilddomänen (125 Bilder pro Domäne). Für die Spezifikation eines Groundtruth wird auf eine manuelle nachträgliche Kategorisierung der Bilder verzichtet und die ursprüngliche Einteilung in semantische Klassen übernommen. Jedes Bild der Datenbank ist somit einer Kategorie zugeordnet.

Für die experimentellen Untersuchungen wird das Anfrageparadigma der Kategoriesuche (vgl. Abschnitt 2.1) gewählt. Dabei wird auf der Grundlage des Query-By-Example Ansatzes versucht, Bilder einer semantischen Kategorie zu finden. Die für die Durchführung der Experimente notwendigen Beispielbilder werden aus drei verschiedenen Bildkategorien ausgewählt: Autorennen, Blumen und Golf. Bei der Auswahl der Bilder wird die Zielsetzung verfolgt, möglichst repräsentative Beispiele für die drei Bildklassen zu selektieren. Die Beispielbilder sollten daher weder „Ausreißer“[20] der entsprechenden Bilddomäne darstellen noch sollten sie sich untereinander zu stark ähneln. Abbildung 4.12 gibt einen Überblick über die dreißig Bilder umfassende Beispielmenge, deren Elemente sich gleichmäßig auf die drei Beispieldomänen verteilen.

Die Evaluationsprozedur ist für alle Experimente identisch und ermöglicht die automatische Durchführung der verschiedenen Beispielsuchen. So lassen sich einerseits die resultierenden Ergebnisse reproduzieren und letztendlich besser analysieren, und andererseits ist die Automatisierung eine wichtige Voraussetzung für die einfache Untersuchung der unterschiedlichen Systemkonfigurationen und Adaptionsstrategien. Der

[20] Einen Ausreißer kennzeichnet, dass er zwar aus semantischen Aspekten zur korrespondierenden Bildkategorie gehört, sich aber ansonsten visuell zu stark von den übrigen Bildern der Kategorie unterscheidet.

(a) Autorennen



(b) Blumen

(c) Golf

Abb. 4.12: Beispielbilder der drei Bilddomänen

für die Experimente elementare Suchprozess basiert auf dem initial selektierten Beispielbild und umfasst acht Suchschritte. Die Suchschritte bestehen aus einem initialen Auswahlschritt zur Selektion des Beispielbildes sowie aus sieben Bewertungsschritten. In jedem Bewertungsschritt werden in Abhängigkeit vom spezifizierten Groundtruth alle Bilder, die zu derselben semantischen Klasse wie das Beispielbild gehören als relevant (+) bezeichnet. Dementsprechend werden alle Bilder der Ergebnismenge, die diese Bedingung nicht erfüllen als nicht-relevant (-) klassifiziert. Es wird somit ausschließlich eine binäre Klassifikation verwendet. Eine feinere Abstufung ist in einem

automatischen Evaluationsschema nur schwer zu erzielen, da in dem zu spezifizierenden Groundtruth nicht nur zwischen den verschiedenen Bildkategorien unterschieden werden muss, sondern zusätzlich auch innerhalb der Bildklassen eine Gruppierung der Bilder benötigt wird. Da die Bildkategorisierung ohnehin ein stark subjektiv geprägter Prozess ist, wird auf eine derartige Vorgehensweise verzichtet.

Die in jedem Suchschritt präsentierte Ergebnismenge umfasst 30 Bilder. Diese Auswahl wird von der Zielsetzung motiviert, ein möglichst praxisnahes und realistisches Suchszenario zu konstruieren. Dabei ist zu beachten, dass die Ergebnismenge überschaubar sein sollte und ein Benutzer einfach in ihr navigieren kann. In der Praxis hat sich gezeigt, dass eine Ergebnismenge von ca. 30 Bildern diese Anforderungen erfüllt (vgl. [Käs03a] oder [Pfe06]). Alternativ wird auch das Systemverhalten für eine Ergebnismenge von 45 Bildern untersucht. Dabei ist jedoch zu beachten, dass in einem automatischen Evaluationsszenario mit der Größe der Ergebnismenge auch die Anzahl der bewerteten Bilder ansteigt. Diese Tatsache ist zwar für den Lernprozess des Bildsuchsystems vorteilhaft, entspricht allerdings nicht den bereits genannten Anforderungen.

Zur qualitativen Bewertung eines Suchergebnisses dient der Precision-Wert. Auf eine zusätzliche Berücksichtigung des Recall-Wertes wird verzichtet, da dieser in einem inhaltsbasierten Bildsuchsystem gewöhnlich sehr gering und weniger aussagekräftig ist (vgl. [Zho03b, S. 115]). Die Berechnung des Precision-Wertes $P(L)$ einer Ergebnismenge vom Umfang L erfolgt für jede Iteration der Beispielsuchen, sodass für jede der dreißig Bildersuchen eine Menge von acht Qualitätsmaßen existiert. Um eine kompakte Repräsentation zu erzielen, wird für jeden Suchschritt ein durchschnittlicher Precision-Wert berechnet. Die Berechnung erfolgt sowohl separat für jede der drei betrachteten Bildkategorien als auch für die gesamten Bildersuchen. Die gemittelten Precision-Werte $\overline{P}(L)$ können schließlich für jeden Suchschritt in ein Precision/Bewertungsschritt-Diagramm eingetragen werden. Der resultierende Kurvenverlauf repräsentiert den Lernvorgang des Bildsuchsystems und ermöglicht dementsprechend eine qualitative Beurteilung der in Abschnitt 4.3 vorgestellten Verfahren zur Systemadaption. Je größer die Precision-Werte sind und je schneller sie mit zunehmender Anzahl der Bewertungsschritte ansteigen, desto besser lernt das System.

4.4.4 Experimentelle Untersuchungen und Ergebnisse

In diesem Abschnitt werden die experimentellen Untersuchungen beschrieben, die zur Evaluation des Systemlernens des Bildsuchsystems INDI durchgeführt wurden. Die Entwicklung der Experimente basiert auf den folgenden Fragestellungen:

1. Wie unterscheiden sich die Suchergebnisse, wenn statt einer kompletten lediglich eine diagonale Gewichtsmatrix zur Abstandsberechnung verwendet wird?

2. Inwieweit können negative Beispiele zum Systemlernen beitragen?
3. Welche Strategie sollte zur Adaption der Repräsentantengewichte gewählt werden und kann durch sie eine Leistungssteigerung erzielt werden?
4. Können mit separaten Bildrepräsentanten bessere Suchergebnisse erzielt werden als mit einem kombinierten Bildrepräsentanten?
5. Lässt sich die Systemleistung durch Regularisierung und Co-Training steigern?

Neben diesen Fragestellungen wurde untersucht, inwieweit sich die Ergebnisse des in Abschnitt 4.2 beschriebenen distanzbasierten Suchprozesses von denen des rangbasierten Suchprozesses unterscheiden. Ein Vorteil des distanzbasierten Ansatzes ist, dass die Abstände innerhalb eines Merkmalsraumes erhalten bleiben, während sie bei der rangbasierten Lösung, durch die Abbildung der Distanzwerte auf die korrespondierenden Ränge der Teilergebnisliste, verloren gehen. Ein Nachteil des distanzbasierten Verfahrens ist allerdings der, dass die Distanzen in den verschiedenen Merkmalsräumen eine unterschiedliche Dynamik besitzen können und deshalb vor deren Linearkombination eine Normierung erforderlich ist. Im Gegensatz dazu kann bei der rangbasierten Variante auf einen solchen Normierungsschritt verzichtet werden. Die folgenden Experimente wurden sowohl für den distanz- als auch für den rangbasierten Suchprozess durchgeführt, sodass beide Ansätze miteinander verglichen werden können.

Die Anfrageverfeinerung auf der Grundlage der Benutzerbewertungen erfolgte in allen Experimenten nach dem in Abschnitt 4.3.1 vorgestellten Optimierungsansatz. Als Standardabstandsmaß wurde der quadrierte generalisierte euklidische Abstand verwendet, dessen Gewichtsmatrix wie beschrieben auf der Grundlage der Kovarianzmatrix der Trainingsbeispiele berechnet wird. Eine Beschränkung auf diagonale Kovarianzen erfolgt nur dann, wenn die ursprüngliche Kovarianzmatrix singulär ist und daher eine alternative Verarbeitung notwendig ist, um einen Suchprozess fortsetzen zu können. In den Ausführungen der experimentellen Untersuchungen, in denen ein alternatives Abstandsmaß verwendet wurde, wird dies explizit betont. Das in Abschnitt 4.3.2 vorgestellte heuristische Verfahren zur Adaption der Repräsentantengewichte erfordert die Spezifikation des Lernratenparameters α. Für diesen wurde in allen Experimenten ein Wert von $\alpha = 0.3$ verwendet.

Für die Untersuchung der formulierten Fragestellungen wurde zunächst das Normierungsverfahren bestimmt, das sich in dem hier betrachteten Evaluationsszenario am besten für den distanzbasierten Ansatz eignet.

Experiment A: Vergleich von Verfahren zur Distanznormierung

Wie in Abschnitt 4.2.1 beschrieben wurde, ist die distanzbasierte Bildersuche durch das sukzessive Zusammenfassen der repräsentantenabhängigen Distanzwerte gekenn-

zeichnet. Abhängig von der jeweiligen Metrik, der Dimension der Merkmalsvektoren und deren Verteilung im Merkmalsraum kann es dabei vorkommen, dass die Abstände der Bildcharakteristika in den verschiedenen Merkmalsräumen eine unterschiedliche Dynamik besitzen. Ist dies der Fall, so werden die Gesamtabstände $D(\mathcal{O}_k, \mathcal{Q}) = \sum_{j=1}^{J} v_j D_j(\mathcal{O}_k, \mathcal{Q})$ der gespeicherten Bildobjekte $\{\mathcal{O}_k | k = 1, 2, \ldots, K\}$ zum Anfrageobjekt $\mathcal{Q}$ von den Merkmalsräumen dominiert, in denen die Abstände eine große Dynamik besitzen. Um dies zu verhindern, müssen die Distanzen $D_j(\mathcal{O}_k, \mathcal{Q})$ eines Bildobjekts $\mathcal{O}_k$ vor ihrer Linearkombination normiert werden, sodass sie vergleichbar werden. Das Ziel der ersten experimentellen Untersuchungen ist es daher, festzustellen, welches Normierungsverfahren für den distanzbasierten Suchprozess des INDI Systems am besten geeignet ist. Die betrachteten Normierungsansätze können in zwei Kategorien eingeteilt werden. Zum einen sind dies Verfahren, die abhängig vom aktuellen Suchschritt und den damit verbundenen aktuell auftretenden Abstandswerten die Normierungsparameter zur Laufzeit (online) berechnen. Zum anderen sind dies Ansätze, die die Normierungsparameter vorab (offline) berechnen. In den experimentellen Untersuchungen wurden die folgenden Ansätze genauer untersucht:

1. **KeineNormierung**: Die unterschiedlichen Distanzwerte $D_j(\mathcal{O}_k, \mathcal{Q})$ eines Bildobjekts $\mathcal{O}_k$ werden ohne Normierung zu einem Gesamtdistanzwert $D(\mathcal{O}_k, \mathcal{Q})$ kombiniert.

2. **Dimension**: Da die Abstandsberechnungen in den verschiedenen Merkmalsräumen zwar auf demselben Abstandsmaß basieren, aber die korrespondierenden Merkmalsvektoren sich in ihrer Dimension unterscheiden, werden die Abstandswerte $D_j(\mathcal{O}_k, \mathcal{Q})$ durch die Dimension des entsprechenden Merkmalsraumes gewichtet:
$$\tilde{D}_j(\mathcal{O}_k, \mathcal{Q}) = D_j(\mathcal{O}_k, \mathcal{Q})/N_j,$$
wobei N_j die Dimension des j-ten Repräsentanten bezeichnet.

3. **Varianz**: In jedem Suchschritt werden die verschiedenen Distanzwerte $\{D_j(\mathcal{O}_k, \mathcal{Q}) | k = 1, 2, \ldots, K\}$ eines Merkmalsraumes als Messreihe interpretiert. Für diese werden Mittelwert μ_j und Standardabweichung σ_j berechnet, die die Grundlage des Normierungsschritts bilden. Ausgehend von der Annahme, dass die Distanzen der Bildbeschreibungen in den jeweiligen Merkmalsräumen normalverteilt sind, wird jeder Distanzwert nach folgender Berechnungsvorschrift in den Wertebereich $[0, 1]$ abgebildet[21]:
$$\tilde{D}_j(\mathcal{O}_k, \mathcal{Q}) = \frac{(D_j(\mathcal{O}_k, \mathcal{Q}) - \mu_j)/(3\sigma_j) + 1}{2}$$

[21] Die 1% der normierten Abstandswerte, die nicht in das Intervall $[0, 1]$ fallen, werden auf die benachbarte Intervallgrenze abgebildet.

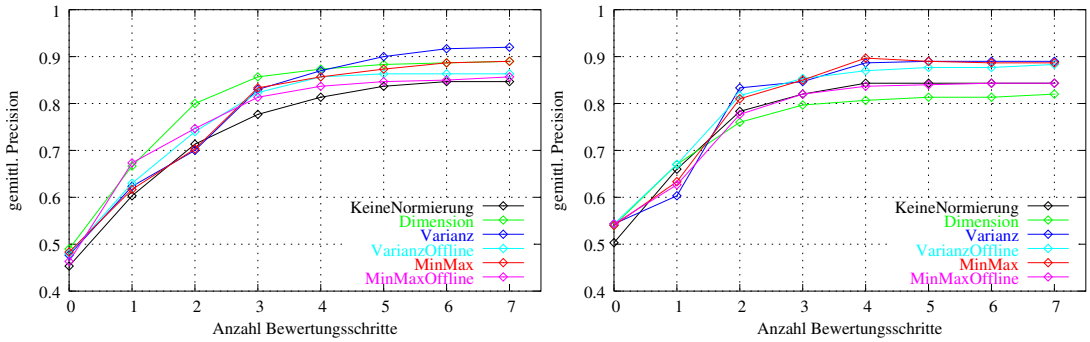
4. **VarianzOffline**: Die Normierung erfolgt analog zu der unter 3. vorgestellten Berechnungsvorschrift. Die Berechnung der notwendigen Normierungsparameter μ_j und σ_j erfolgt jedoch nicht zur Laufzeit, sondern offline im Rahmen der Datenbankinitialisierung. Dabei werden analog zu Ortega et al. [Ort97] in jedem Merkmalsraum die Abstände eines Bildobjekts $\mathcal{O}_k$ zu allen anderen Objekten $\mathcal{O}_l$ der Datenbank berechnet, mit $l, k = 1, 2, \ldots, K$ und $l \neq k$. Als Abstandsmaß wird der quadrierte euklidische Abstand verwendet, der einem quadrierten generalisierten euklidischen Abstand entspricht, dessen Gewichtsmatrix gleich der Einheitsmatrix ist und somit ebenfalls die in Abschnitt 4.3.1 formulierte Nebenbedingung $\det(\mathbf{W}) = 1$ erfüllt. Die resultierenden $\frac{K \times (K-1)}{2}$ Abstandswerte bilden schließlich die Messreihe zur Berechnung der Normierungsparameter.

5. **MinMax**: In jedem Suchschritt werden in den verschiedenen Merkmalsräumen der minimale Distanzwert $D_j^{\min}$ und der maximale Distanzwert $D_j^{\max}$ bestimmt. Darauf aufbauend erfolgt die Normierung der Distanzwerte auf das Intervall $[0, 1]$ nach folgender Berechnungsvorschrift:

$$\tilde{D}_j(\mathcal{O}_k, \mathcal{Q}) = \frac{D_j(\mathcal{O}_k, \mathcal{Q}) - D_j^{\min}}{D_j^{\max} - D_j^{\min}}$$

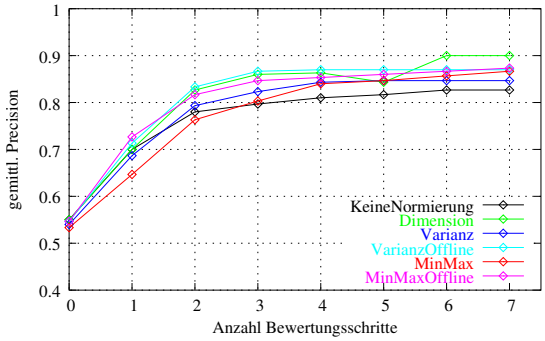
6. **MinMaxOffline**: Dieser Ansatz entspricht der unter 5. formulierten Berechnungsvorschrift. Allerdings werden die Normierungsparameter $D_j^{\min}$ und $D_j^{\max}$ nicht auf der Grundlage der aktuell auftretenden Abstandswerte bestimmt, sondern ähnlich wie unter 4. auf der Grundlage der offline berechneten Abstandswerte. Da die normierten Distanzwerte allerdings in Abhängigkeit von den jeweiligen Normierungsparametern kleiner als Null werden können, werden die Distanzwerte um den Betrag des kleinsten normierten Wertes verschoben, sodass $\tilde{D}_j^{\min}(\mathcal{O}_k, \mathcal{Q}) = 0$ gilt.

In allen Beispielsuchen wurde der in Abschnitt 4.3.2 formulierte heuristische Ansatz zur Adaption der Repräsentantengewichte verwendet. Dementsprechend wurden neben positiven Trainingsbeispielen auch negative Elemente zum Lernen verwendet. Abbildung 4.13 stellt die Ergebnisse der verschiedenen experimentellen Untersuchungen bei einer Ergebnismenge von 30 Bildern dar. Die Kurvenverläufe der unterschiedlichen Messreihen veranschaulichen, dass lediglich durch eine geeignete Wahl des Normierungsverfahrens bessere Suchergebnisse erzielt werden können. Die in Abbildung 4.13(a) dargestellten Resultate zeigen, dass für die Beispielsuchen der Kategorie „Autorennen“ mit der varianzbasierten Normierung („Varianz“) nach sieben Bewertungsschritten gegenüber der schlechtesten Strategie („KeineNormierung“) eine relative Verbesserung von 8% erzielt werden kann. Unter Berücksichtigung aller Beispielsuchen (vgl. Abbildung 4.13(d)) sind dies immer noch knapp 6%.

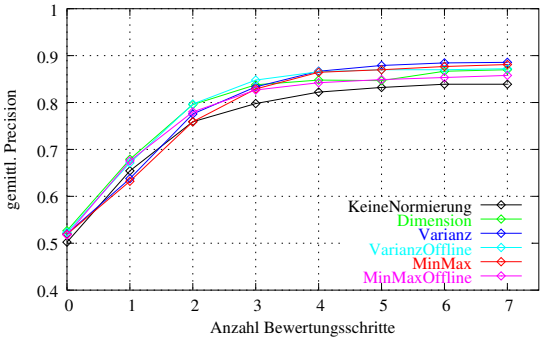


(a) Autorennen

(b) Blumen



(c) Golf



(d) Gesamt

Abb. 4.13: Vergleich der unterschiedlichen Verfahren zur Normierung der Distanzwerte bei einer Ergebnismenge von 30 Bildern.

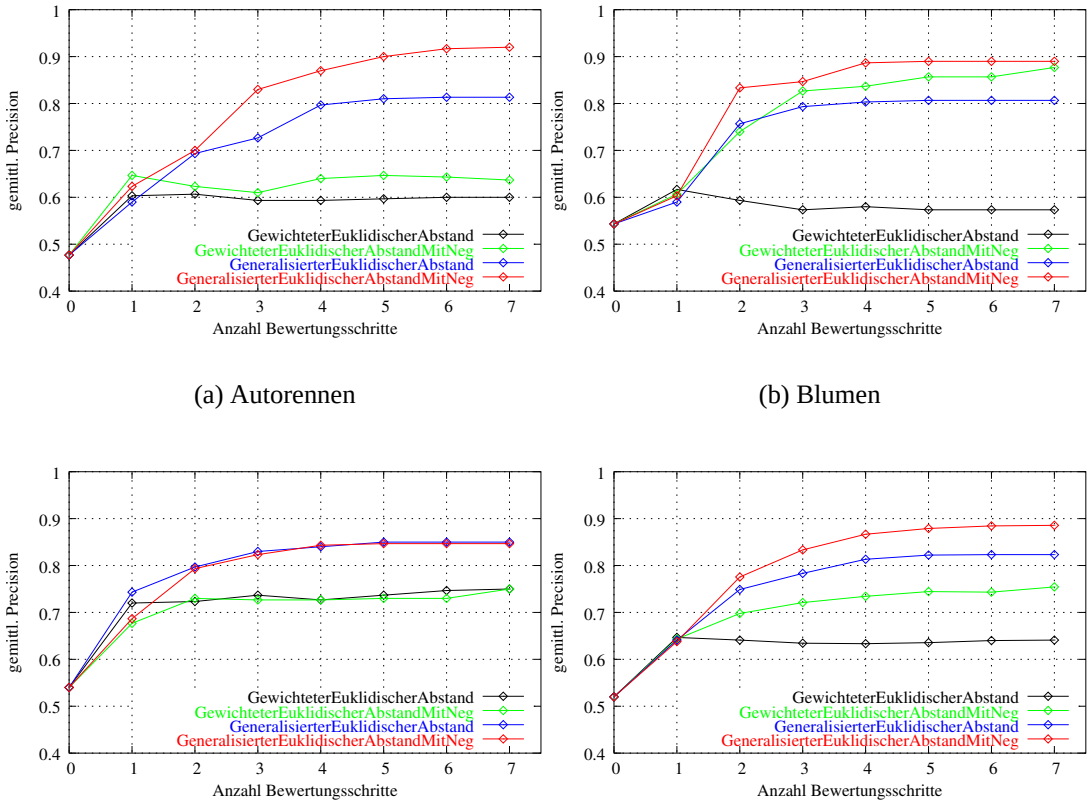
Insgesamt kann beobachtet werden, dass die Ergebnisse der verschiedenen Normierungsverfahren in den verschiedenen Diagrammen variieren. Deshalb kann kein Verfahren identifiziert werden, das für alle Beispielsuchen deutlich bessere Ergebnisse liefert als die übrigen Verfahren. Auffällig ist allerdings, dass eine Verzicht auf eine Normierung der Distanzwerte häufig auch zu schlechteren Resultaten führt. Eine Ausnahme bildet lediglich die Messreihe für die Blumenbilder in Abbildung 4.13(b). Dort liefert die dimensionsbasierte Gewichtung der Distanzwerte („Dimension“) schlechtere Ergebnisse. Werden neben den in Abbildung 4.13(a) bis 4.13(d) dargestellten Kurven auch die in Anhang D präsentierten Ergebnisse für eine Ergebnismenge von 45 Bildern (vgl. Abbildung D.1) berücksichtigt, so wird deutlich, dass von allen Verfahren der varianzbasierte Ansatz („Varianz“) die geringsten Schwankungen aufweist. Da dieser Ansatz außerdem relativ robust gegenüber Ausreißern ist, wird er für die folgenden Experimente als Standardnormierungsverfahren der distanzbasierten Bildersuchen verwendet.

Experiment B: Vergleich von Bildersuchen mit gewichtetem und generalisiertem euklidischen Abstand sowie mit und ohne negativen Beispielen

Nachdem mit der Entscheidung für das varianzbasierte Normierungsverfahren die Grundlage für die distanzbasierten Beispielsuchen geschaffen wurde, ist es das Ziel der folgenden experimentellen Untersuchungen, die in Abschnitt 4.3 vorgestellten Verfahren zum Systemlernen genauer zu analysieren. In einem weiteren Experiment wurde daher zunächst untersucht, wie sich die Systemleistung verändert, wenn statt eines quadrierten generalisierten ein quadrierter gewichteter euklidischer Abstand zur Distanzberechnung verwendet wird. Ergänzend dazu wurde evaluiert, inwieweit negative Beispiele zum Systemlernen beitragen können.

Der Einsatz eines gewichteten euklidischen Abstandes entspricht der Beschränkung auf diagonale Kovarianzen (vgl. Abschnitt 4.3.1). Im Gegensatz dazu wird bei der Verwendung des generalisierten euklidischen Abstandes versucht, die komplette Kovarianzmatrix zu invertieren. Nur wenn dies aufgrund von numerischen Instabilitäten nicht gelingt, wird eine diagonale Kovarianzmatrix berechnet. Ebenso wie in den vorherigen Untersuchungen wurde der heuristische Ansatz zur Adaption der Repräsentantengewichte benutzt. Bei zwei der vier Messreihen wurde ausschließlich mit positiv bewerteten Trainingsbeispielen gelernt. Bei den übrigen experimentellen Untersuchungen wurden ergänzend zu den positiven Beispielen auch negativ bewertete Bilder zum Systemlernen verwendet. Dies ist in den verschiedenen Diagrammen jeweils durch den Zusatz „MitNeg“ gekennzeichnet. Die entsprechenden Ergebnisse der distanz- und rangbasierten Bildersuchen für eine Ergebnismenge von 30 Bildern sind in Abbildung 4.14 und 4.15 dargestellt.

In Abschnitt 4.3.1 wurde demonstriert, dass die Berechnung einer kompletten Gewichtsmatrix einer Transformation der Datenmenge mit anschließender Neugewichtung der Koordinatenachsen des neuen Merkmalsraumes entspricht. Im Gegensatz dazu kann durch den gewichteten euklidischen Abstand lediglich eine Neugewichtigung der Koordinatenachsen des ursprünglichen Vektorraumes erzielt werden. Es ist daher anzunehmen, dass sich die Systemleistung steigern lässt, wenn statt einer diagonalen eine komplette Gewichtsmatrix berechnet werden kann. Die Ergebnisse der experimentellen Untersuchungen belegen diese Hypothese. In fast allen Diagrammen können die besten Resultate mit den Beispielsuchen erzielt werden, die auf einem generalisierten Abstandsmaß basieren. Eine Ausnahme stellen lediglich die in Abbildung 4.14(b) dargestellten Ergebnisse dar. Diese veranschaulichen, dass für die Beispielbilder der Domäne „Blumen“ mit einem gewichteten Abstandsmaß und der Berücksichtigung negativer Trainingsbeispiele („GewichteterEuklidischerAbstandMitNeg“) bessere Resultate erzielt werden können als durch einen generalisierten euklidischen Abstand und die Beschränkung auf positive Beispiele („GeneralisierterEuklidischerAbstand“). Auffällig ist außerdem, dass nach dem ersten Bewertungsschritt in allen Graphen die Ergebnisse der unterschiedlichen Messreihen noch relativ nahe beieinander liegen. Erst nach dem zweiten Bewertungsschritt sind deutliche Unterschiede zwischen den



(a) Autorennen

(b) Blumen

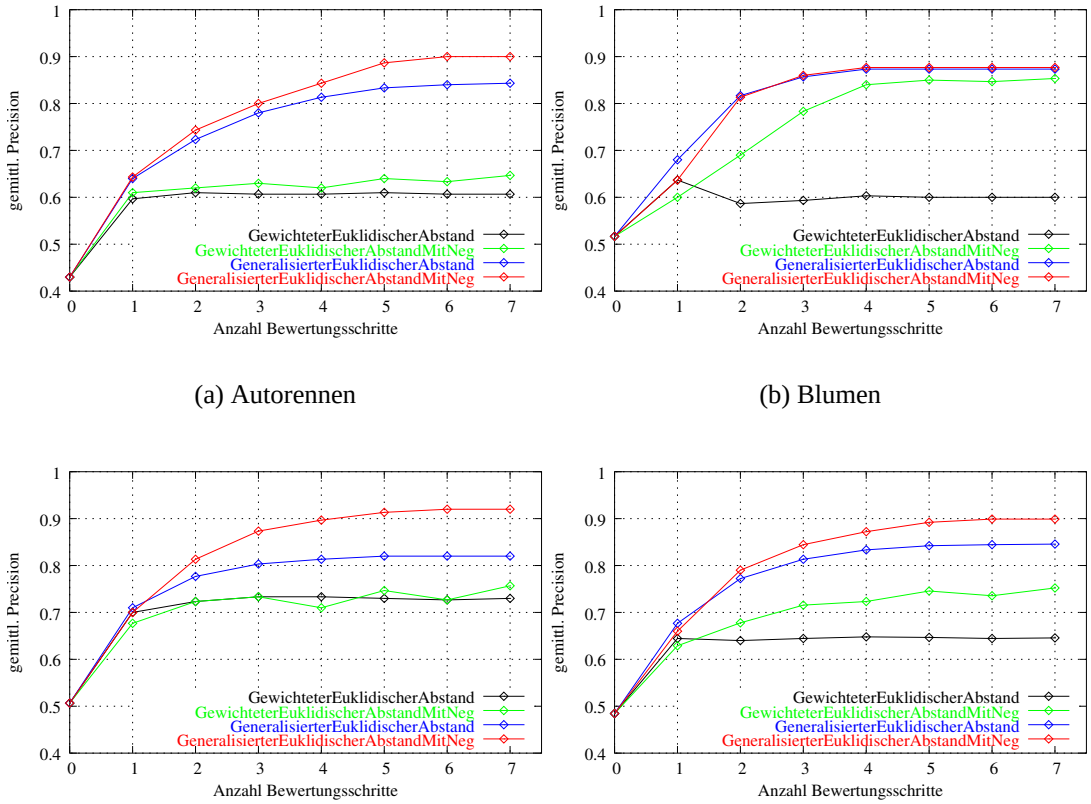
(c) Golf

(d) Gesamt

Abb. 4.14: Distanzbasierte Bildersuche mit gewichtetem und generalisiertem euklidischen Abstand mit und ohne negative Trainingsbeispiele bei einer Ergebnismenge von 30 Bildern.

Resultaten des gewichteten und generalisierten Abstandsmaßes zu beobachten. Der Grund hierfür ist, dass nach der ersten Bewertung der Ergebnisbilder die Kovarianzmatrix der Trainingsvektoren oftmals singulär ist. Daher reduziert sich der generalisierte Abstand zu einem gewichteten Maß. Erst mit ansteigender Iterationenanzahl werden ausreichend viele relevante Bilder gefunden, sodass die entsprechende Kovarianzmatrix der Stichprobenelemente invertierbar ist.

Des Weiteren demonstrieren die Ergebnisse, dass durch die Berücksichtigung der negativ bewerteten Bilder fast immer eine Verbesserung der Suchergebnisse erzielt wird. In den Fällen, in denen dies nicht möglich ist, verschlechtern sich die Ergebnisse zumindest nicht (vgl. Abbildung 4.14(c) und 4.15(b)). Wenn man die Ergebnisse des gewichteten Abstandsmaßes ohne negative Beispiele („GewichteterEuklidischerAbstand“) betrachtet, so fällt auf, dass die Kurven nach dem ersten Bewertungsschritt zwar ansteigen, aber danach näherungsweise auf gleichem gemittelten Precision-Niveau verbleiben bzw. sich der gemittelte Precision-Wert sogar ein wenig



(a) Autorennen (b) Blumen

(c) Golf (d) Gesamt

Abb. 4.15: Rangbasierte Bildersuche mit gewichtetem und generalisiertem euklidischen Abstand mit und ohne negative Trainingsbeispiele bei einer Ergebnismenge von 30 Bildern.

verschlechtert. Der Lernprozess ist daher auf die ersten Bewertungsschritte begrenzt und das System ist nicht in der Lage, auf der Basis des gewichteten Abstandsmaßes weitere relevante Bilder zu finden. Durch die Hinzunahme der negativ bewerteten Bilder können (zumindest teilweise) kleine, für die Beispielsuchen der Blumenbilder sogar sehr große Verbesserungen erzielt werden (vgl. Abbildung 4.14(b) und 4.15(b)). Der Grund für diese deutlichen Verbesserungen liegt in der Gewichtung der verwendeten Bildrepräsentanten. Durch die zusätzliche Berücksichtigung der negativen Beispiele können die Bildcharakteristika bestimmt werden, die sich nicht für die aktuelle Suchaufgabe eignen. Entsprechend der in Abschnitt 4.3.2 formulierten Berechnungsvorschrift werden diese Charakteristika schwach gewichtet oder gegebenfalls sogar deaktiviert. Die übrigen Bildrepräsentanten werden demzufolge stärker berücksichtigt. Für die Beispielsuchen der Blumenbilder scheint diese Strategie für das gewichtete Abstandsmaß besonders geeignet zu sein.

Als nächstes werden die Ergebnisse der gesamten Bildersuchen für den distanzbasierten Ansatz in Abbildung 4.14(d) genauer betrachtet. Die Kurven demonstrieren, dass nach sieben Bewertungsschritten für das gewichtete Abstandsmaß mit negativen Beispielen eine relative Verbesserung des durchschnittlichen Precision-Wertes von 17% erzielt werden kann. In absoluten Zahlen kann einer Steigerung von 0.11 erzielt werden. Dies bedeutet bei einer Ergebnismenge von 30 Bildern, dass im Durchschnitt weitere 3.3 Bilder der gesuchten Kategorie gefunden werden konnten. Für das generalisierte Maß kann durch die Berücksichtigung der negativen Beispiele eine relative Verbesserung von 9% erzielt werden. Obwohl lediglich mit positiven Trainingsbeispielen gelernt wird, kann schon alleine durch den Einsatz des generalisierten Abstandes („GeneralisierterEuklidischerAbstand“) im Vergleich zum gewichteten Abstand mit negativen Beispielen („GewichteterEuklidischerAbstandMitNeg“) eine relative Verbesserung von 9% erreicht werden. Noch größer wird die relative Verbesserung, wenn neben den positiven auch negative Elemente zum Systemlernen verwendet werden („GeneralisierterEuklidischerAbstandMitNeg“). Dann kann im Vergleich zum gewichteten Abstand mit negativen Beispielen eine relative Verbesserung von 19% erzielt werden, was bedeutet, dass hier im Durchschnitt weitere 4.2 Bilder der gesuchten Bildkategorie gefunden werden konnten. Diese Ergebnisse werden von den rangbasierten Resultaten in Abbildung 4.15(d) bestätigt. Wenn man jedoch die unterschiedlichen Diagramme beider Ansätze vergleicht, so kann festgestellt werden, dass die Kurven der rangbasierten Bildersuche ein wenig glatter verlaufen als die der distanzbasierten Beispielsuchen (vgl. z.B. Abbildung 4.14(a) und 4.15(a)). Teilweise ist sogar zu beobachten, dass die Ergebnisse einiger distanzbasierter Messreihen schlechter sind als die entsprechenden Resultate der rangbasierten Bildersuchen. Dies trifft z.B. für den generalisierten euklidischen Abstand mit negativen Beispielen in Abbildung 4.14(c) und 4.15(c) sowie für das generalisierte Abstandsmaß ohne negative Trainingselemente in Abbildung 4.14(b) und 4.15(b) zu.

Die in Anhang D dargestellten Resultate für eine Ergebnismenge von 45 Bildern bestätigen die in Abbildung 4.14 und 4.15 dargestellten Ergebnisse der verschiedenen Beispielsuchen. Bei einer genaueren Betrachtung der Diagramme in Abbildung D.2 und D.3 lässt sich außerdem feststellen, dass schon nach dem ersten Bewertungsschritt deutliche Unterschieden zwischen den gemittelten Precision-Werten der unterschiedlichen Messreihen zu beobachten sind. Das liegt daran, dass bei einer Menge von 45 Bildern nach einem Bewertungsschritt gewöhnlich mehr Trainingsbeispiele vorliegen als dies bei 30 Ergebnisbildern der Fall ist. Die größere Menge an negativen Trainingsbeispielen hat besonders auf die Beispielsuchen der Blumenbilder Einfluss. Sowohl bei der distanz- als auch bei der rangbasierten Bildersuche können die besten Suchergebnisse mit einem gewichteten euklidischen Abstand und der Integration negativer Beispiele („GewichteterEuklidischerAbstandMitNeg“) erzielt werden.

Experiment C: Vergleich der Verfahren zur Adaption der Repräsentantengewichte

Im letzten Experiment wurde demonstriert, dass die Systemleistung sowohl durch die Berechnung eines generalisierten statt eines gewichteten euklidischen Abstandes als auch mit negativen Trainingsbeispielen gesteigert werden kann. In dem hier beschriebenen Experiment wurde untersucht, welche Bedeutung die Adaption der Repräsentantengewichte für den iterativen Suchprozess besitzt. Dabei wurden nicht nur die verschiedenen in Abschnitt 4.3 vorgestellten Ansätze miteinander verglichen, sondern es wurde auch untersucht, wie sich das System verhält, wenn auf eine Adaption der Repräsentantengewichte verzichtet und für den kompletten Suchprozess eine gleichmäßige Gewichtung der verschiedenen Bildcharakteristika beibehalten wird. Zusätzlich wurden die verschiedenen Verfahren, die alle auf dem hierarchischen Bildvergleich basieren, mit einem Suchprozess verglichen, der im Gegensatz dazu auf der flachen Modellierung basiert. Somit lässt sich analysieren, wie sich die Suchleistung verändert, wenn statt der separaten Repräsentation der verschiedenen Bildcharakteristika ein kombinierter Bildrepräsentant verwendet wird (vgl. Abschnitt 4.1). In den experimentellen Untersuchungen wurden die folgenden Ansätze unterschieden:

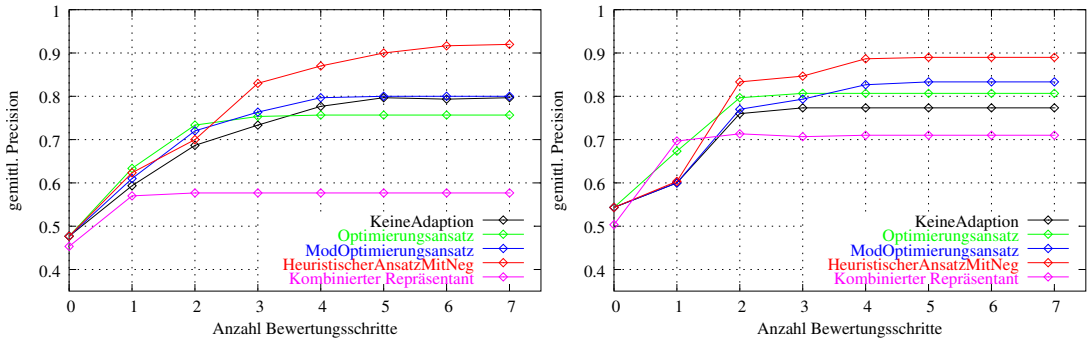
1. **KeineAdaption**: Bei dieser Variante wird auf eine Adaption der verschiedenen Repräsentantengewichte verzichtet. Stattdessen werden die initialen Gewichte für den kompletten Suchprozess konstant gehalten, sodass jeder Bildrepräsentant gleich gewichtet wird.

2. **Optimierungsansatz**: Adaption der Repräsentantengewichte entsprechend des in Abschnitt 4.3.1 Optimierungsansatzes. Dabei werden die Repräsentanten am stärksten gewichtet, in deren Merkmalsraum die Deskriptoren der relevanten Beispielbilder zu dem jeweiligen idealen Anfragevektor den geringsten Abstand besitzen. Da dieser Adaptionsschritt auf dem Vergleich der Distanzwerte der unterschiedlichen Merkmalsräume basiert, müssen diese vergleichbar sein. Deshalb wird jeder Distanzwert entsprechend der Dimension des korrespondierenden Merkmalsraumes gewichtet.

3. **ModOptimierungsansatz**: Modifizierte Variante des vorherigen Verfahrens, die statt der Distanzen der Trainingsbeispiele, deren Ränge in den entsprechenden Teilergebnislisten zur Adaption der Repräsentantengewichte verwendet (vgl. Abschnitt 4.3.1).

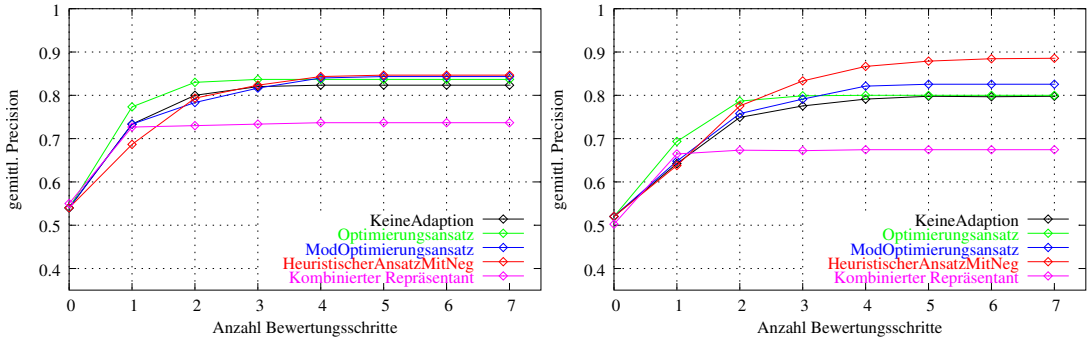
4. **HeuristischerAnsatzMitNeg**: Heuristische Systemadaption, die neben positiv klassifizierten Bildern auch negativ bewertete Beispiele innerhalb des Lernschritts berücksichtigt (vgl. Abschnitt 4.3.2).

5. **Kombinierter Repräsentant**: Um eine Aussage über die Performance der in Abschnitt 4.2 vorgestellten hierarchischen Modellierung treffen zu können, wurde eine Experimentreihe mit kombinierten Bildrepräsentanten durchgeführt. Der



(a) Autorennen

(b) Blumen



(c) Golf

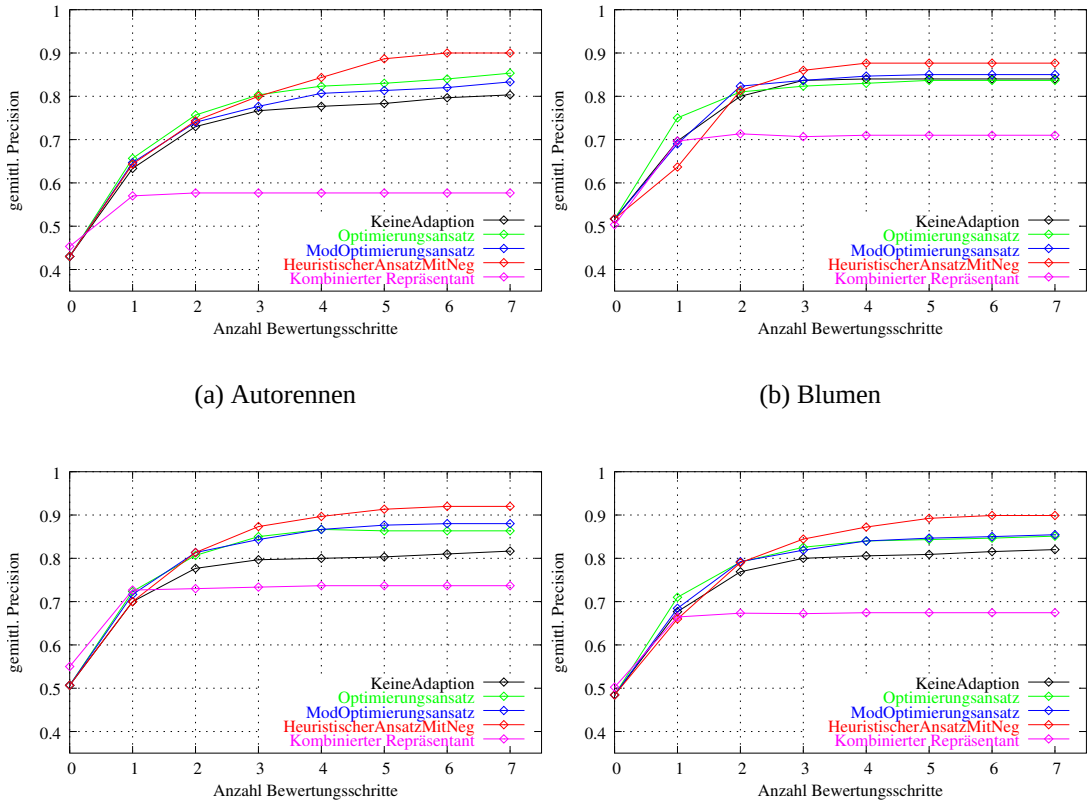
(d) Gesamt

Abb. 4.16: Vergleich der Verfahren zur Adaption der Repräsentantengewichte bei einer Ergebnismenge von 30 Bildern (distanzbasiert).

Lernprozess beschränkt sich demnach auf die Verfeinerung der Anfragevektoren und die Adaption der Gewichtsmatrix.

Die in Abbildung 4.16 und 4.17 dargestellten Ergebnisse der verschiedenen Messreihen demonstrieren die Überlegenheit des hierarchischen Bildvergleichs gegenüber dem flachen Modell. In allen Diagrammen kann zwar ausgehend vom Relevance Feedback des Benutzers auch für den kombinierten Bildrepräsentanten eine Verbesserung der Suchergebnisse erzielt werden, allerdings ist diese hauptsächlich auf den ersten Bewertungsschritt begrenzt[22]. Danach sind, wenn überhaupt, nur noch marginale Veränderungen des durchschnittlichen Precision-Wertes zu beobachten. Dies liegt an der hohen Dimensionalität der Bildcharakteristika (durch die Verkettung der einzelnen Repräsentanten ergibt sich für jedes Bild ein 51-dimensionaler Bilddeskriptor).

[22] An dieser Stelle wird darauf hingewiesen, dass die Ergebnisse des kombinierten Bildrepräsentanten in den Diagrammen der distanz- und rangbasierten Beispielsuchen dieselben sind, da bei einer kombinierten Repräsentation weder Distanzen noch Ränge kombiniert werden müssen (vgl. Abschnitt 4.1).



(a) Autorennen (b) Blumen

(c) Golf (d) Gesamt

Abb. 4.17: Vergleich der Verfahren zur Adaption der Repräsentantengewichte bei einer Ergebnismenge von 30 Bildern (rangbasiert).

Aufgrund der hohen Anzahl an Vektorkomponenten müssen für die Berechnung der Gewichtsmatrix mehr Parameter geschätzt werden als dies in den separaten Merkmalsräumen notwendig ist. Bei zu wenig Trainingsbeispielen kann es deshalb passieren, dass die komplette Kovarianzmatrix der Stichprobe nicht invertierbar ist. Dementsprechend reduziert sich der generalisierte Abstand zu einem gewichteten Abstand, sodass der Lernprozess weniger performant ist. Die Diagramme belegen außerdem, dass die in Abschnitt 4.3.2 vorgestellte heuristische Adaption der Repräsentantengewichte auf der Grundlage der positiv und negativ bewerteten Beispielbilder bessere Suchergebnisse liefert als die übrigen Ansätze. Diese Beobachtung gilt sowohl für die distanz- als auch für die rangbasierte Bildersuche. Obwohl die Kurvenverläufe der Messreihen ohne Adaption der verschiedenen Repräsentantengewichte veranschaulichen, dass der größte Lernerfolg durch die Adaption der Gewichte der Repräsentantenebene erzielt werden kann, ist auch zu beobachten, dass mit einer zusätzlichen Adaption der Repräsentantengewichte in fast allen Diagrammen eine weitere Steigerung der Systemleistung erzielt werden kann. Eine Ausnahme bilden lediglich die distanzbasierten

Beispielsuchen der Kategorie „Autorennen“ in Abbildung 4.16(a) als auch die rangbasierten Beispielsuchen der Kategorie „Blumen“ in Abbildung 4.17(b). Im ersten Fall liefert der Optimierungsansatz nach sieben Bewertungsschritten leicht schlechtere Ergebnisse. Im letztgenannten Fall liefert er ähnliche durchschnittliche Precision-Werte wie die Messreihen, bei denen auf eine Adaption der Repräsentantengewichte verzichtet wurde.

Vergleicht man die Kurven des Optimierungsansatzes mit denen der modifizierten Variante („ModOptimierungsansatz“), so kann festgestellt werden, dass durch die Verwendung der Ränge statt Distanzen (vgl. Abschnitt 4.3.1) keine deutliche Verbesserung der Suchergebnisse erzielt werden konnte. Insgesamt ist zwar in der distanzbasierten Bildersuche eine leichte Steigerung gegenüber dem klassischen Optimierungsansatz zu erkennen (vgl. Abbildung 4.16(d)), allerdings liefern beide Verfahren für die rangbasierten Beispielsuchen (nahezu) identische Resultate (vgl. Abbildung 4.17(d)). Die in Anhang D dargestellten Resultate für eine Ergebnismenge von 45 Bildern bestätigen die bisher diskutierten Ergebnisse.

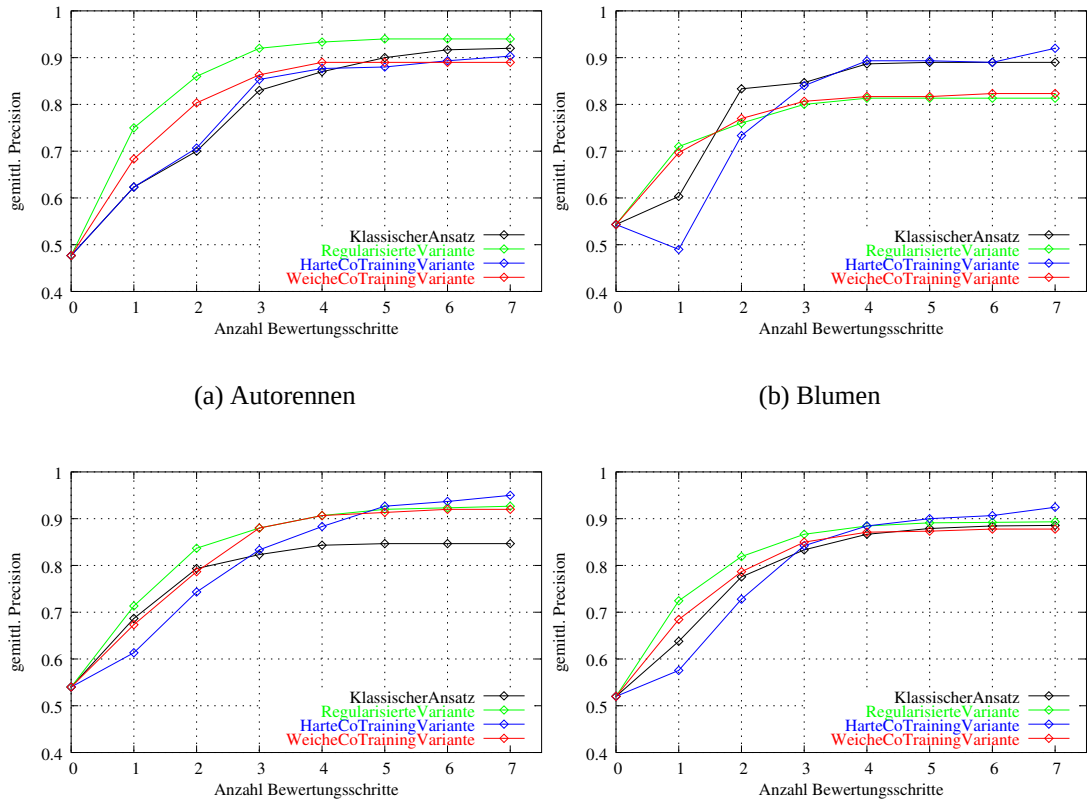
Experiment D: Analyse von Regularisierung und Co-Training

Ein gravierendes Problem der inhaltsbasierten Bildersuche und der in der vorliegenden Arbeit vorgestellten Adaptionsverfahren ist der geringe Umfang der klassifizierten Stichprobe. Die für die Bestimmung der Gewichtsmatrix notwendige Kovarianzmatrix kann auf der Grundlage einer derartig geringen Trainingsmenge in der Regel nur selten so bestimmt werden, dass sie invertierbar ist. Wie beschrieben hat dies zur Folge, dass lediglich eine diagonale Gewichtsmatrix berechnet wird und im kommenden Suchschritt ein gewichteter anstatt eines generalisierten Abstandes zur Ähnlichkeitsbestimmung verwendet wird. Die experimentellen Untersuchungen haben allerdings gezeigt, dass durch die Berechnung einer kompletten Gewichtsmatrix bessere Suchergebnisse erzielt werden können. Die Problematik der Matrixinvertierung tritt besonders in den ersten Suchschritten auf, in denen gewöhnlich noch nicht so viele Bewertungen erfolgt sind und dementsprechend relativ wenig Trainingsbeispiele vorliegen. Mit der Regularisierung nach Friedman [Fri89] wurde in Abschnitt 4.3.1 ein Verfahren vorgestellt, dass zur Lösung des Invertierungsproblems dient. Die in Abschnitt 4.3.3 beschriebene Kombination von überwachtem und unüberwachtem Lernen zielt darüber hinaus auch auf eine robustere Parameterschätzung der Kovarianzmatrix ab, sodass einerseits eine bessere Beschreibung der relevanten Bilder erzielt werden kann und andererseits als Folge davon gewöhnlich auch eine Invertierung der Kovarianzmatrix möglich ist. Aufbauend auf dem heuristischen Verfahren zur Adaption der Repräsentantengewichte sowie den positiven und negativen Bildbewertungen wurde daher in einem weiteren Experiment untersucht, inwieweit diese Ansätze wirklich den inhaltsbasierten Suchprozess unterstützen können.

Die experimentellen Untersuchungen wurden wie folgt durchgeführt: Die Regularisierung der singulären Kovarianzmatrix erfolgt iterativ. Dabei werden die Einträge

der Matrix nach der in Abschnitt 4.3.1 formulierten Berechnungsvorschrift solange variiert, bis sie invertierbar ist oder ein Maximalwert von $\gamma_{\text{max}} = 1.0$ erreicht bzw. überschritten wird. Falls der letztgenannte Fall eintritt wird eine diagonale Kovarianzmatrix berechnet. Die Bestimmung des Klassenradius innerhalb des Co-Training Verfahrens erfordert die Spezifikation des Parameters ρ (vgl. Abschnitt 4.3.3). Für diesen wird in den experimentellen Untersuchungen ein Wert von $\rho = 0.95$ gewählt. Für das Co-Training Verfahren wird zusätzlich zwischen zwei Varianten unterschieden. In der einen Variante („HarteCoTrainingVariante“) wird in dem unüberwachtem Lernschritt eine harte Klassifikation verwendet, d.h. ein bisher unklassifiziertes Element $\mathbf{r}$ wird, wenn es innerhalb des Klassenradius liegt, mit einem Gewicht $\pi(\mathbf{r}) = 1.0$ zur benachbarten Stützstelle klassifiziert. Im Gegensatz dazu wird in der zweiten Variante („WeicheCoTrainingVariante“) eine weiche Klassifikation in Abhängigkeit vom Klassenradius benutzt (vgl. Abschnitt 4.3.3). Die Verfahren wurden jeweils mit dem klassischen Ansatz verglichen, bei dem zunächst versucht wird, die komplette Kovarianzmatrix zu invertieren. Falls dies nicht gelingt, wird die diagonale Kovarianzmatrix zur Berechnung der Gewichtsmatrix verwendet.

Die Ergebnisse in Abbildung 4.18 und 4.19 demonstrieren, dass gerade in den ersten beiden Bewertungsschritten mit der Regularisierung und der weichen Co-Training Variante eine Verbesserung der Suchergebnisse erzielt werden kann. Wenn man die Resultate für die distanzbasierten Beispielsuchen der Kategorie „Autorennen“ in Abbildung 4.18(a) betrachtet, so kann festgestellt werden, dass nach dem ersten Bewertungsschritt mit der Regularisierung im Vergleich zum klassischen Ansatz („KlassischerHeuristischerAnsatz“) eine relative Verbesserung von 21% erzielt werden kann. Dies bedeutet, dass mit der Regularisierung bereits nach der ersten Bewertungsiteration im Durchschnitt 3.9 zusätzliche Bilder der gesuchten Kategorie gefunden werden. Demgegenüber wird mit der Co-Training Variante eine relative Verbesserung von 10% erreicht. Nach dem zweiten Bewertungsschritt wird mit der Regularisierung bzw. dem weichen Co-Training im Vergleich zum klassischen Ansatz sogar einen relative Verbesserung von 22% bzw. 14% erzielt. Die Kurvenverläufe in den Diagrammen der übrigen distanz- und rangbasierten Beispielsuchen bestätigen die betrachteten Ergebnisse, wobei die Unterschiede mal stärker und mal weniger stark ausgeprägt sind. Mit der harten Co-Training Variante können in den ersten Suchschritten im Vergleich zum klassischen Ansatz keine Verbesserungen erzielt werden. Es ist sogar in den ersten beiden Bewertungsschritten hauptsächlich eine Verschlechterung des Suchprozesses zu beobachten (vgl. z.B. Abbildung 4.18(d)). Die Ursache dafür ist, dass ein zuvor unbewertetes Bild mit vollem Gewicht ($\pi(r) = 1.0$) in den Adaptionsschritt einfließt. Dementsprechend wird die Stichprobe durch eine Fehlklassifikation zu stark verrauscht, sodass das System nicht in gewünschter Weise lernt und damit weniger relevante Bilder findet. Entsprechend der Kurvenverläufe der harten Co-Training Variante scheint das Problem einer Fehlklassifikation besonders in den ersten Suchschritten aufzutreten.



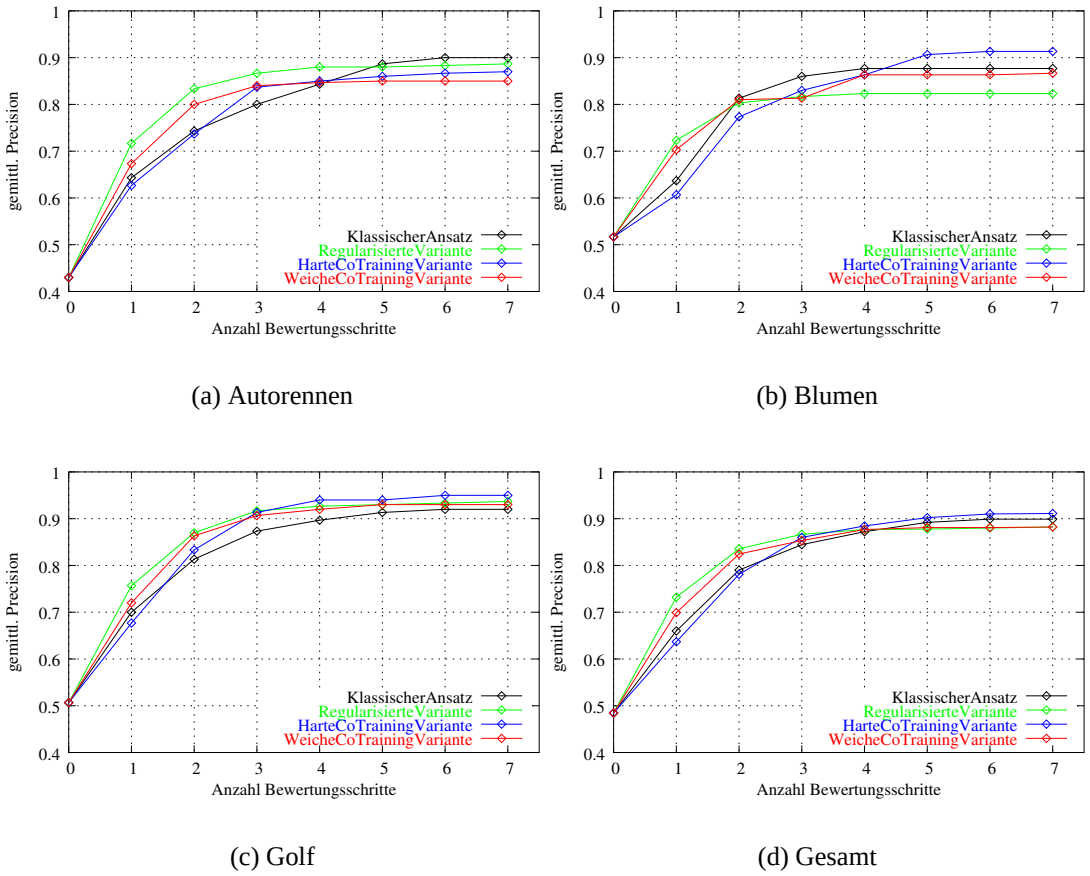
(a) Autorennen (b) Blumen

(c) Golf (d) Gesamt

Abb. 4.18: Co-Training und Regularisierung bei einer Ergebnismenge von 30 Bildern (distanzbasiert).

Mit zunehmender Anzahl der Bewertungsiterationen ist in den Graphen der gesamten Beispielsuchen (vgl. Abbildung 4.18(d) und Abbildung 4.19(d)) festzustellen, dass sich die verschiedenen Kurven angleichen und gegen einen maximalen durchschnittlichen Precision-Wert konvergieren. Die Verbesserung der klassischen Variante ist auf die sukzessive Vergrößerung der Trainingsmenge zurückzuführen. Insgesamt gesehen ist die Wirksamkeit der Regularisierung und des Co-Trainings besonders in den ersten Suchschritten auffällig. Dies ist besonders vorteilhaft, da es für ein Bildsuchsystem wichtig ist, möglichst schnell gegen die optimale Lösung zu konvergieren, sodass ein Anwender wenig Suchschritte benötigt, um die gemäß der Suchintention relevanten Bilder der Datenbank zu finden.

In den Ergebnisgraphen der experimentelle Untersuchungen ist außerdem ein stark domänenabhängiges Verhalten zu beobachten. Während sich bei den Beispielsuchen der Kategorien „Autorennen“ und „Golf“ überwiegend eine eindeutige Verbesserung der Suchergebnisse mit Co-Training und Regularisierung zeigt, werden die Ergebnisse



(a) Autorennen (b) Blumen

(c) Golf (d) Gesamt

Abb. 4.19: Co-Training und Regularisierung bei einer Ergebnismenge von 30 Bildern (rangbasiert).

der Kategorie „Blumen“ eher schlechter.[23] Lediglich mit der harten Co-Training Variante können ab dem dritten Bewertungsschritt vergleichbare Ergebnisse wie mit dem klassischen Ansatz erzielt werden (vgl. Abbildung 4.18(b)).

Die in Anhang D dargestellten Ergebnisse für eine Menge von 45 Ergebnisbildern bestätigen die bisherigen Ausführungen. Allerdings ist zu beobachten, dass aufgrund der größeren Trainingsmenge die Abstände zwischen den unterschiedlichen Kurven geringer sind. Dies ist insbesondere für die rangbasierten Beispielsuchen zu erkennen.

[23] Bei Betrachtung aller experimentellen Untersuchungen scheint es so, als ob die Bilder der Kategorie „Blumen“ deutlich unterschiedlicher sind als die der übrigen betrachteten Beispielkategorien. Dementsprechend schwieriger ist es für das System zu lernen, welche gemeinsamen Charakteristika die gesuchten Bilder besitzen. Die zum Teil recht auffälligen Kurvenverläufe könnten ein Indiz dafür sein.

Vergleich von distanz- und rangbasierter Bildersuche

Mit der Ausnahme des ersten Experiments wurden alle experimentellen Untersuchungen sowohl für den distanzbasierten als auch für den rangbasierten Suchansatz durchgeführt. Allerdings ist kein eindeutiger qualitativer Unterschied zwischen beiden Ansätzen zu beobachten. In einigen Diagrammen verlaufen die Kurven der rangbasierten Beispielsuchen glatter als die entsprechenden Kurven des distanzbasierten Ansatzes. Dies ist beispielsweise in Abbildung 4.14(b) und 4.15(b) zu beobachten. In anderen Abbildungen wiederum sind die Unterschiede der Suchergebnisse für den distanzbasierten Ansatz ein wenig deutlicher als für den rangbasierten Suchprozess. Beispiele dafür sind die Ergebnisse der ersten beiden Bewertungsschritte in Abbildung 4.18(d) und 4.19(d), wobei sich die Kurvenverläufe mit anwachsender Iterationenzahl immer mehr ähneln. In einigen Experimenten ist auch zu beobachten, dass sich mit dem rangbasierten Suchprozess bessere Ergebnisse erzielen lassen als mit der distanzbasierten Bildersuche (siehe Ausführungen zu Experiment B). Betrachtet man jedoch für jedes Experiment die gemittelten Suchergebnisse über alle Beispielsuchen (Diagramme mit der Bezeichnung „Gesamt“) sowohl für eine Ergebnismenge von 30 Bildern (Abbildungen 4.14 bis 4.19) als auch für eine Ergebnismenge von 45 Bildern (Abbildungen D.2 bis D.7 in Anhang D), so können, wenn überhaupt, nur marginale Unterschiede festgestellt werden. Dementsprechend kann keiner der beiden Ansätze eindeutig als besser oder schlechter bezeichnet werden.

4.4.5 Zusammenfassung der Ergebnisse

Die Ergebnisse der beschriebenen experimentellen Untersuchungen belegen die Leistungsfähigkeit des entwickelten inhaltsbasierten Suchverfahrens. In allen Diagrammen ist ein Lernverhalten des Systems zu erkennen. Dieses wird durch den Anstieg der dargestellten gemittelten Precision-Wert Kurven repräsentiert, die mit der Zunahme der Bewertungsschritte ein Konvergenzverhalten gegen einen maximalen gemittelten Precision-Wert zeigen. Außerdem konnte gezeigt werden, dass eine Modellierung mit separaten Bildrepräsentanten flexibler ist und mit dieser Darstellung bessere Suchergebnisse erzielt werden können als durch einen kombinierten Bildrepräsentanten. Weiterhin haben die experimentellen Untersuchungen gezeigt, dass das INDI System dann die besten Suchergebnisse erzielt, wenn sowohl ein generalisiertes Abstandsmaß als auch negative Beispiele zum Systemlernen verwendet werden. In den Untersuchungen der Adaptionsverfahren zum Lernen der Repräsentantengewichte konnte mit dem heuristischen Verfahren, das auf positiven und negativen Beispielen basiert, die besten Ergebnisse erzielt werden. Darüber hinaus haben die Experimente gezeigt, dass die Suchleistung des Systems durch Regularisierung und Co-Training verbessert werden kann.

In den Tabellen 4.1 bis 4.3 sind abschließend die über alle Beispielsuchen („Gesamt“) gemittelten Ergebnisse der Experimente B bis D zusammengefasst dargestellt. Die ver-

schiedenen Spalten repräsentieren jeweils die Position, an der das Verfahren entsprechend seines durchschnittlichen Precision-Wertes platziert ist. Dabei repräsentiert der erste Platz (1) das beste Suchergebnis. Die dargestellten Werte sind relative Verbesserungen (RV) in Prozent, die sowohl für die distanz- (dist) als auch rangbasierten (rang) Bildersuchen angegeben sind. Als Referenzergebnis für die relativen Verbesserungen wird für alle Experimente das jeweils schlechteste Verfahren gewählt.

Experiment B				
Platz	1	2	3	4
Ansatz	GeneralEuklidMitNeg	GeneralEuklid	GewEuklidMitNeg	GewEuklid
RV (dist)	39%	28%	17%	-
RV (rang)	38%	30%	15%	-

Tabelle 4.1: Experiment B: Ergebnisse aller experimentellen Untersuchungen (Gesamt) nach sieben Bewertungsschritten in relative prozentuale Verbesserungen (RV).

Experiment C				
Platz	1	2	3	4
Ansatz	HeuristischMitNeg	ModOpt	Opt/KeineAdaption	KombiRep
RV (dist)	33%	24%	19%	-
Ansatz	HeuristischMitNeg	ModOpt/Opt	KeineAdaption	KombiRep
RV (rang)	34%	27%	22%	-

Tabelle 4.2: Experiment C: Ergebnisse aller experimentellen Untersuchungen (Gesamt) nach sieben Bewertungsschritten in relative prozentuale Verbesserungen (RV).

Experiment D				
Platz	1	2	3	4
Ansatz	Regular.	WeicheCoTrain.	Klassisch.	HarteCoTrain
RV (dist-1. Bew.Schritt)	24%	17%	10%	-
RV (dist-2. Bew.Schritt)	12%	8%	7%	-
RV (rang-1. Bew.Schritt)	14%	9%	3%	-
RV (rang-2. Bew.Schritt)	8%	5%	1%	-

Tabelle 4.3: Experiment D: Ergebnisse aller experimentellen Untersuchungen (Gesamt) nach dem ersten und zweiten Bewertungsschritt (Bew.Schritt) in relative prozentuale Verbesserungen (RV).

4.5 Zusammenfassung

In diesem Kapitel wurde der für das INDI System entwickelte inhaltsbasierte Suchprozess vorgestellt. Dabei wurde ausgehend von der separaten Repräsentation der verschiedenen inhärenten Charakteristika eines Bildes zunächst ein formales Suchmodell formuliert. Aufbauend auf dem Suchmodell wurden mit der distanz- und rangbasierten Bildersuche zwei Ansätze zur merkmalsgetriebenen Datenbanksuche beschrieben. Das Hauptkennzeichen der ersten Variante ist das sukzessive Zusammenfassen von Einzeldistanzen zu einem Gesamtdistanzwert. Im Gegensatz dazu zeichnet den zweiten Ansatz aus, dass die Distanzen eines Bildobjekts in den verschiedenen Merkmalsräumen auf Ränge abgebildet werden, die zu einem Gesamtrang kombiniert werden. Beide Werte sind ein Maß für die Relevanz des entsprechenden Bildobjekts in Bezug auf die aktuellen Anfrage.

Der anschließend beschriebene Lernprozess basiert auf einem Verfahren, in dem die Abstände der Trainingsbeispiele zum idealen Anfrageobjekt minimiert werden. Im Adaptionsschritt werden auf der Grundlage einer klassifizierten Stichprobe die idealen Anfragevektoren, Gewichtsmatrizen sowie Gewichte der verwendeten Bildrepräsentanten gelernt. Dabei erfordert die Berechnung der Gewichtsmatrix die Invertierung der gewichteten Kovarianzmatrix der entsprechenden Trainingsvektoren. Um bei auftretenden Singularitäten den Suchprozess fortsetzen zu können, wurden mit der Beschränkung auf diagonale Kovarianzen und der Regularisierung zwei Varianten erläutert, die zur Lösung dieser Problematik dienen.

Außerdem wurde ein Ansatz vorgestellt, der neben den positiven Bildern auch negative Elemente zum Systemlernen der Repräsentantengewichte verwendet. Das Ziel dieses Verfahrens ist es zu lernen, welche Bildcharakteristika für die aktuelle Suchintention eines Benutzers am besten geeignet sind. Des Weiteren wurde ein Verfahren entwickelt, dessen Ziel die Erweiterung der normalerweise geringen Stichprobe ist, sodass eine robuste Schätzung der Kovarianzmatrix erzielt werden kann. Den Ansatz zeichnet aus, dass er durch die Kombination von überwachtem und unüberwachtem Lernen unklassifizierte Bilder der Datenbank in den Lernschritt integriert. In einer abschließenden Evaluation wurde der inhaltsbasierte Suchprozess des INDI Systems schließlich ausführlich evaluiert und seine Leistungsfähigkeit demonstriert.

5 Multidimensionale Indizierung

Viele der in der Literatur beschriebenen inhaltsbasierten Bildsuchsysteme operieren auf einer relativ kleinen Datenmenge ($< \ 10000$). Obwohl ihre Leistungsfähigkeit hinsichtlich der Qualität des Suchergebnisses demonstriert wird, bleiben sie häufig den Beweis schuldig, dass sie auf große Datenbestände skalierbar sind. Um jedoch den Charakter eines Forschungsprototypen abzulegen und den Status eines Bilddatenbanksystems zu erlangen, das auch in einem industriellen Umfeld eingesetzt werden kann, ist es notwendig, dass die inhaltsbasierten Techniken eines Bildsuchsystems auf umfangreiche Datenbestände skalierbar sind. Die Datenmengen industrieller Systeme übersteigen die von experimentellen Prototypen normalerweise erheblich, sodass das komplette Durchsuchen des gespeicherten Datenbestandes die Systemleistung stark verringern würde. Die für die komfortable Interaktion von Mensch und System benötigten kurzen Antwortzeiten können beispielsweise durch den Einsatz geeigneter Indizierungsmechanismen, die die Einschränkung des Suchraumes ermöglichen, erreicht werden. Da diese Verfahren die Organisation hochdimensionaler Datenelemente ermöglichen, werden sie als multidimensionale Indizierungsverfahren bezeichnet.

Die aus der Indizierung der gespeicherten Bildmenge resultierende Datenbankorganisation hat zudem einen entscheidenden Vorteil. Auf ihrer Grundlage kann das *Page Zero* Problem zwar nicht sicher gelöst, aber zumindest abgeschwächt werden.[1] Dabei wird unter Page Zero die erste Datenbankübersicht verstanden, die einem Benutzer zur Auswahl des initialen Beispielbildes präsentiert wird [La 98, Le 02]. Da der Erfolg eines Suchprozesses durchaus von der Güte des initial selektierten Beispielbildes abhängt, hat die Qualität der initialen Bildübersicht Auswirkungen auf das Suchergebnis. Gewöhnlich ist diese initiale Auswahl zufällig, sodass die Bildübersicht nicht unbedingt eine angemessene Repräsentation der gespeicherten Bildmenge darstellt. Existiert dagegen schon eine Gruppierung der gespeicherten Bilder, wie dies durch die multidimensionale Organisation der Daten gegeben ist, so kann eine repräsentative initiale Übersicht durch Auswahl typischer Elemente jeder Gruppe erzeugt werden.

Die genannten Aspekte veranschaulichen, dass die multidimensionale Indizierung neben extrahierten Bildcharakteristika, adäquaten Abstandsmaßen und adaptiven Suchverfahren einen wichtigen Bestandteil eines inhaltsbasierten Bildsuchsystems darstellt. Die Erläuterung der multidimensionalen Datenbankorganisation bildet daher

[1] Um das Page Zero Problem zu lösen, müsste für einen Benutzer auf der initialen Bildübersicht zumindest ein relevantes Bild vorhanden sein. Da die Suchintention eines Anwenders allerdings a priori nicht bekannt ist, kann die Erfüllung dieser Anforderung bei der Menge der gespeicherten Bilder und der entsprechenden Anzahl an verschiedenen Bildkategorien nicht garantiert werden.

den Schwerpunkt dieses Kapitels. Es werden sowohl etablierte Verfahren vorgestellt als auch der für das INDI System entwickelte experimentelle Indizierungsmechanismus beschrieben. Zusätzlich wird der Fragestellung nachgegangen, inwieweit sich die Qualität der Suchergebnisse durch die Einschränkung des Suchraumes verändert.

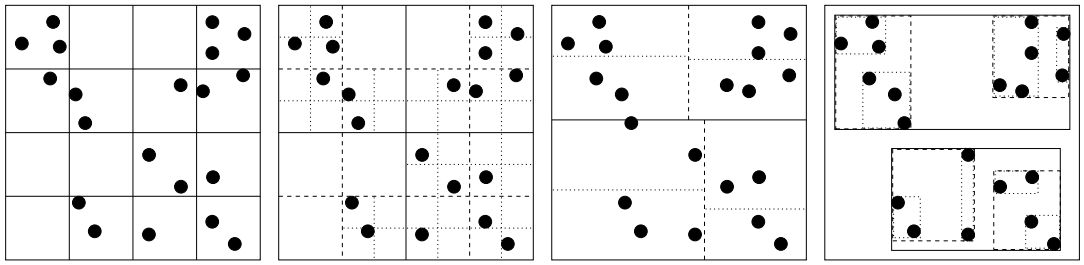
5.1 Indizierung hochdimensionaler Daten

Klassische Datenbanksysteme, wie sie beispielsweise von Lang & Lockemann [Lan95] oder von Matthiessen & Unterstein [Mat97] beschrieben werden, verwalten überwiegend Daten, die entweder Zeichenketten, Integerwerte oder reelle Zahlen sind. Zur Beschleunigung der Datenbankoperationen dienen Indexstrukturen, die hauptsächlich auf Binärbäumen oder *Hashes* basieren (vgl. z.B. [Elm02, Kapitel 6]). In speziellen Anwendungen, wie z.B. CAD[2]-Systemen oder geografischen Informationssystemen, ist auch die Speicherung von komplexen Datentypen wie z.B. Punkten, Linien, Rechtecken oder anderen geometrischen Objekten möglich. Der schnelle Zugriff auf solche hochdimensionale Daten erfordert allerdings eine andere Datenorganisation als der Zugriff auf klassische, „eindimensionale“ Datenelemente. Aus dieser Anforderung heraus wurden verschiedene multidimensionale Indizierungsmechanismen entwickelt, die den schnellen Zugriff auf hochdimensionale Daten ermöglichen. Den größten Beitrag dazu lieferten Arbeiten aus den Bereichen der Computergeometrie, Datenbanktechnologie und Mustererkennung [Rui99].

Die verschiedenen Indizierungsverfahren können in zwei Kategorien eingeteilt werden [Cas01]. Auf der einen Seite die sogenannten *Spatial Access Methods* (SAMs), die die Indizierung räumlicher Objekte wie z.B. Linien, Rechtecke oder Polygone ermöglichen. Auf der anderen Seite die *Point Access Methods* (PAMs), die Punkte in hochdimensionalen Vektorräumen strukturieren. Eine Trennung der beiden Klassen ist jedoch oftmals schwierig, da verschiedene Verfahren existieren, die sowohl als SAM als auch durch geringe Modifikationen als PAM eingesetzt werden können. In vielen Anwendungen werden räumliche Objekte häufig auch in einen hochdimensionalen Merkmalsraum transformiert, in dem sie als Punkte repräsentiert werden und schließlich durch eine PAM organisiert werden können. Da die in dieser Arbeit betrachtete inhaltsbasierte Bildersuche die Indizierung hochdimensionaler Merkmalsvektoren erfordert, werden in den folgenden Absätzen ausschließlich verschiedene punktbasierte Verfahren vorgestellt. Interessierte Leser finden beispielsweise in der Arbeit von Samet [Sam95] detaillierte Beschreibungen zur Indizierung von räumlichen Datentypen wie z.B. Linien oder Rechtecken.

Die naheliegendste Variante einen Eingaberaum zu indizieren, stellt die gleichmäßige Partitionierung dar. Bei diesem als *Bucketing* [Whi96a] oder *Fixed-Grid-*

[2] CAD: Abkürzung für *Computer Aided Design*



(a) Bucketing (b) Quadtree (c) k-d Tree (d) R-Tree

Abb. 5.1: Multidimensionale Indizierungsverfahren am Beispiel einer zweidimensionalen Datenmenge (basierend auf [Cas01]). Eine einfache, flache Indizierung wird durch das in (a) präsentierte Bucketing Verfahren erzielt. Im Gegensatz dazu partitionieren die in (b), (c) und (d) dargestellten Ansätze den Datenraum rekursiv, was zu einer hierarchischen Datenorganisation führt. Für Quadtrees, *k*-d Trees und R-Trees sind jeweils drei Zerlegungsstufen dargestellt. Die verschiedenen Linientypen korrespondieren mit den verschiedenen Ebenen des Baumes. Ausgehend vom Wurzelknoten sind dies die folgenden Linientypen: durchgezogen, gestrichelt und gepunktet.

Method [Cas01] bezeichnetem Verfahren wird der ursprüngliche Datenraum in Hyperkuben fester Größe eingeteilt. Darauf aufbauend werden die Datenelemente, die in demselben Kubus liegen, in einem Behälter[3] (engl. *Bucket*) gruppiert. Abbildung 5.1(a) demonstriert dieses Indizierungsverfahren am Beispiel einer zweidimensionalen Datenmenge. In der Praxis ist die zu indizierende Datenmenge oftmals jedoch nicht gleichmäßig in dem hochdimensionalen Datenraum verteilt, was bei einer Einteilung in fixe Hyperkuben dazu führt, dass viele Behälter nicht besetzt sind. Erweiterungen wie *Grid Files* [Cas01] versuchen diese Problematik zu lösen, indem sie die Bedingung einer festen Teilraumgröße lockern und zusätzlich erlauben, dass sich verschiedene Hyperkuben einen Behälter teilen.

Im Gegensatz zu diesen flach strukturierten Indizierungsverfahren existieren auch hierarchische Methoden, die den Eingaberaum rekursiv partitionieren. Die wohl bekanntesten Verfahren sind *Quadtrees* [Fin74], *k-d Trees* [Whi96a] und *R-Trees* [Gut84]. Ein *Quadtree* unterteilt den *k*-dimensionalen Datenraum in 2^k Regionen, indem jede Dimension in zwei gleichgroße Bereiche unterteilt wird. Die resultierenden Hyperkuben einer Baumebene werden somit durch $(k-1)$-dimensionale Hyperflächen getrennt, die durch die Teilungspunkte verlaufen und orthogonal zu den Teilungspunktachsen orientiert sind. Nicht-Terminalknoten des Baumes besitzen wiederum 2^k Kindknoten. Ein Beispiel für die Quadtree-Partitionierung eines zweidimensionalen Datenraumes ist in Abbildung 5.1(b) dargestellt. Die drei Zerlegungsstufen des Baumes sind durch die verschiedenen Linientypen gekennzeichnet. Gleiche Linien gehören zur selben Baumebene.

[3]Ein Behälter ist ein Speicherblock, der einen oder mehrere Datensätze beinhaltet.

Ein ähnliches und ebenfalls hierarchisches Verfahren ist der *k*-dimensionale Baum, der als *k*-d Tree bezeichnet wird. Bei dieser multidimensionalen Erweiterung des klassischen Binärbaumes wird der Merkmalsraum durch $(k-1)$-dimensionale Hyperflächen in einzelne Bereiche unterteilt. Dabei wird im Gegensatz zum Quadtree in jedem Schritt nur in einer Dimension gesplittet. In welcher Dimension und an welcher Stelle ein Vektorraum unterteilt wird, hängt vom gewählten Splittingkriterium ab. Ein mögliches Kriterium ist beispielsweise die Varianz der Datenmenge, sodass in der Dimension gesplittet wird, in der die Datenmenge am stärksten streut [Whi96a]. Als Splittingpunkt wird meistens der korrespondierende Mittelwert ausgewählt. Andere Ansätze wiederum unterteilen die Datenmenge in jedem Partitionierungsschritt in zwei gleichgroße Teilmengen [Cas01]. Abbildung 5.1(c) veranschaulicht einen varianzbasierten *k*-d Tree, in dem als Splittingpunkt der Mittelwert gewählt wird. Auch hier repräsentieren verschiedene Linien die unterschiedlichen Ebenen des multidimensionalen Binärbaumes.

In einem R-Tree werden Datenpunkte eines Merkmalsraumes durch minimal umgebene Hyperrechtecke (engl. *Minimal Bounding Rectangle*, MBR) repräsentiert [Gut84]. Analog zu den bereits erläuterten Verfahren erfolgt dies rekursiv, sodass die resultierende Baumstruktur eine Kaskade von MBRs darstellt. Dabei können die verschiedenen Hyperrechtecke einer Baumebene durchaus überlappen. Ein Beispiel für den R-Tree einer zweidimensionalen Datenmenge ist in Abbildung 5.1(d) dargestellt. Umgebene Rechtecke einer Ebene sind jeweils durch identische Linien gekennzeichnet. Aufbauend auf dieser von Guttman entwickelten Indexstruktur sind weitere Verfahren wie der *R⁺-Tree* [Sel87], *R*-Tree* [Bec90] oder *SS-Tree* [Whi96b] entwickelt worden. Sie unterscheiden sich von der ursprünglichen Methode durch effizientere Algorithmen zur Konstruktion des Baumes, bessere Suchleistung, optimierte Berechnungen der minimalen Hyperrechtecke oder der Verwendung von Hyperkugeln statt hochdimensionaler Rechtecke.

Da die vorgestellten Verfahren jedoch nicht mit der Dimension der zu speichernden Merkmalsvektoren skalieren [Ng96, Kri99, Che00, Cas01], ist ihr Einsatz auf niedrigdimensionalere Vektorräume (< 20) beschränkt. Gegebenfalls müssen die zu speichernden Datenvektoren vor dem Indizierungsprozess durch ein geeignetes Verfahren, wie beispielsweise die in Anhang C beschriebene Hauptachsentransformation, dimensionsreduziert werden. Die Leistungsfähigkeit eines Bildsuchsystems lässt sich in der Regel jedoch gerade durch höherdimensionale Bildbeschreibungen steigern, da in ihnen gewöhnlich mehr Informationen kodiert sind und daher oftmals eine bessere Unterscheidung der verschiedenen Bilder erzielt werden kann. Deshalb werden Indizierungsverfahren benötigt, die auch die effiziente Verwaltung hochdimensionaler Merkmalsvektoren (> 20) erlauben. Clusterverfahren aus der Mustererkennung erfüllen diese Anforderung und bieten sich daher zur multidimensionalen Indizierung an (vgl. z.B. [Laa00], [Mei02] oder [Käs03b]).

5.2 Verfahren der Clusteranalyse

Das Ziel der Clusteranalyse besteht darin, durch Anwendung unüberwachter numerischer Verfahren Häufungsgebiete einer unbekannten Datenverteilung zu finden. Solche Häufungsgebiete werden als Cluster oder Gruppe bezeichnet und die Elemente eines Clusters zeichnet aus, dass sie ausgehend von einer gewählten Repräsentation ähnlich im Sinne eines Abstandsmaßes sind. Grundlegende Eigenschaft eines Clusters sind sowohl die Homogenität als auch die Separierbarkeit. Diese Attribute beschreiben wie kompakt eine Datengruppe ist und wie gut sie von den anderen Häufungsgebieten des Datenraumes separiert ist. Gehört jedes Element der Datenmenge ausschließlich zu einem Cluster, so wird von einer disjunkten, andernfalls von einer nicht-disjunkten Gruppierung gesprochen. Alternativ werden auch die Bezeichnungen harte und weiche Gruppierung verwendet.

Die in der Literatur verfügbaren Clusterverfahren sind zahlreich (vgl. z.B. [Dud73], [Jai88] oder [Fuk90]). Neben hierarchischen Verfahren wie dem divisiven oder agglomerativen Clustern existieren vor allem Clusteralgorithmen, deren Ziel die Optimierung eines Gütekriteriums ist. In diese Kategorie gehören sowohl das von Ben Hur et al. [Ben02] vorgestellte und bereits in Kapitel 3.1.2 skizzierte Support Vector Clustering als auch die verschiedenen Verfahren zur Vektorquantisierung [Mac67, Lin80, Llo82]. Da die Vektorquantisierung bereits in anderen Arbeiten erfolgreich zur multidimensionalen Indizierung verwendet wurde [Che97, AM98, Mei02], wird in den folgenden Abschnitten die theoretische Grundlage dieses Verfahrens näher erläutert. Dabei orientieren sich die formalen Ausführungen an den Arbeiten von Fink [Fin03] sowie Gersho & Gray [Ger92]. Zusätzlich werden mit den selbstorganisierenden Karten und dem *Neural-Gas*-Algorithmus zum Vektorquantisierer verwandte Gruppierungsmethoden vorgestellt, die sich ebenfalls zur Organisation hochdimensionaler Daten eignen.

5.2.1 Vektorquantisierung

Motiviert durch die Eigenschaft, dass ähnliche Merkmalsvektoren Häufungsgebiete in einem hochdimensionalen Vektorraum bilden, unterteilt ein Vektorquantisierer einen Datenraum $\mathbb{R}^N$ in L Gebiete. Jedes dieser Gebiete besitzt einen typischen Repräsentantenvektor[4], auf den die Vektoren des Gebietes abgebildet werden. Die daraus resultierende kompakte Darstellung der Daten eines Merkmalsraumes ist gerade für die Übertragung und Speicherung digitaler Muster eine wichtige Eigenschaft. Da ähnliche Informationen in einem Vektor zusammengefasst werden, können durch die komprimierte Darstellung sowohl begrenzte Übertragungskapazitäten als auch begrenzter Speicherplatz effizient genutzt werden.

[4] Alternativ wird auch die Bezeichnung Prototyp oder Codewort verwendet.

Die Menge aller Repräsentantenvektoren $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L\}$ wird als Codebuch bezeichnet und der Vektorquantisierer (*VQ*) bildet jedes Element des *N*-dimensionalen Eingabedatenraumes auf ein Element des Codebuchs ab:

$$VQ : \mathbb{R}^N \mapsto Y$$

Dabei ist zu beachten, dass ein Vektorquantisierer neben dem Codebuch auch durch die entsprechende Partitionierung des Datenraumes charakterisiert ist. Da die Elemente eines Gebietes auf genau einen Prototypen abgebildet werden,

$$C_l = \{\mathbf{x} \mid VQ(\mathbf{x}) = \mathbf{y}_l\}, \text{ mit } \mathbf{x}, \mathbf{y}_l \in \mathbb{R}^N,$$

definiert ein *VQ* eine disjunkte Zerlegung des Datenraumes in verschiedene Quantisierungszellen:

$$\bigcup_{l=1}^{L} C_l = \mathbb{R}^N \quad \text{und} \quad C_l \cap C_k = \varnothing, \quad \forall l, k \text{ mit } l \neq k$$

Obwohl das Verfahren der Vektorquantisierung eine kompakte Repräsentation hochdimensionaler Daten ermöglicht, entsteht durch die Abbildung eines Vektors $\mathbf{x} \in \mathbb{R}^N$ auf einen von ihm verschiedenen Repräsentantenvektor $\mathbf{y}_l = VQ(\mathbf{x})$ ein Quantisierungsfehler. Dieser ist von der jeweiligen Abbildungsvorschrift *VQ* abhängig und lässt sich durch ein geeignetes Abstandsmaß $d : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}_0^+$ auch qualitativ erfassen (vgl. [Fin03, S. 55]):

$$E(\mathbf{x}, VQ) = d(\mathbf{x}, VQ(\mathbf{x})) = d(\mathbf{x}, \mathbf{y_l})$$

Die Charakterisierung der Güte eines Vektorquantisierers erfordert allerdings eine globale Betrachtung des Quantisierungsfehlers. Aus diesem Grund wird die zu erwartende Verzerrung, die dem statistischen Mittel des zu erwartenden Fehlers entspricht, bestimmt:

$$\overline{E}(VQ) = \mathcal{E}(E(X, VQ)) = \mathcal{E}(d(X, VQ(X))) = \int_{\mathbb{R}^N} d(\mathbf{x}, VQ(\mathbf{x}))p(\mathbf{x})d\mathbf{x} \tag{5.1}$$

Dabei wird vorausgesetzt, dass eine Zufallsvariable *X* existiert, die die statistischen Eigenschaften der Elemente $\mathbf{x} \in \mathbb{R}^N$ beschreibt und außerdem der Dichtefunktion $p(\mathbf{x})$ genügt. Ein Vektorquantisierer ist daher dann für eine gegebene Anzahl *L* von Gebieten optimal, wenn der Verzerrungsfehler (engl. *Distortion Error*) minimal ist. Da ein Quantisierer durch das Codebuch und die korrespondierende Partitionierung des Datenraumes $\mathbb{R}^N$ definiert ist, gilt es nun, Kriterien zu formulieren, die beide Komponenten optimieren. Allerdings ist bisher keine analytische Lösung dieses Problems bekannt, sodass eine iterative Lösung gefunden werden muss, in der die eine Komponente unter Berücksichtigung der anderen optimiert wird.

Eine optimale Partitionierung ist dann gegeben, wenn alle Vektoren einer Quantisierungszelle zu dem entsprechenden Zellenprototypen den minimalen Abstand besitzen.

Dementsprechend ist der Prototyp, verglichen mit allen anderen Codebuchvektoren, der sogenannte nächste Nachbar:

$$VQ(\mathbf{x}) = \mathbf{y}_l, \text{ falls } d(\mathbf{x}, \mathbf{y}_l) \leq d(\mathbf{x}, \mathbf{y}_k)\ \forall k \neq l \tag{5.2}$$

Ausgehend von diesem Nächster-Nachbar-Kriterium lässt sich der Quantisierungsfehler aus Gleichung 5.1 durch

$$\overline{E}(VQ) = \int_{\mathbb{R}^N} d(\mathbf{x}, VQ(\mathbf{x}))p(\mathbf{x})d\mathbf{x} \geq \int_{\mathbb{R}^N} \{\min_{\mathbf{y}\in Y} d(\mathbf{x}, \mathbf{y})\}p(\mathbf{x})d\mathbf{x}$$

nach unten abschätzen. Für ein gegebenes Codebuch ist daher dann die optimale Partitionierung des Datenraumes erzielt, wenn jeder Vektor $\mathbf{x} \in \mathbb{R}^N$ dem am nächsten benachbarten Prototypen zugeordnet wird.

Des Weiteren gilt es, bei gegebener Einteilung des Vektorraumes das entsprechende Codebuch zu optimieren. Dieses wird durch die Bestimmung der optimalen Repräsentantenvektoren erreicht. Ein Prototyp $\mathbf{y}_l$ einer Zelle C_l ist genau dann optimal, wenn er Zentroid der Zelle ist (vgl. [Fin03, S.56f]):

$$\mathbf{y}_l = \text{cent}(C_l) = \underset{\mathbf{y}\in C_l}{\text{argmin}}\ \mathcal{E}\{d(X, \mathbf{y})|X \in C_l\} \tag{5.3}$$

Für die in diesem Kapitel betrachteten elliptisch-symmetrischen Abstandsmaße wie beispielsweise der euklidische Abstand entspricht der Zentroid dem Erwartungswert aller Datenvektoren einer Quantisierungszelle.

Ausgehend von einer festen Partitionierung des Datenraumes lässt sich der mittlere Quantisierungsfehler durch die Zentroid-Bedingung wie folgt minimieren[5]:

$$\begin{aligned}
\overline{E}(VQ) &= \sum_{l=1}^{L} P(X \in C_l) \int_{\mathbf{x}\in C_l} d(\mathbf{x}, \mathbf{y}_l)p(\mathbf{x}|\mathbf{x} \in C_l)d\mathbf{x} \\
&= \sum_{l=1}^{L} P(X \in C_l)\ \mathcal{E}\{d(X, \mathbf{y}_l)|X \in C_l\} \xrightarrow{5.3} \min
\end{aligned}$$

Da die Wahrscheinlichkeiten $P(X \in C_l)$ positiv und die Gebiete disjunkt sind, erfordert die Minimierung des Gesamtfehlers $\overline{E}(VQ)$ lediglich die Minimierung der lokalen Fehler. Diese sind nach Gleichung 5.3 genau dann minimal, wenn der Zentroid einer Zelle als Repräsentantenvektor gewählt wird.

Wie in den letzten Abschnitten demonstriert wurde, sind für einen optimalen Vektorquantisierer Codebuch und Partitionierung direkt voneinander abhängig. Daher reicht

[5]Dabei werden anstelle der Wahrscheinlichkeitsdichte $p(\mathbf{x})$ die a priori Wahrscheinlichkeiten $P(X \in C_l)$ der Gebiete sowie die bedingten Dichten $p(\mathbf{x}|\mathbf{x} \in C_l)$ der auf eine Zelle beschränkten Vektoren betrachtet (vgl. [Fin03, S. 57]).

es aus, einen Quantisierer lediglich durch sein Codebuch zu charakterisieren, da ausgehend von der Nächster-Nachbar Bedingung auch die entsprechende Partitionierung des Datenraumes definiert ist. Des Weiteren fällt bei genauerer Betrachtung auf, dass der Entwurf eines Quantisierers auf der Verteilungsdichte $p(\mathbf{x})$ aller Vektoren des Datenraumes $\mathbb{R}^N$ basiert. Diese ist in der Praxis normalerweise nicht bekannt, sodass stattdessen eine Stichprobe $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ verwendet wird, die die ursprüngliche Verteilung des Eingabedatenraumes annähert. Im Folgenden wird mit dem Lloyd-Algorithmus, das wohl prominenteste Verfahren zum Design eines Vektorquantisierers vorgestellt.

Lloyd-Algorithmus

Die Hauptidee des Lloyd-Algorithmus[6] [Llo82] besteht in der schrittweisen und abwechselnden Optimierung von Codebuch und Partitionierung. Obwohl mit dem Verfahren zwar kein globales Minimum des Quantisierungsfehlers erreicht wird, kann abhängig vom initialen Codebuch zumindest ein lokales Minimum bestimmt werden. Ausgehend von einer Stichprobe *S* des Datenraumes $\mathbb{R}^N$ optimiert der in Abbildung 5.2 dargestellte Algorithmus ein initiales Codebuch Y^0 der Größe *L* so, dass der globale Quantisierungsfehler möglichst klein wird.

Entscheidenden Einfluss auf die Qualität des resultierenden Quantisierers haben dabei die Auswahl der Codebuchgröße, die gleichbedeutend mit der Anzahl der Quantisierungszellen bzw. Cluster ist, sowie das initial gewählte Codebuch. Während für das erstgenannte keine generelle Lösung existiert[7], können schon mit einer zufälligen Auswahl der initialen Codebuchvektoren gute Ergebnisse erzielt werden. Alternativ können auch die ersten *L* Vektoren der Stichprobe als initiale Prototypen ausgewählt werden. Dabei ist jedoch zu beachten, dass diese nicht miteinander korrelieren. Ergänzend zur Größe des Codebuchs sowie den Codebuchvektoren muss initial auch eine Güteschwelle ϵ spezifiziert werden. Diese dient als Abbruchkriterium, sodass der Algorithmus terminiert sobald die Differenz zweier aufeinander folgender Quantisierungsfehler unterhalb der Schwelle liegt. In einer leicht abgeänderten Variante des Algorithmus kann zusätzlich die Anzahl der maximalen Iterationen festgelegt werden, sodass das Verfahren auch terminiert, wenn dieses Maximum überschritten wird.

[6] Dieser ursprünglich 1957 an der Bell Forschungseinrichtung als Manuskript veröffentlichte Algorithmus wird in der Literatur häufig fälschlicherweise als *k*-means Algorithmus bezeichnet. Im Gegensatz zu dem von MacQueen [Mac67] vorgestellten original *k*-means Verfahren, das eine Stichprobe nur einmal durchläuft, handelt es sich bei dem Lloyd-Algorithmus jedoch um ein iteratives Verfahren, in dem die Beispielvektoren der Stichprobe mehrfach verarbeitet werden.

[7] Linde, Buzo und Gray schlagen zur Lösung dieser Problematik den von ihnen entwickelten LBG-Algorithmus [Lin80] vor. Dieses Verfahren basiert auf dem sukzessiven Splitten und Optimieren von Klassengebieten, wobei mit einem initialen Prototypen gestartet wird. In anderen Arbeiten wiederum wird die Verwendung von Güteindizes vorschlagen, um die für die gegebene Stichprobe beste Clusteranzahl zu bestimmen (vgl. z.B. [Mil85] oder [Mau02]).

Gegeben sei eine Stichprobe von Datenvektoren $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, eine Anzahl von Gebieten L und eine untere Schranke ϵ.

Wähle zufällig oder aufgrund von Vorwissen die Prototypen $\mathbf{y}_l^0$ des Codebuchs Y^0

Initialisiere Quantisierungsfehler $E^0 = \infty$ und Iterationszähler $i = 0$

$i = i + 1$; $E^i = 0$ (a)

FOR alle Vektoren $\mathbf{x}_m$ der Stichprobe (b)

Bestimme die zugehörige Zelle C_l durch Identifikation des benachbarten Prototypen $\mathbf{y}_l^{i-1}$ gemäß Gleichung 5.2

$E^i = E^i + d(\mathbf{x}_m, \mathbf{y}_l^{i-1})$ (c)

FOR alle Zellen C_l (d)

Aktualisiere Codebuchprototypen durch Berechnung der Zentroiden:
$\mathbf{y}_l^i = \text{cent}(C_l) = \frac{1}{N_l} \sum_{\mathbf{x}_m \in C_l} \mathbf{x}_m, \quad N_l = \text{card}\{\mathbf{x}_m \mid \mathbf{x}_m \in C_l\}$

$E^i = E^i / M$

UNTIL $(E^{i-1} - E^i)/E^i < \epsilon$

Abb. 5.2: Lloyd-Algorithmus

Nach der Initialisierung wird die erste Iteration gestartet (a). Für jeden Iterationsschritt muss zu Beginn sowohl der Iterationszähler i aktualisiert als auch der aktuelle Quantisierungsfehler E^i initialisiert werden. Darauf aufbauend wird im nächsten Verarbeitungsschritt unter Berücksichtigung des aktuellen Codebuchs die optimale Partitionierung des Datenraumes bestimmt (b). Dabei wird für jedes Element der Stichprobe der am nächsten benachbarte Prototyp identifiziert und die zugehörige Quantisierungszelle bestimmt. Zusätzlich wird für jedes Element der Stichprobe der aktuelle Quantisierungsfehler neu berechnet (c). Nach der Klassifikation aller Stichprobenelemente zu den verschiedenen Gebieten wird im nächsten Schritt das Codebuch optimiert (d). Dazu wird für jede Zelle ausgehend von der Zentroid-Bedingung der optimale Codebuchvektor berechnet. Liegt nach dem Ende eines Iterationsschritts die relative Verbesserung des Fehlers unterhalb der spezifizierten Güteschwelle, dann terminiert der Algorithmus. Andernfalls wird das Verfahren mit der nächsten Iteration fortgesetzt (a).

5.2.2 Selbstorganisierende Karten

Bei der von Kohonen entwickelten selbstorganisierenden Karte (engl. *Self Organizing Map*, SOM)[8] handelt es sich um ein neuronales Clusterverfahren, das ebenfalls in die

[8] In der Literatur werden oftmals auch die Bezeichnungen Kohonenkarte oder *Kohonen-Feature-Map* verwendet.

Gegeben sei eine Stichprobe $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, eine Topologie von Neuronen l (mit $l = 1, 2, \ldots, L$), eine maximale Anzahl von Lernschritten $t_{\max}$ sowie die in Gleichung 5.5 formulierte Nachbarschaftsfunktion $h_{bl}(t)$.

Wähle initiale Referenzvektoren $\mathbf{y}_l$ (z.B. zufällig) sowie Anfangswerte für den Breitenparameter $\sigma(t)$ und die Lernrate $\alpha(t)$.

$t = 0$

WHILE $t < t_{\max}$

- Wähle zufällig Beispielvektor $\mathbf{x} \in S$
- Bestimme BMU für $\mathbf{x}$: $b = \underset{l}{\operatorname{argmin}}\{\|\mathbf{x} - \mathbf{y}_l(t)\|\}$
- Aktualisiere für jedes Neuron l den Referenzvektor $\mathbf{y}_l$:
  $\mathbf{y}_l(t+1) = \mathbf{y}_l(t) + \alpha(t) h_{bl}(t)[\mathbf{x} - \mathbf{y}_l(t)]$
- Verändere $\sigma(t) \rightarrow \sigma(t+1)$ und $\alpha(t) \rightarrow \alpha(t+1)$ nach Plan
- $t = t + 1$

Abb. 5.3: SOM-Algorithmus

Kategorie der Vektorquantisierer eingeteilt werden kann. Eine SOM besteht aus einer Menge von Neuronen, die in einer festen Topologie, wie z.B. rechteckige oder hexagonale Strukturen, organisiert sind. Mit jedem Neuron l ist ein Referenzvektor $\mathbf{y}_l$ assoziiert, der ein Element des betrachteten Eingaberaumes $\mathbb{R}^N$ darstellt. Im Sinne der Vektorquantisierung entsprechen die Referenzvektoren den Prototypen des Codebuchs. Da jedes Element $\mathbf{x}$ des Eingaberaumes einem Referenzvektor $\mathbf{y}_l$ und somit einem Neuron l zugeordnet werden kann, realisiert eine SOM eine Abbildung des Eingaberaumes auf eine Schicht von Neuronen. Dabei wird die topologische Struktur des Eingaberaumes beibehalten, sodass im Merkmalsraum benachbarte Eingabemuster auch auf topologisch benachbarte Neuronen abgebildet werden.

Eine SOM wird wie der Algorithmus in Abbildung 5.3 veranschaulicht iterativ trainiert. Dabei wird in jedem Iterationsschritt aus der Menge der Beispielvektoren zufällig ein Eingabevektor $\mathbf{x}$ ausgewählt.[9] Analog zum Lloyd-Algorithmus wird im ersten Verarbeitungsschritt der Referenzvektor bestimmt, der zum betrachteten Eingabevektor den geringsten Abstand besitzt. Das Neuron, dessen Referenzvektor dieses Kriterium erfüllt wird als *Best Matching Unit* (BMU) bezeichnet. Viele Anwendungen

[9] Um zu gewährleisten, dass in einem Durchlauf jeder Beispielvektor der Stichprobe verarbeitet wird, bietet es sich an, vor jedem Durchlauf die Stichprobe zufällig zu permutieren und dann sequentiell zu verarbeiten.

verwenden in diesem Schritt den euklidischen Abstand $d(\mathbf{x}, \mathbf{y}_l) = \|\mathbf{x} - \mathbf{y}_l\|$, sodass die durch den Index b repräsentierte BMU durch

$$b = \underset{l=1,2,\ldots,L}{\operatorname{argmin}}\{\|\mathbf{x} - \mathbf{y}_l\|\}$$

definiert ist. Im Gegensatz zum klassischen Vektorquantisierer ist der Einfluss eines Eingabemusters $\mathbf{x}$ allerdings nicht ausschließlich auf den Referenzvektor $\mathbf{y}_b$ der BMU beschränkt. Zusätzlich werden auch die Referenzvektoren der topologisch benachbarten Neuronen beeinflusst, sodass auch sie in Richtung des Eingabevektors gezogen werden. Für die Aktualisierung des Prototypen $\mathbf{y}_l$ eines Neurons l gilt daher:

$$\mathbf{y}_l(t+1) = \mathbf{y}_l(t) + \alpha(t)h_{bl}(t)[\mathbf{x} - \mathbf{y}_l(t)], \tag{5.4}$$

wobei $t \in [0, 1, \ldots, t_{\max} - 1]$ einen Zeit- bzw. Lernschritt des Verfahrens repräsentiert. Die Anzahl der Iterationsschritte wird durch die Intervalgrenze $t_{\max}$ definiert. Die Zeiteinheit eines Lernverfahrens wird häufig in Lernepochen formuliert. Dabei wird unter einer Epoche die einmalige Verarbeitung aller Beispielvektoren verstanden. Die Anzahl der Iterationsschritte einer Epoche entspricht demzufolge dem Umfang der Stichprobe S, $\text{card}(S) = M$. Entsprechend ergibt sich die maximale Anzahl der Lernschritte aus der Multiplikation der Epochenzahl N_E mit der Anzahl der Beispielvektoren M, $t_{\max} = N_E M$.

Wie Gleichung 5.4 demonstriert, wird ein Lernschritt durch den Lernratenparameter $\alpha \in [0, 1]$ begrenzt. Dieser wird zu Beginn des Verfahrens recht groß gewählt und nimmt mit der Zeit gewöhnlich monoton ab, z.B. linear $\alpha(t) = \alpha(0)(1 - t/t_{\max})$. Ähnliches gilt für die Nachbarschaftsfunktion $h_{bl} \in [0, 1]$, die in vielen Anwendungen in Abhängigkeit von der Gaußfunktion definiert ist:

$$h_{bl}(t) = e^{\frac{\|\mathbf{r}_b - \mathbf{r}_l\|^2}{2\sigma^2(t)}} \tag{5.5}$$

Dabei bezeichnen $\mathbf{r}_b$ und $\mathbf{r}_l$ die Koordinaten der jeweiligen Neuronen b und l auf dem SOM Gitter. Der Breitenparameter $\sigma(t)$ der Gaußglocke wird mit wachsender Zeit t so variiert, dass h_{bl} gegen Null konvergiert, $h_{bl} \to 0$ für $t \to \infty$ und $l \neq b$. Die Parameteradaption kann beispielsweise nach folgender Berechnungsvorschrift erfolgen:

$$\sigma(t) = \sigma_s(\sigma_e/\sigma_s)^{t/t_{\max}},$$

wobei σ_s und σ_e geeignete Start- und Endwerte repräsentieren. Der Lernschritt der SOM fokussiert mit ansteigendem t dementsprechend immer stärker auf die direkte Nachbarschaft der BMU.

5.2.3 Neural-Gas

Die Topologiebindung einer selbstorganisierenden Karte erfordert die initiale Wahl einer Neuronenstruktur. Diese sollte die Datenverteilung des Eingaberaumes möglichst

Gegeben sei eine Stichprobe $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$, eine Menge von Neuronen l, $l = 1, 2, \ldots, L$, und eine maximale Anzahl von Lernschritten $t_{\max}$.

Wähle initiale Referenzvektoren $\mathbf{y}_l$ (z.B. zufällig), sowie Funktionen für die Lernrate $\alpha(t)$ und den Parameter $\lambda(t)$ der Nachbarschaftsfunktion $h_{\lambda(t)}$

$t = 0$

WHILE $t < t_{\max}$

- Wähle zufällig Beispielvektor $\mathbf{x} \in S$
- Bestimme für jedes Neuron l die Anzahl k_l der Neuronen j, die näher am aktuellen Eingabevektor $\mathbf{x}$ liegen: $k_l = \text{card}\{\, j \mid \|\mathbf{x} - \mathbf{y}_j\| < \|\mathbf{x} - \mathbf{y}_l\|\}$
- Aktualisiere für jedes Neuron $l = 1, \ldots, L$ den Referenzvektor $\mathbf{y}_l$:
  $\mathbf{y}_l(t+1) = \mathbf{y}_l(t) + \alpha(t) h_{\lambda(t)}(k_l)[\mathbf{x} - \mathbf{y}_l], \quad \text{mit } h_{\lambda(t)}(k_l) = e^{-k_l/\lambda(t)}$
- $t = t + 1$

Abb. 5.4: Neural-Gas-Algorithmus

adäquat beschreiben. Passt die Struktur nicht zu der Topologie der zu repräsentierenden Datenmenge, entstehen gegebenenfalls topologische Defekte, die zu einer fehlerhaften Abbildung führen können. In dem von Martinetz et al. [Mar91, Mar93] vorgestellten Verfahren des *Neural-Gas* (NG) existiert diese Problematik nicht, da Neuronennachbarschaften nicht in einem Topologieraum definiert werden. Stattdessen werden Nachbarschaften von Neuronen direkt im entsprechenden Eingabedatenraum definiert. Da sich die Neuronen ähnlich wie ausbreitende Gaspartikel frei in diesem Merkmalsraum bewegen dürfen, können sie die Topologie der Datenverteilung selbständig bestimmen.[10]

Ähnlich wie bei einer SOM ist mit jedem Neuron l ein Referenzvektor $\mathbf{y}_l$ assoziiert. Doch statt wie bei einer Kohonenkarte für jeden beobachteten Eingabevektor $\mathbf{x}$ die BMU zu bestimmen und die Referenzvektoren der im Topologieraum benachbarten Neuronen zu verschieben, verfolgt das NG eine andere Strategie (vgl. Algorithmus in Abbildung 5.4). Für jedes Eingabemuster $\mathbf{x} \in \mathbb{R}^N$ werden die verschiedenen Referenzvektoren entsprechend ihres Abstandes zu $\mathbf{x}$ geordnet:

$$\{\mathbf{y}_l^0, \mathbf{y}_l^1, \ldots, \mathbf{y}_l^{L-1}\}, \text{ mit } \|\mathbf{x} - \mathbf{y}_l^0\| \leq \|\mathbf{x} - \mathbf{y}_l^1\| \leq \cdots \leq \|\mathbf{x} - \mathbf{y}_l^{L-1}\|$$

Ausgehend von dieser Sortierung kann jedem Neuron l ein Rang k_l in der geordneten Liste zugewiesen werden. Dabei entspricht k_l der Anzahl von Neuronen j, deren Referenzvektor $\mathbf{y}_j$ näher am aktuellen Vektor $\mathbf{x}$ liegen als $\mathbf{y}_l$:

$$k_l = \text{card}\{\, j \mid \|\mathbf{x} - \mathbf{y}_j\| < \|\mathbf{x} - \mathbf{y}_l\|\}$$

[10] Dabei wird parallel zum eigentlichen Quantisierungsprozess mittels Hebb'schem Wettbewerbslernen eine Topologie erzeugt (vgl. [Mar91]).

Die Ordnung der Referenzvektoren ist gleichbedeutend mit der Definition einer Nachbarschaft. Der Grad der Nachbarschaft nimmt vom ersten bis zum letzten Element der Liste ab. Darauf aufbauend ist der Adaptionsschritt eines Referenzvektors durch

$$\Delta \mathbf{y}_l = \alpha(t) h_{\lambda(t)}(k_l)[\mathbf{x} - \mathbf{y}_l]$$

definiert. Analog zur SOM wird jeder Adaptionsschritt durch eine zeitabhängige Lernrate $\alpha(t) \in [0, 1]$ begrenzt. Die Nachbarschaftsfunktion $h_{\lambda(t)}(k_l)$ ist für $k_l = 0$ Eins und nähert sich abhängig von einem Verzögerungsparameter $\lambda(t)$ mit ansteigendem k_l dem Wert Null, $h_{\lambda(t)}(k_l) \to 0$ für $k_l \to (L-1)$. Martinetz et al. [Mar93] verwenden in ihrer Arbeit eine Exponentialfunktion zur Beschreibung der Nachbarschaftsfunktion:

$$h_{\lambda(t)}(k_l) = e^{-k_l/\lambda(t)}$$

Damit das Verfahren konvergiert, nehmen sowohl die Lernrate $\alpha(t)$ als auch der Verzögerungsparameter $\lambda(t)$ mit der Anzahl der Lernschritte monoton ab. Oftmals wird eine von der maximalen Anzahl der Lernschritte $t_{\max}$ abhängige Parameteränderung verwendet:

$$\lambda(t) = \lambda_s(\lambda_e/\lambda_s)^{t/t_{\max}} \quad \text{und} \quad \alpha(t) = \alpha_s(\alpha_e/\alpha_s)^{t/t_{\max}}$$

Die Wertebelegungen für die jeweiligen Start- und Endparameter sind in der Regel von der Anwendung abhängig.

5.3 Güteindizes

Die in den vorherigen Abschnitten vorgestellten Clusterverfahren sind alle von der Zielsetzung motiviert, eine gegebene Menge von Beispielvektoren in disjunkte Cluster zu gruppieren. Ihre Anwendung setzt jedoch eine manuelle Parametrisierung voraus. Neben der Clusteranzahl müssen Parameter wie beispielsweise das zu verwendende Abstandsmaß, Lernraten oder Nachbarschaftsfunktionen spezifiziert werden. Dementsprechend führen unterschiedliche Einstellungen der Systemparameter zu unterschiedlichen Gruppierungen der Datenmenge. Um festzustellen, welche Gruppierung für eine Datenmenge die beste Gruppierung ist, wurden Güteindizes (engl. *Validity Indices*) entwickelt, die ein quantitatives Maß für die Qualität einer gegeben Datengruppierung darstellen (vgl. z.B. [Mil85], [Mau02] oder [Käs03b]). Dabei ist jedoch sicherzustellen, dass die strukturellen Annahmen der Qualitätsmaße die adäquate Beschreibung eines bestimmten Clusterergebnisses überhaupt zulassen. Ein Güteindex, dessen Berechnungsmodell von sphärischen Clustern ausgeht, liefert keine sinnvollen Ergebnisse, wenn er für ein Clusterverfahren eingesetzt wird, das beispielsweise wie das Support Vector Clustering[11] [Ben02] beliebig geformte Datengruppen erzeugt. Qualitätsmaße

[11] Vorausgesetzt wird dabei, dass das SVC nicht im Modus der Dichteschätzung verwendet wird.

sind daher relative Maße in Bezug auf ein bestimmtes Clusterverfahren. Sie dienen dazu, bei gegebener Datenmenge die optimale Parametrisierung eines Clusterverfahrens zu finden. Ein auf ihnen basierender Vergleich unterschiedlicher Clusterverfahren ist in der Regel nur dann sinnvoll, wenn sich die Eigenschaften der Verfahren ähneln. Mit dem durchschnittlichen Quantisierungsfehler wurde bereits ein Maß vorgestellt, das für eine gegebene Anzahl von Clustern einen quantitativen Vergleich der Ergebnisse zweier Quantisierungsverfahren ermöglicht.

Milligan und Cooper [Mil85] haben in ihrer Arbeit dreißig verschiedene Qualitätsmaße miteinander verglichen. Aus den experimentellen Untersuchungen geht hervor, dass der von Davies und Bouldin [Dav79] entwickelte Güteindex gute Resultate liefert. Da sich diese Ergebnisse in experimentellen Voruntersuchungen bestätigt haben, wird das Qualitätsmaß in dieser Arbeit sowohl zur Bestimmung der optimalen Clusteranzahl einer gegebenen Datenmenge als auch für den Vergleich zweier Clusterverfahren verwendet (vgl. Abschnitt 5.5.3).

Davies-Bouldin-Index

Die Grundlage des Davies-Bouldin-Index [Dav79] bildet die Forderung nach einer möglichst großen Separierbarkeit der verschiedenen Cluster $\{C_l | l = 1, 2, \ldots, L\}$. Ausgehend von dieser Forderung werden die Cluster einer Gruppierung paarweise verglichen. Wie gut zwei Cluster C_l und C_k voneinander separiert sind, lässt sich durch das Verhältnis der summierten Intraclusterstreuungen S_l und S_k zu dem jeweiligen Clusterabstand d_{lk} quantitativ erfassen:

$$R_{lk} = \frac{S_l + S_k}{d_{lk}(\mathbf{y}_l, \mathbf{y}_k)}$$

Dabei ist die Intraclusterstreuung S_l ein Maß für die Ausdehnung des Clusters C_l bzw. die Streuung seiner Elemente $\mathbf{x} \in C_l$ um das Clusterzentrum $\mathbf{y}_l$. Sie ist durch folgende Berechnungsvorschrift definiert:

$$S_l = \frac{1}{|C_l|} \sum_{\mathbf{x} \in C_l} \|\mathbf{x} - \mathbf{y}_l\|, \; |C_l| = \text{card}(C_l)$$

Der Abstand zweier Cluster wiederum ist durch den Abstand der entsprechenden Zentroiden $d_{lk}(\mathbf{y}_l, \mathbf{y}_k) = \|\mathbf{y}_l - \mathbf{y}_k\|$ gegeben. Je geringer die jeweiligen Ausdehnungen zweier Cluster sind und je größer der Abstand ihrer Zentroiden ist, desto besser sind sie voneinander separiert. Quantitativ wird dies durch einen kleinen Wert R_{lk} symbolisiert. Umgekehrt repräsentiert ein großer Wert R_{lk} eine schlechte Separierung zweier Cluster C_l und C_k. Die schlechteste oder auch geringste Separierung eines Cluster l zu allen anderen Clustern k, $k \neq l$, ist durch

$$R_l = \max_{k \neq l} R_{lk}$$

gegeben. Darauf aufbauend ist der Davies-Bouldin-Index I_{DB} einer Gruppierung $\{C_l | l = 1, 2, \ldots, L\}$ wie folgt definiert:

$$I_{DB}(L) = \frac{1}{L} \sum_{l=1}^{L} R_l$$

Das Qualitätsmaß entspricht somit der mittleren geringsten Separierbarkeit aller Cluster. Je geringer $I_{DB}(L)$ ist, desto besser sind die verschiedenen Cluster voneinander separiert. Im Sinne des Güteindizes signalisiert dies eine bessere Qualität der Gruppierung der Datenmenge. Für $L = 1$ ist das Qualitätsmaß nicht definiert und für die triviale Lösung, in der jeder Cluster aus einem Element besteht (L = Anzahl der Datenelemente), gilt für seinen Wert $I_{DB}(L) = 0$. Der Güteindex sollte in der Praxis daher nur berechnet werden, wenn die verschiedenen Cluster eine sinnvolle Anzahl von Datenvektoren beinhalten.

5.4 Multidimensionale Indizierung in INDI

Im vorherigen Kapitel wurden mit dem distanz- und rangbasierten Suchansatz sowie den verschiedenen Verfahren der Systemadaption die elementaren Bestandteile des adaptiven Bildsuchsystems INDI vorgestellt. Die Ausführungen konzentrierten sich bislang auf die adaptiven Mechanismen und wie durch Mensch-Maschine Interaktion die semantische Lücke zwischen Anwender und System verringert werden kann. Ob und wie sich die beschriebenen Algorithmen auf große Datenmengen skalieren lassen wurde bisher nicht erläutert. In den folgenden Abschnitten werden daher sowohl die Aspekte der Skalierbarkeit des INDI Systems näher untersucht als auch der in das System integrierte experimentelle Indizierungsmechanismus vorgestellt.

5.4.1 Aufwandsanalyse

Das in Kapitel 4 beschriebene Suchverfahren des INDI Systems basiert auf dem kompletten bzw. linearen Durchsuchen des gespeicherten Datenbestandes. Dabei werden alle gespeicherten Bildobjekte mit dem Anfrageobjekt verglichen. Wird beispielsweise der Suchschritt einer distanzbasierten Bildersuche auf der Grundlage eines Bildrepräsentanten (d.h jedes Bildobjekt der Datenbank wird genau durch einen Merkmalsvektor beschrieben) betrachtet, so kann die Rechenzeit für eine Suchiteration durch

$$T_{\text{komplett}} = T_{\text{dist}} + T_{\text{sort}} = K\, T_{\text{objdist}} + O(K \log_2 K) \tag{5.6}$$

abgeschätzt werden.[12] Der zeitliche Aufwand eines Suchschritts resultiert demnach aus der Rechenzeit zweier Verarbeitungsschritte. Der erste Summand T_{dist} bezeichnet die Zeit, die notwendig ist, um das Anfrageobjekt mit den Bildobjekten der Datenbank zu vergleichen. Bei einer Datenbank von K Objekten bedeutete dies, dass K Abstandswerte berechnet werden, wobei die Zeit für eine Abstandsberechnung T_{objdist} beträgt. Der zweite Summand T_{sort} repräsentiert den zeitlichen Aufwand, der notwendig ist, um die K Bildobjekte auf der Grundlage ihrer Abstandswerte zu sortieren. Die Ausführungen demonstrieren, dass der Rechenaufwand eines Suchschritts mit der Anzahl der gespeicherten Bildobjekte zunimmt. Demnach können mit dem kompletten Durchsuchen der Datenbank nur dann akzeptable Antwortzeiten erzielt werden, wenn die gespeicherte Bildmenge nicht so umfangreich ist.

Um auch für große Datenmengen kurze Antwortzeiten erzielen zu können, ist es notwendig, den Suchraum[13] einzuschränken bzw. zu beschneiden.[14] Anstatt mit allen Elementen der Datenbank, wird ein Anfrageobjekt dann lediglich mit einer Teilmenge des Datenbestandes verglichen. Die zentrale Aufgabe, die für die Suchraumeinschränkung gelöst werden muss, ist die Identifikation der Bildobjekte, die gemäß der aktuellen Anfrage die größte Relevanz besitzen und daher mit dem Anfrageobjekt verglichen werden sollten. Ausgangspunkt dieser Identifikation ist eine Gruppierung der Bildobjekte $\mathcal{O}_k$ auf der Grundlage ihrer Merkmalsvektoren $\mathbf{r}^k$. Diese Gruppierung kann z.B. mit einem der in Abschnitt 5.2 vorgestellten Vektorquantisierungsverfahren erzielt werden. Da jeder Merkmalsvektor dementsprechend einem Cluster zugeordnet ist und außerdem mit jedem Vektor ein Bildobjekt assoziiert ist, ist eine Gruppierung der Bildobjekte

$$G = \{C_1, C_2, \ldots, C_L\}, \text{ mit } C_l = \{\mathcal{O}_1^l, \mathcal{O}_2^l, \ldots, \mathcal{O}_{N_l}^l\}$$

gegeben. Dabei besteht jeder Cluster C_l aus einer Menge von $N_l = \text{card}(C_l)$ Bildobjekten $\mathcal{O}_k^l$. Zusätzlich ist jeder Cluster C_l durch den entsprechenden Prototyp $\mathbf{y}_l \in \mathbb{R}^N$ charakterisiert (vgl. Abschnitt 5.2.1). Basierend auf der Datenraumpartitionierung können die Bildobjektcluster bestimmt werden, deren Elemente dem aktuellen Anfrageobjekt am ähnlichsten sind und daher den Suchraum für den nächsten Suchschritt bilden. Dazu wird für jeden Cluster C_l der Abstand zum aktuellen Anfrageobjekt $\mathcal{Q}$ auf der Grundlage des jeweiligen Clusterprototypen $\mathbf{y}_l$ und des aktuellen Anfragevektors $\mathbf{q}$ berechnet, wobei $\mathbf{W}$ wie in Abschnitt 4.1 die Gewichtsmatrix des Abstandsmaßes repräsentiert:

$$D(C_l, \mathcal{Q}) = d(\mathbf{y}_l, \mathbf{q}, \mathbf{W}) \tag{5.7}$$

[12] Für die Abschätzung der Rechenzeit des Sortiervorgangs dient der Quicksort Algorithmus als Grundlage. Dieser besitzt bei einer Datenmenge von K Elementen einen durchschnittlichen Rechenaufwand von $O(K \log_2 K)$ [Knu73].

[13] Mit dem Begriff Suchraum ist hier eine Menge von Bildobjekten gemeint, mit denen das Anfrageobjekt in einem Suchschritt verglichen wird. Bei der kompletten Bildersuche besteht der Suchraum aus allen Bildobjekten der Datenbank.

[14] Dieser Vorgang wird in der Literatur auch als *Pruning* bezeichnet (vgl. z.B. [Fin03, S. 163]).

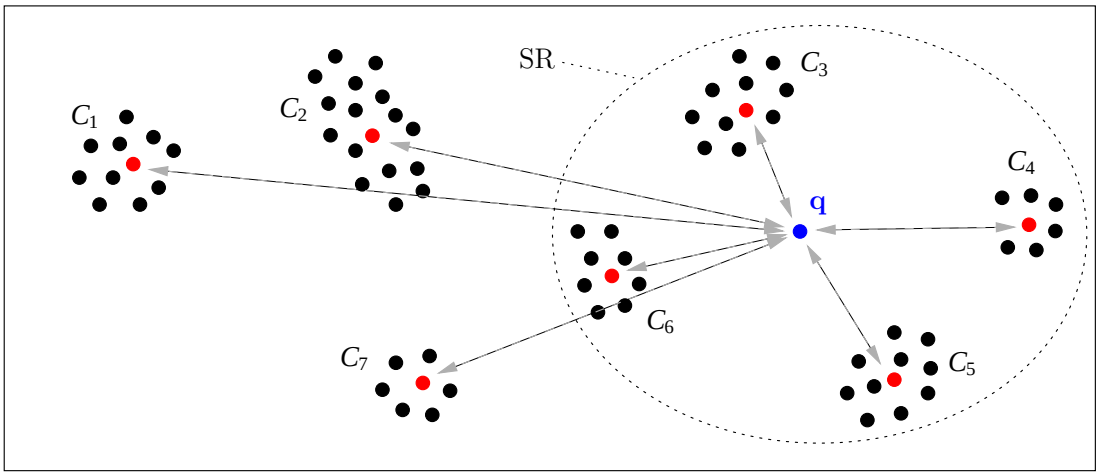


Abb. 5.5: Beispiel für die Selektion des Suchraumes. Die hier dargestellte Datenverteilung wurde in sieben Cluster gruppiert. Ausgehend von einer unteren Schranke $\Omega = 35$, gilt es die Cluster zu bestimmen, die am nächsten zum aktuellen Anfragevektor $\mathbf{q}$ (blau) liegen und die zusammen so viele Elemente besitzen, dass die untere Schranke Ω erreicht bzw. überschritten wird. Zunächst wird daher für jeden Cluster auf der Grundlage seines Clusterprototypen (rot) der Abstand (grauer Pfeil) zu $\mathbf{q}$ berechnet. Daraus resultiert die sortierte Liste $CL = \{C_3, C_5, C_6, C_4, C_7, C_2, C_1\}$, in der das linke Element C_3 den kleinsten und das rechte Element C_1 den größten Abstand besitzt. Als nächstes werden so viele der nahegelegensten Cluster zusammengefasst bis $\Omega = 35$ erreicht oder überschritten wird. Für den Suchraum *SR* gilt schließlich: $SR = C_3 \cup C_5 \cup C_6 \cup C_4$ und $\text{card}(SR) = 38 \geq \Omega = 35$.

Die verschiedenen Cluster können schließlich in Abhängigkeit von ihrem Abstand zum aktuellen Anfrageobjekt sortiert werden:

$$CL = \{C_l^1, C_l^2, \ldots, C_l^L\},$$

mit $D(C_l^1, \mathcal{Q}) \leq D(C_l^2, \mathcal{Q}) \leq \cdots \leq D(C_l^L, \mathcal{Q})$. Die Sortierung bildet die Grundlage für den folgenden Selektionsschritt. Dieser basiert auf einer unteren Schranke Ω, die ein Maß für die Größe des eingeschränkten Suchraumes darstellt. Es werden schließlich so viele der Cluster $C_l \in CL$ zusammengefasst, bis die Anzahl ihrer Elemente diese Schranke erreicht bzw. überschreitet:

$$SR = \bigcup_{i=1}^{L_{SR}} C_l^i, \quad \text{mit} \quad \sum_{i=1}^{L_{SR}} \text{card}(C_l^i) = N_{SR} \quad \text{und} \quad N_{SR} \geq \Omega \tag{5.8}$$

Dabei bezeichnet N_{SR} die Größe des resultierenden Suchraumes *SR*. Außerdem ist zu beachten, dass mit den Elementen der Cluster $\{C_l^i | i = 1, 2, \ldots, L_{SR} - 1\}$ die untere Schranke Ω für die Suchraumgröße noch nicht erreicht wird. Was dieser Selektionsschritt anschaulich bedeutet, demonstriert das Beispiel in Abbildung 5.5.

Die formalen Ausführungen demonstrieren wie, ausgehend von der Gruppierung der gespeicherten Bildobjekte, ein hierarchischer (zweistufiger) Suchprozess entwickelt

werden kann, in dem das Anfrageobjekt lediglich mit einer Teilmenge der Bildobjekte verglichen wird. Der entsprechende Rechenaufwand kann wie folgt abgeschätzt werden:

$$\begin{aligned} T_{\text{cluster}} &= T_{\text{suchraum}} + T_{\text{kompsuche}} \\ &= \underbrace{L\, T_{\text{objdist}} + O(L \log_2 L)}_{\text{Selektion des Suchraumes}} + \underbrace{N_{SR}\, T_{\text{objdist}} + O(N_{SR} \log_2 N_{SR})}_{\text{komplette Suche}} \end{aligned} \tag{5.9}$$

Die Rechenzeit des hierarchischen Suchprozesses setzt sich demnach aus der Rechenzeit zweier Verarbeitungsschritte zusammen. Einerseits aus der Zeit, die notwendig ist, um den eingeschränkten Suchraum zu selektieren (T_{suchraum}). Andererseits aus der Rechenzeit, die benötigt wird, um diesen eingeschränkten Suchraum komplett zu durchsuchen ($T_{\text{kompsuche}}$). Der in Gleichung 5.9 formulierte Rechenaufwand ist im Gegensatz zur Aufwandsabschätzung in Gleichung 5.6 nicht mehr direkt von der Anzahl K der gespeicherten Bildobjekte abhängig, sondern lediglich nur noch indirekt über die Anzahl L der Cluster. Da sowohl die Anzahl L der Bildobjektcluster als auch die resultierende Suchraumgröße N_{SR} viel geringer als die Anzahl K der gespeicherten Bildobjekte ist, gilt für die entsprechenden Rechenzeiten einer Suchiteration mit und ohne Suchraumeinschränkung:

$$T_{\text{cluster}} \ll T_{\text{komplett}}$$

Somit wurde gezeigt, dass durch die Gruppierung der gespeicherten Bildobjekte eine Skalierung des Suchprozesses auf große Datenmengen erzielt werden kann. Auf die entsprechenden Ausführungen für den Suchprozess basierend auf separaten Bildrepräsentanten (vgl. Abschnitt 4.1) wird verzichtet, da diese Betrachtung den Rahmen dieser Arbeit übersteigen würde. Die entsprechenden Bildobjekte werden in den verschiedenen Merkmalsräumen zwar analog zu den bisherigen Ausführungen organisiert, allerdings erfordert die Kombination der Einzelergebnisse der unterschiedlichen Merkmalsräume eine spezielle Verarbeitung. Diese Aufgabenstellung sollte daher in weiterführenden Arbeiten genauer untersucht werden.

Nachdem in diesem Abschnitt der zeitliche Aufwand der Bildersuche mit und ohne Gruppierung der Bildobjekte theoretisch betrachtet wurde, wird im kommenden Abschnitt die technische Umsetzung der multidimensionalen Indizierung im INDI System beschrieben. Dabei wird ausgehend von einer Beschreibung der essentiellen Datenbanktabellen erläutert, wie durch einfache Erweiterung der Datenbankanfrage der Suchraum eines Suchschritts der Bildersuche eingeschränkt werden kann.

5.4.2 Technische Umsetzung

Indizierungsmechanismen sind gewöhnlich ein fester Bestandteil eines Datenbankmanagementsystems. Auch das als Datenbank-Backend (vgl. Abbildung 3.2 auf S. 42) verwendete MySQL besitzt seit Version 3.22 einen derartigen Mechanismus. Dabei

kann ein Index sowohl für eine als auch für mehrere Spalten einer Datenbanktabelle angelegt werden. Die entsprechenden Werte der indizierten Tabellenspalten werden schließlich in einer Baumstruktur gespeichert. Ausgehend von dieser Datenorganisation können die zur Beantwortung der Datenbankanfrage notwendigen Zugriffe auf eine Datenbanktabelle reduziert und somit kurze Antwortzeiten erzielt werden. Obwohl MySQL neben der Indizierung von Datentypen wie beispielsweise Integerwerten und Strings auch die Indizierung von sogenannten *Binary Large Objects* (BLOBs)[15] ermöglicht, erfüllt das Datenbankprodukt nicht die Voraussetzungen, um die Bildobjekte auf der Grundlage ihrer Repräsentanten zu gruppieren. Die für diese Gruppierung notwendigen Verfahren zur multidimensionalen Indizierung werden von MySQL bislang nicht unterstützt.[16] Seit MySQL Version 4.1 besteht zwar auch die Möglichkeit, geographische Daten wie Punkte räumlich zu indizieren (R-Trees), allerdings ist diese Punktdarstellung auf zwei Dimensionen (x- und y- Koordinate) beschränkt und kann daher nicht zur Speicherung der hochdimensionalen Bildbeschreibungen verwendet werden.

Um die in dem vorherigen Abschnitt vorgestellte Suchraumeinschränkung zu ermöglichen, wurde daher ein eigener Indizierungsmechanismus entwickelt. Der entsprechende Indizierungsprozess besteht aus zwei wesentlichen Verarbeitungsschritten, von denen der eine offline im Rahmen der Datenbankinitialisierung und der andere zur Laufzeit durchgeführt wird. Während der erstgenannte zur Organisation der gespeicherten Bilder dient, ist die Aufgabe des zweiten Verarbeitungsschritts, den eingeschränkten Suchraum zu selektieren (vgl. Gleichung 5.9). Die Datenbankorganisation basiert auf einer Gruppierung der gespeicherten Datenmenge. Da für jedes Bild eine Menge von Bildrepräsentanten berechnet wird (vgl. Abschnitt 4.1), bietet es sich an, die Bilder in den verschiedenen Merkmalsräumen mit einem geeigneten Clusterverfahren zu gruppieren. Für jeden Bildrepräsentanten existiert somit eine Gruppierung, die zur Suchraumeinschränkung in dem entsprechenden Merkmalsraum die folgenden Informationen bereitstellt:

- die Prototypen der verschiedenen Bildobjektcluster,
- die Klassifikation der gespeicherten Bildobjekte zu den verschiedenen Clustern und die entsprechenden Abstände zu den jeweiligen Clusterprototypen, sowie
- die Anzahl der Elemente der verschiedenen Cluster.

[15] Die für das INDI System essentiellen Bildrepräsentanten der zu verwaltenden Bildobjekte werden als BLOBs in der Datenbank gespeichert.

[16] Inwieweit andere Datenbankprodukte diese Indizierungsmechanismen bereitstellen, wurde nicht weiter untersucht, da MySQL als Datenbank-Backend des INDI Systems ansonsten bislang alle Anforderungen erfüllt hat. Außerdem stellte die multidimensionale Indizierung keine der zentralen Anforderungen an das INDI System dar (vgl. Systemanforderungen auf S. 43), sodass der Aspekt der multidimensionalen Indizierung bei der Wahl des Datenbankprodukts nicht berücksichtigt wurde.

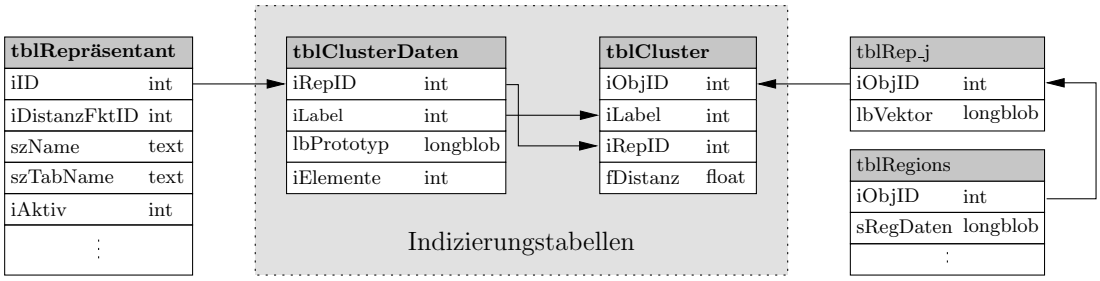


Abb. 5.6: Tabellenmodell zur multidimensionalen Indizierung. Die für die Einschränkung des Suchraumes erforderlichen Gruppierungsdaten werden in den Tabellen `tblClusterDaten` und `tblCluster` gespeichert. Die Tabelle `tblClusterDaten` beinhaltet die Gruppierungs-spezifischen Daten, wie z.B. die jeweiligen Clusterprototypen einer Gruppierung oder die Label der entsprechenden Cluster. Die Zuordnung einer Bildgruppierung zum entsprechenden Bildrepräsentanten ist durch die eingezeichnete Fremdschlüsselbeziehung zwischen den Tabellen `tblRepräsentant` und `tblClusterDaten` sichergestellt. Welches Bildobjekt in den verschiedenen Merkmalsräumen zu welchem Cluster klassifiziert wird, kann auf der Grundlage der systemweit eindeutigen Bildobjekt-Identifikationsnummer (`iObjID`), dem Clusterlabel (`iLabel`) sowie der Repräsentanten-Identifikationsnummer (`iRepID`) in der Tabelle `tblCluster` festgestellt werden.

Da diese Daten die Grundlage der zur Laufzeit durchgeführten Suchraumeinschränkung bilden, werden sie dauerhaft in der Datenbank gespeichert. Die entsprechende Tabellenstruktur ist in Abbildung 5.6 dargestellt. Dabei beinhalten die Tabellen `tblClusterDaten` und `tblCluster` die benötigten Indizierungsinformationen. Die erstgenannte Tabelle ist neben dem entsprechenden Label des Clusters (`iLabel`), den binär gespeicherten Clusterprototypen (`lbPrototyp`) und der Anzahl der Clusterelemente (`iElemente`) durch eine Identifikationsnummer (`iRepID`) gekennzeichnet. Diese Nummer entspricht der Identifikationsnummer (`tblRepräsentant.iID`) des korrespondierenden Bildrepräsentanten, auf dessen Grundlage die gespeicherten Bildobjekte gruppiert wurden. In der Abbildung ist dies durch die eingezeichnete Fremdschlüsselbeziehung zwischen der zentralen Metadatentabelle `tblRepräsentant` (vgl. Abschnitt 3.1.1) und der Clustertabelle `tblClusterDaten` angedeutet. Alle Datensätze der Tabelle `tblClusterDaten` mit identischer Repräsentantennummer (`iRepID`) beschreiben somit die Gruppierung der Bilddaten für den entsprechenden Bildrepräsentanten.

Die korrespondierende Klassifikation der gespeicherten Bildobjekte wird in der Tabelle `tblCluster` abgelegt. Ein Datensatz dieser Tabelle ist durch die systemweit eindeutige Identifikationsnummer des entsprechenden Bildobjekts (`iObjID`), ein Clusterlabel (`iLabel`), die Repräsentantennummer (`iRepID`) sowie die Distanz (`fDistanz`) des Merkmalsvektors (`tblRep.lbVektor`) zum jeweiligen Clusterprototypen (`lbPrototyp`) definiert.

Mit der a priori durchgeführten Organisation der Bildobjekte und der Speicherung der entsprechenden Gruppierungsdaten in den Indizierungstabellen sind die Voraussetzungen für die Einschränkung des Suchraumes gegeben. Zur Laufzeit werden die Bildgruppen bestimmt, deren Elemente dem aktuellen Anfrageobjekt ähneln und den aktuellen Suchraum bilden. Dementsprechend werden die für die Bildersuche notwendigen Bildvergleiche auf die entsprechenden Bildgruppen beschränkt. Wie bereits in Abschnitt 3.1.3 demonstriert wurde, kann der Abstand der gespeicherten Bildobjekte zum Anfrageobjekt auf der Grundlage eines bestimmten Bildrepräsentanten[17] durch einen einzigen SQL Aufruf bestimmt werden. Wird vorausgesetzt, dass der gewichtete euklidische Abstand (`EDistanz`) als Abstandsmaß gewählt wird und außerdem ein Gewichtsvektor (`GewichtsVektor`) gegeben ist, so kann der Abstand zwischen einem Anfragevektor (`AVektor`) und den in der Tabelle (`tblRep_j`) gespeicherten Merkmalsvektoren (`lbVektor`) der verschiedenen Bildobjekte wie folgt berechnet werden:

```
SELECT tblRep_j.iObjID,
       EDistanz('<AVektor>',tblRep_j.lbVektor,'Dim',
                '<GewichtsVektor>','GewichtsDim')
       AS Distanz FROM tblRep_j
```

Zur Verarbeitung wird die entsprechende Repräsentantentabelle `tblRep_j` komplett gescannt. Dabei wird jeder gespeicherte Merkmalsvektor `lbVektor` mit dem entsprechenden Deskriptor `AVektor` des aktuellen Anfrageobjektes $\mathcal{Q}$ verglichen. Dementsprechend existiert in dem jeweiligen Merkmalsraum für jedes Bildobjekt $\mathcal{O}_k$ ein Distanzwert $D_j(\mathcal{O}_k, \mathcal{Q})$ (vgl. Abschnitt 4.2). Um eine Einschränkung des Suchraumes zu erzielen, sollten die Abstandsberechnungen auf die Bildobjekte beschränkt werden, die zu den Gruppen gehören, die in unmittelbarer Nähe des Anfrageobjekts liegen. Dies lässt sich durch eine einfache Erweiterung der SQL Anfrage erzielen. Doch zuvor müssen die Cluster bestimmt werden, die unter Berücksichtigung der unteren Schranke Ω (vgl. Abschnitt 5.4.1) und der entsprechenden Einträge der Indizierungstabellen für das aktuelle Anfrageobjekt relevant sind. Dazu wird im entsprechenden Merkmalsraum zunächst der aktuelle Anfragevektor mit den Prototypen der korrespondierenden Bildobjektgruppierung verglichen:

```
SELECT tblClusterDaten.iLabel, tblClusterDaten.iElemente,
       EDistanz('<AVektor>',tblClusterDaten.lbPrototyp,'Dim',
                '<GewichtsVektor>','GewichtsDim') AS Distanz
       FROM tblClusterDaten WHERE iRepID = j ORDER BY DISTANZ
```

Als Anwort auf die Anfrage liefert das Datenbanksystem eine Ergebnisliste zurück, deren Datensätze die folgenden Elemente beinhalten: ein Clusterlabel, die Anzahl der Clusterelemente und den Abstand des Clusterprototypen zum aktuellen

[17] In den folgenden Ausführungen wird davon ausgegangen, dass die Bildersuche auf der Grundlage des *j*-ten Bildrepräsentanten erfolgt. Die entsprechende Repräsentantentabelle ist die in Abbildung 5.6 dargestellte Tabelle `tblRep_j` (vgl. auch Entity-Relationship-Diagramm auf S. 46).

Anfragevektor. Da die Datensätze entsprechend ihres Abstandes zum jeweiligen Anfragevektor sortiert sind (`ORDER BY DISTANZ`), können die Bildobjektgruppen bestimmt werden, die für das gegebene Anfrageobjekt relevant sind und deren Elemente ausreichen, um die untere Schranke Ω zu erreichen. Darauf aufbauend kann der abschließende Suchschritt des zweistufigen Suchprozesses durch die Formulierung einer zusätzlichen Bedingungsklausel auf die entsprechenden Gruppen eingeschränkt werden:

```
SELECT tblRep_j.iObjID,
       EDistanz('<AVektor>',tblRep_j.lbVektor,'Dim',
       '<GewichtsVektor>','GewichtsDim') AS Distanz
       FROM tblRep_j WHERE (tblRep_j.iObjID = tblCluster.iObjID
                       AND tblCluster.iRepID = j
                       AND (tblCluster.iLabel = l1
                         OR tblCluster.iLabel = l2
                         ...
                         OR tblCluster.iLabel = lN))
```

Diese Anfrage entspricht bis zur `WHERE` Klausel exakt der Anfrage, die für das komplette Durchsuchen der Datenbank formuliert wird. Durch die zusätzliche Formulierung der WHERE Bedingung wird dieser Suchprozess auf die Bildgruppen beschränkt, die sowohl zu dem entsprechenden Bildrepräsentanten (`tblCluster.iRepID = j`) als auch zu den spezifizierten Clustern (`tblCluster.iLabel =` l_1 OR ... OR `tblCluster.iLabel =` l_n) gehören. Die Verknüpfung der Indizierungstabelle `tblCluster` und der Repräsentantentabelle `tblRep_j` erfolgt dabei über die Bedingung `tblRep_j.iObjID = tblCluster.iObjID` (vgl. Abbildung 5.6).

5.5 Evaluation

Mit der Aufwandsanalyse in Abschnitt 5.4.1 wurde demonstriert, dass die Rechenzeit eines Suchschritts direkt von der Menge der gespeicherten Bilder abhängt. Dementsprechend ist der in Kapitel 4 vorgestellte inhaltsbasierte Suchprozess nur dann auf große Datenmengen skalierbar, wenn die gespeicherte Bildmenge nicht linear, sondern hierarchisch durchsucht wird, sodass eine Einschränkung des Suchraumes erzielt werden kann. Wie diese Suchraumeinschränkung im INDI System umgesetzt ist, wurde in Abschnitt 5.4.2 beschrieben. Die bisherigen Ausführungen konzentrierten sich allerdings auf die technischen Details des Indizierungsverfahrens. Im Folgenden wird der auf dem eingeschränkten Suchraum basierende Suchprozess genauer analysiert.

5.5.1 Ziel der experimentellen Untersuchungen

Der Schritt eines Bildsuchsystems vom Forschungsprototypen hin zur industriellen Anwendung erfordert, dass die inhaltsbasierten Techniken auf große Datenbestände skalierbar sind. Das Ziel der folgenden experimentellen Untersuchungen ist es daher zu analysieren, wie sich die Qualität der Bildersuche des INDI Systems mit der Einschränkung des Suchraumes verändert. Im Gegensatz zu anderen Arbeiten (vgl. z.B. [AM98] oder [Le 02]) wird dabei auch untersucht, welchen Einfluss die Interaktion von Mensch und System und der damit verbundene Lernprozess des Bildsuchsystems auf die Ergebnisse hat. Dies ist besonders deshalb interessant, da die a priori durchgeführte Indizierung der gespeicherten Bildobjekte auf Clusterverfahren basiert, die zur Gruppierung der Bildobjekte ein bestimmtes Abstandsmaß verwenden, wie z.B. die in Abschnitt 2.3.1 beschriebenen Minkowski Metriken. Durch das Systemlernen innerhalb des Suchprozesses wird aber in der Regel genau dieses Abstandsmaß modifiziert, sodass eine Adaption an die Suchintention eines Benutzers erzielt werden kann (vgl. Abschnitt 4.3). Ideal wäre es natürlich, wenn die Datenbankorganisation auf der Grundlage des neu gelernten Abstandsmaßes aktualisiert werden könnte. Dies ist aber aufgrund des damit verbundenen Rechenaufwands bei umfangreichen Datenbeständen zumindest zur Laufzeit nicht möglich. Die Experimente sollen deshalb Aufschluss darüber bringen, inwieweit es sinnvoll ist, den adaptiven Suchprozess auf eine derartige Datenbankorganisation zu stützen.

5.5.2 Evaluationsdatenbank und Merkmalsextraktion

Die Grundlage der experimentellen Untersuchungen bildet ebenso wie in Abschnitt 4.4 eine Teilmenge der Fotokollektion der „ArtExplosion® 600000 Images“ Bildsammlung der Nova Development Corporation. Die Evaluationsdatenbank besteht aus 15000 Bildern, die fünfzehn unterschiedliche Bilddomänen (vgl. Abbildung 5.7) repräsentieren und sich gleichmäßig auf diese Bildkategorien verteilen.

Für jedes der gespeicherten Bilder werden verschiedene Charakteristika der Merkmalsklasse Farbe berechnet. Neben den bereits in Abschnitt 4.4 vorgestellten Farbrepräsentanten $\mathbf{r}_{\text{Fuzzyhistogramm}}$ und $\mathbf{r}_{\text{Farbmomente}}$ wird ein weiterer Deskriptor verwendet, der das Farblayout eines Bildes beschreibt. Ausgangspunkt der Berechnung dieses Merkmalsvektors ist die in Anhang B beschriebene Farbdarstellung im CIE $L^*u^*v^*$ Farbraum sowie die Partitionierung des zu beschreibenden Bildes in zwei überlappende 3×3 und 5×5 Gitter. In jedem der daraus resultierenden Bildraster werden die Mittelwerte der drei Farbkanäle berechnet. Die verschiedenen Farbmittelwerte der Bildraster werden schließlich zu einem Gesamtvektor $\mathbf{r}_{\text{Farblayout}}$ kombiniert, der entsprechend der Bildpartitionierung und der Dimension des Farbraumes aus 102 Komponenten besteht. Für die experimentellen Untersuchungen wird auf eine separate Repräsentation

Abb. 5.7: Übersicht über die fünfzehn Bildkategorien der Evaluationsdatenbank

der verschiedenen Farbcharakteristika verzichtet, da diese die Kombination verschiedener Suchraumergebnisse erfordern würde. Die Kombination der Teilergebnisse stellt allerdings eine komplexere Aufgabe dar, die nicht im Fokus dieser Arbeit liegt. Für die Analyse des Systemverhaltens mit und ohne Datengruppierung wird daher ein kombinierter Bildrepräsentant für jedes Bild der Datenbank verwendet. Die verschiedenen Farbbeschreibungen werden dementsprechend zu einem Bildrepräsentanten $\mathbf{r}_{\text{Farbe}}$ kombiniert:

$$\mathbf{r}_{\text{Farbe}} = \mathbf{r}_{\text{Fuzzyhistogramm}} \oplus \mathbf{r}_{\text{Farbmomente}} \oplus \mathbf{r}_{\text{Farblayout}}, \quad \mathbf{r}_{\text{Farbe}} \in \mathbb{R}^{169}$$

Jedes Bild der Datenbank wird demnach exakt durch einen Merkmalsvektor beschrieben. Da zu erwarten ist, dass dieser redundante Information enthält und er außerdem mit 169 Komponenten für die adaptive inhaltsbasierte Bildersuche sehr hochdimensional ist, werden die Bildrepräsentanten der gespeicherten Bilder durch das Verfahren der Hauptachsentransformation (vgl. Anhang C) in ihrer Dimension reduziert. Der Grad der Dimensionsreduktion wird so gewählt, dass 90% des ursprünglichen Informationsgehaltes des Datenraumes erhalten bleibt. Daraus hat sich ein 27-dimensionaler Eigenraum ergeben, in den die Merkmalsvektoren transformiert werden, $\mathbf{r}'_{\text{Farbe}} \in \mathbb{R}^{27}$. Abgeschlossen wird die Optimierung der Merkmalsvektoren durch die Normierung der Varianzanteile der verschiedenen Vektorkomponenten (siehe S. 100).

5.5.3 Auswahl des Clusterverfahrens

Wie in Abschnitt 5.4.1 beschrieben wurde, basiert die multidimensionale Indizierung bzw. die Einschränkung des Suchraumes im INDI System auf einer Gruppierung der

gespeicherten Bildobjekte. In diesem Abschnitt erfolgt daher die Auswahl des Clusterverfahrens, dass sich für die gegebene Datenmenge und den entsprechenden Merkmalsvektoren am besten zur Datenbankorganisation eignet. Bei den hier betrachteten Clusteralgorithmen handelt es sich um den Lloyd- (Lloyd) sowie den Neural-Gas-Algorithmus (NG) (vgl. Abschnitt 5.2). Da mit dem Verfahren Neural-Gas bereits ein neuronales Clusterverfahren untersucht wird, das darüber hinaus Neuronennachbarschaften direkt im Eingabedatenraum und nicht in einem zu spezifizierenden Topologieraum definiert, wird auf eine zusätzliche Betrachtung der in Abschnitt 5.2 vorgestellten selbstorganisierenden Karten verzichtet.

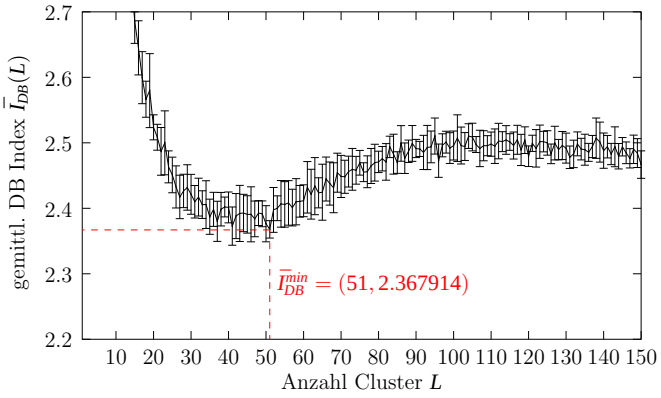
In den experimentellen Untersuchungen wird zum einen die für die gegebene Datenmenge optimale Clusteranzahl der beiden Clusterverfahren bestimmt und zum anderen werden die Ergebnisse beider Verfahren miteinander verglichen. Aufbauend auf dem Vergleich kann schließlich das Quantisierungsverfahren bestimmt werden, das sich am besten zur Organisation der Evaluationsdatenbank eignet. Um die optimale Clusteranzahl zu bestimmen wird jeder der beiden Clusteralgorithmen mit unterschiedlichen Parametrisierungen für die Anzahl L der Cluster durchgeführt. Dabei wird die Clusteranzahl von $L_{\text{min}} = 2$ bis $L_{\text{max}} = 150$ variiert. Dies entspricht in etwa der von Maulik und Bandyopadhyay [Mau02] vorgeschlagenen Strategie zur Bestimmung einer adäquaten Clusteranzahl, die ihr Parametrisierungsintervall für L durch die minimale Schranke $L_{\text{min}} = 2$ und die maximale Schranke $L_{\text{max}} = \sqrt{K}$ begrenzen. Für die $K = 15000$ Bilder umfassende Evaluationsdatenbank würde somit für die obere Schranke $L_{\text{max}} \approx 122$ gelten. Die Qualität der in Abhängigkeit von der Clusteranzahl erzeugten Datengruppierung wird anhand des in Abschnitt 5.3 beschriebenen Davies-Bouldin-Index I_{DB} (DB-Index) bewertet. Da in beiden Verfahren allerdings die initialen Codebuchvektoren zufällig bestimmt werden und die Qualität der resultierenden Gruppierung signifikant von der Güte der initialen Clusterprototypen abhängt, wird jeder Algorithmus für eine gegebene Clusteranzahl L zehnmal wiederholt. Die resultierenden Qualitätsmaße werden schließlich gemittelt. Der Lloyd-Algorithmus wird in den Experimenten sowohl für 100 als auch für 500 Lernepochen N_E durchgeführt. Für den Neural-Gas-Algorithmus werden $N_E = 100$ Lernepochen verwendet.[18] Die übrigen Parameter des Verfahrens, wie die Verzögerung λ und die Lernrate α, werden für einen Lernschritt t wie folgt berechnet:

$$\lambda(t) = \lambda_s(0.01/\lambda_s)^{t/t_{\text{max}}} \quad \text{und} \quad \alpha(t) = \alpha_s(0.005/\alpha_s)^{t/t_{\text{max}}}$$

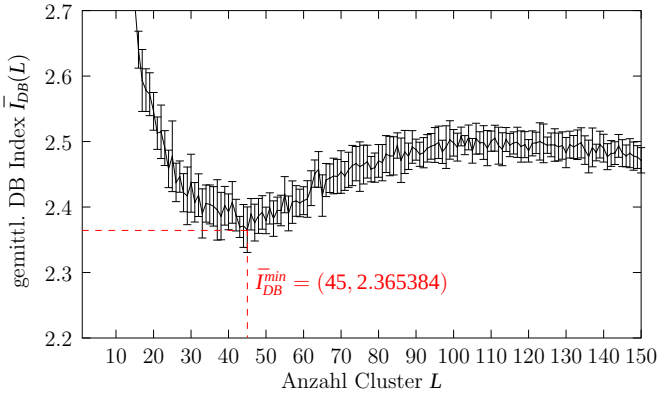
Dabei gilt für die maximale Anzahl der Lernschritte $t_{\text{max}} = N_E K = 1500000$ sowie für $\lambda_s = L/2$ und $\alpha_s = 0.5$. In beiden Verfahren wird als Abstandsmaß der quadrierte euklidische Abstand $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$ verwendet.

Die Ergebnisse der experimentellen Untersuchungen zur Bestimmung der optimalen Clusteranzahl sind in Abbildung 5.8 dargestellt. In den entsprechenden Diagrammen

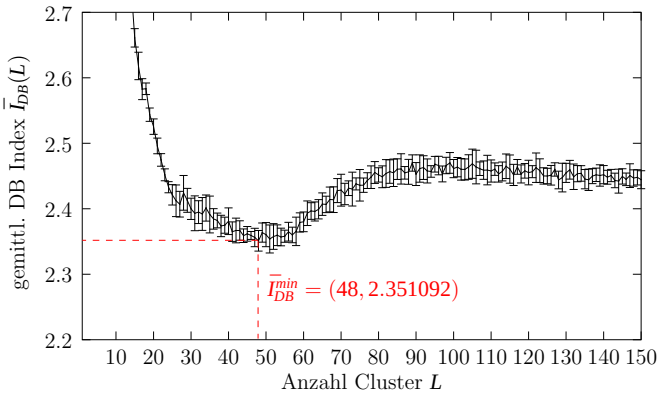
[18] Dies ist ausreichend, da der Neural-Gas-Algorithmus in den experimentellen Untersuchungen von Martinetz et al. [Mar93] ein gutes Konvergenzverhalten gezeigt hat.



(a) gemittelter DB-Index für Lloyd (100)



(b) gemittelter DB-Index für Lloyd (500)



(c) gemittelter DB-Index für NG (100)

Abb. 5.8: Optimale Clusteranzahl für den Lloyd- und den Neural-Gas-Algorithmus

sind für die verschiedenen Clusterverfahren (Lloyd (100), Lloyd (500) und NG (100)) die gemittelten DB-Indizes in Abhängigkeit von der Clusteranzahl L aufgetragen. In jeder der drei Kurven ist ein eindeutiges Minimum zu erkennen, das die jeweilige optimale Clusteranzahl kennzeichnet. Die Cluster der entsprechenden Gruppierung sind im Vergleich zu den übrigen Gruppierungen am besten voneinander separiert. Für den Lloyd-Algorithmus mit 100 Lernepochen liegt die optimale Clusteranzahl bei $L_{\text{opt}} = 51$. Wird das Verfahren mit 500 Epochen durchgeführt gilt für die optimale Clusteranzahl $L_{\text{opt}} = 45$. Das entsprechende Extremum des Neural-Gas-Algorithmus liegt bei $L_{\text{opt}} = 48$. Zusätzlich zu den gemittelten Werten ist auch die entsprechende Standardabweichung der Qualitätsmaße für die zehn Wiederholungen dargestellt. Dabei fällt auf, dass die Güteindizes für das Verfahren des Neural-Gas weniger streuen als die entsprechenden Werte des Lloyd-Algorithmus. Das Verfahren nach Lloyd scheint dementsprechend stärker von der Auswahl der initialen Prototypen abzuhängen als der Neural-Gas-Algorithmus.

	gemittelter Davies-Bouldin-Index		
L_{opt}	Lloyd (100)	Lloyd (500)	NG (100)
51	**2.3679 (±0.0133)**	2.3979 (±0.0238)	2.3539 (±0.0214)
45	2.3907 (±0.0285)	**2.3654 (±0.0348)**	2.3618 (±0.0112)
48	2.3927 (±0.0189)	2.3864 (±0.0266)	**2.3511 (±0.0157)**
	gemittelter Quantisierungsfehler		
L_{opt}	Lloyd (100)	Lloyd (500)	NG (100)
51	**0.6377 (±0.0007)**	0.6378 (±0.0009)	0.6369 (±0.0002)
45	0.6443 (±0.0006)	**0.6439 (±0.0007)**	0.6435 (±0.0001)
48	0.6409 (±0.0008)	0.6404 (±0.0007)	**0.6399 (±0.0002)**

Tabelle 5.1: Vergleich der beiden Quantisierungsverfahren Lloyd und NG anhand des gemittelten Davies-Bouldin-Index und des gemittelten Quantisierungsfehlers. Die entsprechenden Standardabweichungen der verschiedenen Messungen sind in Klammern angegeben. Die fett gedruckten Einträge kennzeichnen das Clusterverfahren, für das die jeweilige optimale Clusteranzahl der Zeile gilt.

Um die verschiedenen Verfahren miteinander vergleichen zu können, werden die unterschiedlichen Qualitätsmaße für die jeweils optimale Clusteranzahl in eine Tabelle eingetragen (vgl. Tabelle 5.1). Für die optimale Clusteranzahl L_{opt} eines Verfahrens wird dabei nicht nur das entsprechende Qualitätsmaß des jeweiligen Algorithmus betrachtet, sondern ebenfalls die korrespondierenden Qualitätsmaße der anderen Verfahren. Neben dem gemittelten Davies-Bouldin-Index wird auch der gemittelte Quantisierungsfehler berücksichtigt. Obwohl derNeural-Gas-Algorithmus sowohl für den Davies-Bouldin-Index als auch für den Quantisierungsfehler immer die besten Resultate erzielt, können keine deutlichen Unterschiede erkannt werden. Am auffälligsten sind die Unterschiede noch bei der für das NG Verfahren optimalen Clusteranzahl

von $L_{\text{opt}} = 48$ und dem Davies-Bouldin-Index (dritte Ergebniszeile). Im Vergleich zum Lloyd-Algorithmus mit 100 Lernepochen kann mit dem Neural-Gas-Algorithmus eine relative Verbesserung von 1.74% erzielt werden. Gegenüber dem Lloyd Verfahren mit 500 Epochen beträgt die relative Verbesserung 1.48%. Die entsprechenden relativen Verbesserungen des durchschnittlichen Quantisierungsfehlers in der letzten Zeile der Tabelle sind mit 0.16% und 0.08% viel geringer. Bis auf eine Ausnahme in der ersten Ergebniszeile von Tabelle 5.1 können für den Lloyd-Algorithmus mit 500 Lernepochen marginal bessere bzw. vergleichbare Resultate erzielt werden als mit 100 Epochen. Aufgrund der größeren Robustheit gegenüber der Auswahl der initialen Clusterprototypen sowie der insgesamt leicht besseren Ergebnisse wird zur Organisation der Evaluationsdatenbank der Neural-Gas-Algorithmus verwendet.[19] Entsprechend der optimalen Clusteranzahl für dieses Verfahren werden die Bildobjekte der Evaluationsdatenbank somit in 48 Cluster gruppiert (vgl. Abbildung 5.8(c)).

5.5.4 Bildersuche mit und ohne Suchraumeinschränkung

Mit der Auswahl eines geeigneten Clusterverfahrens sowie der entsprechenden Gruppierung der gespeicherten Bildobjekte wurden die Voraussetzungen für die in diesem Abschnitt beschriebenen experimentellen Untersuchungen geschaffen. Dabei wird analysiert, inwieweit sich die Suchergebnisse der eingeschränkten Bildersuche von denen des kompletten Suchprozesses unterscheiden. Ebenso wie in Kapitel 4.4 wird als Anfrageparadigma die Kategoriesuche gewählt. Dazu werden 100 Beispielbilder aus fünf verschiedenen Domänen (*Blumen*, *Wüste*, *Wolkenhimmel*, *Stadt und Land* sowie *Unter Wasser*) der Evaluationsdatenbank ausgewählt (20 Bilder pro Domäne). Die Ergebnismenge umfasst 60 Bilder und es werden pro Beispielsuche vier Suchschritte durchgeführt. Dementsprechend besteht der Suchprozess aus einem initialen Auswahlschritt sowie drei Bewertungsiterationen. In jedem der Bewertungsschritte werden die Bilder der Ergebnismenge als relevant bewertet, die zu derselben Bildkategorie wie das Beispielbild gehören. Ausgehend vom kombinierten Bildrepräsentanten fokussiert der Lernprozess auf die Verfeinerung der Anfrage $\mathcal{Q}$ sowie die Adaption der Gewichtsmatrix $\mathbf{W}$ (vgl. Abschnitt 4.1 und Abschnitt 4.3.1). Als Abstandsmaß wird der quadrierte generalisierte euklidische Abstand verwendet. Da analog zu Abschnitt 4.2.1 initial die Gewichtsmatrix der Einheitsmatrix entspricht, $\mathbf{W} = \mathbf{E}$, wird im ersten Suchschritt ein quadrierter euklidischer Abstand zur Abstandsberechnung verwendet. Die Metrik des ersten Suchschritts der Bildersuche stimmt dementsprechend mit der Metrik überein, die zur Organisation der Datenbank verwendet wurde (vgl. Abschnitt 5.5.3). Um zu analysieren, wie stark die Qualität der Suchergebnisse mit der Größe des Suchraumes variiert, werden die verschiedenen Beispielsuchen für unterschiedlich große Suchräume durchgeführt. Dabei wird der ursprünglich 15000 Bilder umfassende Such-

[19] Da die Datenbankorganisation offline erfolgt, kann akzeptiert werden, dass der Neural-Gas-Algorithmus rechenintensiver als der Lloyd-Algorithmus ist.

raum auf 5000, 3000, 1500 und 750 Bilder eingeschränkt.[20] Dies entspricht einer prozentualen Einschränkung des Suchraumes auf 33%, 20%, 10% und 5% der gespeicherten Datenmenge.

Im Gegensatz zur Evaluation der verschiedenen Suchverfahren und Adaptionsansätze in Abschnitt 4.4 wird jedoch auf eine Spezifikation eines Groundtruth, der auf der Kategorisierung der gespeicherten Bilder basiert, verzichtet.[21] Diese Entscheidung wird wie folgt begründet: Erstens ist es sehr schwierig für eine derartig umfangreiche Testdatenbank eine semantische Referenzgruppierung zu entwerfen, in der garantiert ist, dass die verschiedenen Bildklassen adäquat beschrieben werden und sich visuell voneinander unterscheiden. Zweitens ist es nicht das Ziel der folgenden Untersuchungen die Lernfähigkeit des Systems zu evaluieren. Diese Zielsetzung wurde bereits mit den in Kapitel 4 beschriebenen Experimenten verfolgt. Stattdessen wird analysiert wie sich die Qualität des inhaltsbasierten Suchprozesses verändert, wenn auf eine komplette Bildersuche verzichtet und der Suchraum eingeschränkt wird. Ausgehend von einer fixen Systemkonfiguration und der damit verbundenen Qualität des Bildsuchsystems können die für eine gegebene Anfrage relevanten Bilder genau dann gefunden werden, wenn der komplette Datenbestand mit der Anfrage verglichen wird. Die aus dieser vollständigen Bildersuche resultierende Ergebnismenge definiert daher den Groundtruth. Sie bildet die Grundlage für die Berechnung der Suchgenauigkeit, die mit der zweistufigen Bildersuche erzielt werden kann. Wird vorausgesetzt, dass der Suchprozess auf dem *i*-ten Beispielbild basiert, dann ist die Suchgenauigkeit, die mit der eingeschränkten Bildersuche im Vergleich zur kompletten Bildersuche erzielt wird durch

$$\Psi_i = \frac{M}{N} 100$$

definiert (vgl. Abdel-Mottaleb et al. [AM98]). Dabei bezeichnet N die Anzahl der Ergebnisbilder, die einem Benutzer präsentiert werden (in den Experimenten ist $N = 60$) und M repräsentiert die Anzahl der Bilder, die sowohl in der Ergebnismenge der kompletten als auch in der Ergebnismenge der zweistufigen Bildersuche enthalten sind. Entsprechend der Definition ist die Suchgenauigkeit nicht anderes als die prozentuale Übereinstimmung zwischen dem Suchergebnis der vollständigen und dem Suchergebnis der eingeschränkten Bildersuche. Deshalb stellen die entsprechenden Suchgenau-

[20] Um in der Evaluation die spezifizierte Suchraumgröße exakt und nicht nur näherungsweise zu erreichen, wurde das in Abschnitt 5.4.1 vorgestellte Verfahren zur Selektion des Suchraumes (vgl. Gleichung 5.8) ein wenig modifiziert. Auf eine detaillierte Erläuterung wird an dieser Stelle allerdings verzichtet.

[21] An dieser Stelle wird darauf hingewiesen, dass die gegebene Kategorisierung der Bilder zwar wie beschrieben zur Bewertung der Ergebnisbilder eines Suchschritts dient, allerdings wird sie nicht zur Auswertung der Beispielsuchen verwendet.

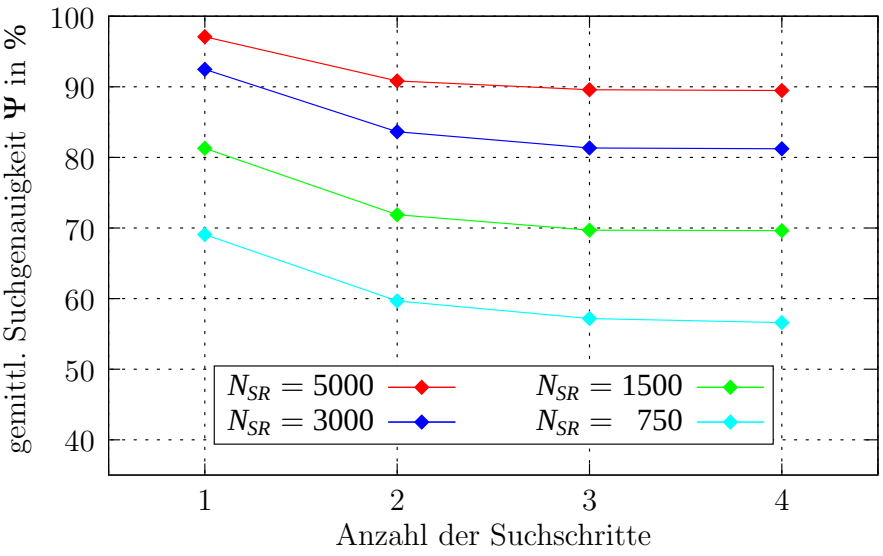


Abb. 5.9: Verlauf der durchschnittlichen Suchgenauigkeit der 100 Beispielsuchen für vier Suchschritte bestehend aus einem initialen Auswahl- und drei Bewertungsschritten. Die verschiedenen Kurven repräsentierten die Ergebnisse für eine Suchraumgröße von 5000, 3000, 1500 und 750 Bildern. Dies entspricht einer Einschränkung des Suchraumes auf 33%, 20%, 10% und 5% der ursprünglichen Datenmenge.

igkeitswerte prozentuale Angaben dar. Die durchschnittliche Suchgenauigkeit für eine Menge von I Beispielbildern berechnet sich demnach wie folgt:

$$\Psi = \frac{1}{I} \sum_{i=1}^{I} \Psi_i$$

Da das Qualitätsmaß ein Referenzergebnis voraussetzt, werden die 100 Beispielsuchen zunächst linear durchgeführt, sodass in jedem Suchschritt das Anfrageobjekt mit allen Bildobjekten der Datenbank verglichen wird. Somit existiert für jeden Suchschritt der verschiedenen Beispielsuchen ein Referenzergebnis. Daran anschließend werden die Beispielsuchen entsprechend der Datenbankorganisation und der spezifizierten Suchraumgröße N_{SR} durchgeführt.

Die aus dem Vergleich der eingeschränkten und kompletten Bildersuchen resultierenden durchschnittlichen Suchgenauigkeiten Ψ sind in Abbildung 5.9 und Tabelle 5.2 dargestellt. Wie die Ergebnisse des ersten Suchschritts in Abbildung 5.9 demonstrieren, nimmt die Suchgenauigkeit mit der Einschränkung des Suchraumes ab. Dieses Verhalten ist zu erwarten, da mit stärkerer Einschränkung in der Regel weniger Gruppen von Bildobjekten durchsucht werden. Daraus resultiert, dass einige Bildobjekte, die zwar in der kompletten Bildersuche gefunden wurden, nicht zu den selektierten Gruppen gehören. Allerdings ist festzustellen, dass bei einer Einschränkung auf 33% der gespeicherten Bildmenge eine durchschnittliche Suchgenauigkeit von 97.10% erreicht werden kann. Bei einer Suchraumgröße von 3000 Bildern sind dies 92.47% und

	Suchschritte (Ψ, σ)							
N_{SR}	1		2		3		4	
5000	97.10	($\pm$3.80)	90.82	($\pm$15.39)	89.58	($\pm$15.78)	89.48	($\pm$15.98)
3000	92.47	($\pm$7.94)	83.63	($\pm$19.21)	81.33	($\pm$19.48)	81.22	($\pm$19.45)
1500	81.28	($\pm$17.00)	71.88	($\pm$23.64)	69.68	($\pm$24.08)	69.62	($\pm$23.64)
750	69.08	($\pm$22.45)	59.67	($\pm$25.80)	57.17	($\pm$26.48)	56.60	($\pm$26.85)

Tabelle 5.2: Durchschnittliche Suchgenauigkeit Ψ (in %) aller Beispielsuchen für vier Suchschritte. In den Klammern sind die jeweiligen Standardabweichungen σ der Messergebnisse dargestellt.

bei einem Suchraum von 1500 Bildern immerhin noch 81.28%. Wenn das Anfrageobjekt nur noch mit 5% der gesamten Datenmenge verglichen wird, wird eine Genauigkeit von 69.08% erzielt. Die in der ersten Ergebnisspalte von Tabelle 5.2 dargestellten Standardabweichungen der Messergebnisse demonstrieren außerdem, dass die Streuung der Ergebnisse mit der Einschränkung des Suchraumes zunimmt.

Wie in Abbildung 5.9 außerdem zu sehen ist, verlaufen die Kurven der verschiedenen Suchräume vom ersten bis zum vierten Suchschritt immer flacher. Die relative Verschlechterung der Suchgenauigkeit beträgt dabei für den Suchraum mit 5000 Elementen 7.85%. Für $N_{SR} = 3000$ beträgt die relative Verschlechterung 12.17% und für $N_{SR} = 1500$ gilt 14.36%. Für eine Suchraumgröße von 750 ist die relative Verschlechterung vom ersten bis zum vierten Suchschritt mit 18.07% schließlich am größten. Der Grund für die Verschlechterung der Suchergebnisse ist die Lernfähigkeit des Systems. Mit der Abgabe der ersten Bewertung beginnt das System zu lernen, indem die Anfrage verfeinert und die Gewichtsmatrix adaptiert wird. Während bei der Anfrageverfeinerung der Anfragevektor lediglich im Merkmalsraum verschoben wird, bewirkt die Adaption der Gewichtsmatrix eine Veränderung des Abstandsmaßes. Da die Elemente der Datenbank auf der Grundlage eines anderen Abstandsmaßes gruppiert wurden, stellt die entsprechende Gruppierung für das adaptierte Abstandsmaß nicht mehr die ideale Datenbankorganisation dar. Lediglich im ersten Suchschritt sind die Abstandsmaße identisch, sodass auch die besten Suchgenauigkeiten erzielt werden können. Obwohl der Lernprozess die Veränderung des Abstandsmaßes bewirkt, können zumindest für eine Suchraumgröße von $N_{SR} = 5000$ und $N_{SR} = 3000$ mit einer Suchgenauigkeit von 89.48% und 81.22% zufriedenstellende Resultate erzielt werden. Allerdings zeigen die Ergebnisse in Tabelle 5.2 auch, dass mit der Anzahl der Suchiterationen auch die Streuungen der Suchgenauigkeiten zunehmen. Eine deutliche Zunahme ist besonders bei dem Übergang vom ersten zum zweiten Suchschritt zu beobachten. Parallel dazu ist auch die Abnahme der durchschnittlichen Suchgenauigkeit am größten. Die experimentellen Untersuchungen in Abschnitt 4.4.4 haben bereits demonstriert, dass das Systemlernen auf der Grundlage eines kombinierten Bildrepräsentanten besonders auf die ersten Bewertungsschritte fokussiert. Die in Tabelle 5.2 dargestellten Resultate bestätigen diese Beobachtungen. Vom ersten zum zweiten Suchschritt lernt das System

am besten, dementsprechend verändert sich das Abstandsmaß am stärksten, so dass die Suchgenauigkeit abnimmt und die Streuung der Suchgenauigkeit zunimmt. In den folgenden Lernschritten lernt das System nur wenig neues, sodass die Suchgenauigkeiten der verschiedenen Untersuchungen sich nur geringfügig verschlechtern.

Insgesamt lässt sich feststellen, dass trotz der teilweise recht starken Einschränkung des Suchraumes noch gute Suchergebnisse erzielt werden können. Dabei konnten vor allem mit dem initialen Suchschritt die besten Resultate erzielt werden. In dem aktuellen Evaluationszenario hat sich sowohl eine Einschränkung des Suchraumes auf $N_{SR} = 5000$ als auch auf $N_{SR} = 3000$ Bilder als sinnvoll erwiesen. Obwohl ausgehend vom Systemlernen das Abstandsmaß variiert, können nach vier Suchschritten für diese Suchräume noch zufriedenstellende Ergebnisse erzielt werden. Bei den experimentellen Untersuchungen mit einer Suchraumgröße von $N_{SR} = 1500$ und $N_{SR} = 750$ wird zwar eine Suchgenauigkeit von 56.60% nicht unterschritten, allerdings ist die Streuung der Ergebnisse deutlich größer als die der anderen Messreihen. Demnach ist eine solche Reduzierung des Suchraumes nicht zu empfehlen.

Allerdings haben die Untersuchungen auch auf die generelle Problematik hingewiesen, die auftritt, wenn ein adaptiver Suchprozess auf eine unüberwacht erzeugte Datenbankorganisation gestützt wird. Dieser beschriebenen Problematik könnte zwar mit einer Anpassung der Clusteranzahl entgegengewirkt werden, allerdings erscheint es vielversprechender, andere Strategien zu verfolgen. Anstatt eine Datenmenge zuvor unüberwacht zu organisieren, wäre es auch vorstellbar, eine Einteilung der Bilder in grobe semantische Klassen überwacht zu trainieren, wie z.B. *Indoor* und *Outdoor* Bilder. Die resultierende Datenbankorganisation stellt dann eine semantische Kategorisierung dar, die sich unüberwacht in der Regel nur äußerst schwer erzielen lässt (vgl. [Käs03b]). Darauf aufbauend könnten dann die Verfahren des Kurzzeitlernens dazu verwendet werden, um innerhalb einer semantischen Bildgruppe zu suchen. Zur Datenbankorganisation sollten dabei verstärkt Verfahren zum Langzeitlernen berücksichtigt werden (vgl. z.B. [Min96] oder [Kos03]). Diese Verfahren versuchen, über einen einzelnen Suchprozess hinaus, semantische Konzepte der Benutzer zu lernen.

5.6 Zusammenfassung

Die Anforderung an inhaltsbasierte Bildsuchsysteme, umfangreiche Datenbestände effizient verwalten zu können, erfordert, dass neben der Entwicklung von innovativen Verfahren zum Systemlernen, neuen Bildbeschreibungen und perzeptiven Abstandsmaßen zukünftig auch die Aspekte der Skalierbarkeit verstärkt berücksichtigt werden. Das aktuelle Kapitel beschäftigte sich daher mit der Indizierung von hochdimensionalen Daten. Aufbauend auf einer einleitenden Motivation für die multidimensionale Datenorganisation wurden mit dem Bucketing Verfahren, Quadtrees, k-d Trees und R-Trees verschiedenen Ansätze vorgestellt, die z.B. in CAD- oder geographischen Informationssystemen zur Indizierung multidimensionaler Datentypen wie Punkte, Linien

oder Rechtecke verwendet werden. Da diese Ansätze allerdings auf niedrigdimensionalere Merkmalsräume beschränkt sind, werden vor allem Verfahren aus der Mustererkennung verwendet, um hochdimensionale Merkmalsvektoren zu strukturieren. Daher wurden mit der Vektorquantisierung, den selbstorganisierenden Karten sowie dem Neural-Gas-Algorithmus drei Verfahren vorgestellt, die sich zur Gruppierung multidimensionaler Bildbeschreibungen eignen.

Aufbauend auf den theoretischen Ausführungen wurde schließlich der für das INDI System entwickelte Indizierungsmechanismus vorgestellt. In einer Aufwandsanalyse des inhaltsbasierten Suchprozesses wurde dabei zunächst demonstriert, wie durch eine zuvor durchgeführte Datenbankgruppierung eine Skalierung der Suchalgorithmen auf umfangreiche Datenbestände erzielt werden kann. Darauf aufbauend wurde die technische Umsetzung der multidimensionalen Indizierung des INDI Systems beschrieben. In der abschließenden Evaluation wurde der entwickelte Ansatz schließlich genauer analysiert. Ausgehend von einer 15000 Bilder umfassenden Evaluationsdatenbank und einem kombinierten Farbrepräsentanten für jedes Bild der Datenbank wurde dabei zunächst ein Clusterverfahren bestimmt, das sich zur Gruppierung der Datenmenge eignet. Dabei wurden der Lloyd- und der Neural-Gas-Algorithmus auf der Basis des Davies-Bouldin-Index und des Quantisierungsfehlers miteinander verglichen. Des Weiteren wurde die für die Gruppierung der gespeicherten Bilder optimale Clusteranzahl bestimmt.

Aufbauend auf den Ergebnissen des ersten Experiments wurden schließlich die Bildersuchen mit und ohne Einschränkung des Suchraumes miteinander verglichen. Die Ergebnisse demonstrieren, dass trotz deutlicher Einschränkung des Suchraumes gute Suchergebnisse erzielt werden können. Allerdings zeigt sich auch, dass mit verstärktem Lernen des Systems die Suchergebnisse immer schlechter werden, die Datenbankorganisation also immer weniger geeignet ist, um die für eine Suche relevanten Bilder der Datenbank zu finden.

6 Zusammenfassung und Ausblick

Heutzutage spielt die einfache Verwaltung großer Bestände von digitalen Bildern in vielen Anwendungsbereichen eine wichtige Rolle. Werbeleute, Journalisten und Designer benötigen den schnellen Zugang zu umfangreichen Bildkatalogen, um Werbungen, Artikel und Entwürfe durch entsprechende Bilder oder Bildelemente visuell hervorzuheben. Aber nicht nur im industriellen Bereich besteht die Anforderung einer organisierten Datenhaltung. Durch die Fortschritte in der Entwicklung elektronischer Geräte, wie z.B. digitale Fotokameras oder Scanner, nimmt die Menge an digitalen Bildern in privaten Haushalten tagtäglich zu. Anwender dieser Geräte werden zunehmend vor die Aufgabe gestellt, die zahlreichen Bilder strukturiert zu verwalten, um einfach in der gespeicherten Bildmenge navigieren zu können.

Seit den 70er Jahren haben sich textbasierte Bilddatenbanksysteme zur Verwaltung digitaler Bilder etabliert. Ihre Grundlage bildet die aufwendige manuelle Erfassung von Bildinhalten; die sogenannte Verschlagwortung. Obwohl textbasierte Bilddatenbanksysteme einen einfachen semantischen Zugang zu einer Menge von digitalen Bildern bieten, erfordern die verschiedenen Nachteile dieses Ansatzes, dass innovative Bildsuchsysteme entwickelt werden. Nachteile eines textbasierten Systems sind beispielsweise die subjektive Prägung der Verschlagwortung sowie der mangelnde visuelle Zugang zur gespeicherten Datenmenge.

Motiviert durch die Probleme verschlagworteter Systeme wurde seit Anfang der 90er Jahre verstärkt die Entwicklung sogenannter inhaltsbasierter Bildsuchsysteme vorangetrieben. Die rein inhaltsbasierte Bildersuche verzichtet auf die Verwendung von textuellen Annotationen und versucht einzig und allein auf der Grundlage visueller Bildmerkmale, die für eine Anfrage relevanten Bilder der Datenbank zu finden.

Da die inhaltsbasierte Bildersuche eine vielversprechende Alternative zur verschlagworteten Bildersuche darstellt, wurde sich in dieser Arbeit ausführlich mit diesem Thema beschäftigt. Den Mittelpunkt bildet dabei das im Rahmen eines BMB+F Verbundprojekts entwickelte Bildsuchsystem INDI. Das Ziel der vorliegenden Arbeit war die Entwicklung der für das INDI System benötigten inhaltsbasierten Suchmechanismen, sodass ein Anwender auf der Grundlage automatisch extrahierter Bildinhalte in der gespeicherten Bildmenge navigieren kann. Entsprechend der Eigenschaften moderner Bildsuchsysteme sollte das zu entwickelnde System dabei lernfähig sein und sich innerhalb des Suchprozesses an einen Anwender adaptieren können. Dazu wurde ein Suchverfahren entwickelt, das aus verschiedenen Komponenten besteht, die eine

Adaption an die Suchintention eines Benutzers ermöglichen. Da in dem Entwurf unterschiedliche Aspekte und Ansätze berücksichtigt wurden, war es Ziel einer ausführlichen Evaluation, die Systemkonfiguration zu bestimmen, mit der das INDI System am leistungsfähigsten ist. Um in dieser Arbeit auch der Fragestellung nachgehen zu können, inwieweit das Suchverfahren und adaptive Techniken im Allgemeinen skalierbar sind, wurde außerdem ein Verfahren zur Einschränkung des Suchraumes entwickelt. Ausgehend von der multidimensionalen Indizierung des gespeicherten Datenbestandes können somit auch für umfangreiche Bildmengen kurze Systemantwortzeiten erzielt werden.

6.1 Zusammenfassung

Zu Beginn dieser Arbeit wurde eine theoretische Grundlage für das Verständnis und die Funktionsweise der inhaltsbasierten Bildersuche geschaffen. Dabei wurden mit der merkmalsbasierten Bildrepräsentation und dem Ähnlichkeitsvergleich durch Abstandsberechnung die elementaren Bestandteile eines inhaltsbasierten Suchprozesses vorgestellt. Die bestehende semantische Lücke zwischen der Bildbetrachtung eines Anwenders und der formalen Bildbeschreibung eines Bildsuchsystems erfordert allerdings, dass ein Bildsuchsystem lernfähig ist. Deshalb wurden sowohl die Besonderheiten dieses Lernprozesses diskutiert als auch aktuelle Ansätze beschrieben, die zum Systemlernen innerhalb einer iterativen Bildersuche verwendet werden. Das größte Problem dieses inkrementellen Lernvorgangs stellt dabei der in der Regel geringe Umfang der Trainingsmenge dar, auf deren Grundlage es sehr schwierig ist, eine gute Generalisierungsfähigkeit eines Bildsuchsystems zu erzielen.

Da der beschriebene Lernprozess auf der Interaktion von Anwender und System basiert, sollte ein Bildsuchsystem nicht nur lernfähig, sondern darüber hinaus auch einfach und intuitiv zu bedienen sein. Das beschriebene und im Rahmen des LOKI Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“ entwickelte Bildsuchsystem INDI ist exakt aus diesen Anforderungen heraus entstanden. Seine wichtigsten Eigenschaften sind die natürliche und intuitive Bedienung sowie die adaptive inhaltsbasierte Bildersuche. Durch die Kombination dieser Eigenschaften unterscheidet sich dieses System deutlich von anderen Bildsuchsystemen, die in der Regel hauptsächlich auf den inhaltsbasierten Suchprozess fokussieren. Die Hauptbestandteile des INDI Systems sind das Datenbank-Backend, die Such- und Konfigurationseinheit sowie der Datenbank-Client, der die eigentliche Benutzerapplikation darstellt. Diese Einheit ist in der Lage, die Modalitäten Sprache und Touchscreen-Gestik zu verarbeiten und ermöglicht somit eine natürliche und intuitive Navigation in der gespeicherten Bildmenge. Die für die sprachliche Referenzierung von Bildregionen notwendige Fusion von Sprache und Bild wird als probabilistischer Dekodierprozess modelliert und erfolgt auf der Grundlage von Bayes-Netzen.

Damit das INDI System in der Lage ist, merkmalsbasiert in einer gespeicherten Bildmenge zu navigieren, wurde ein inhaltsbasiertes Suchverfahren entwickelt. Dabei wurde ausgehend von der separaten Repräsentation der verschiedenen inhärenten Charakteristika eines Bildes zunächst ein formales Suchmodell formuliert. In diesem werden Bilder als Objekte modelliert, die einerseits durch eine binäre Bilddarstellung beschrieben werden und andererseits durch eine Menge von Merkmalsklassen und eine Menge von Merkmalsrepräsentanten charakterisiert werden. Auf der Grundlage dieser Modellierung wurden schließlich mit dem distanz- und rangbasierten Suchansatz zwei Varianten vorgestellt, inhaltsbasiert in einer Menge von Bildern zu suchen. Der anschließend beschriebene Lernprozess ist von der Zielsetzung motiviert, die Abstände der Trainingsbeispiele zum idealen Anfrageobjekt zu minimieren. Ausgehend von einer klassifizierten Stichprobe werden sowohl die Anfragevektoren adaptiert als auch die Gewichtsmatrizen zur Abstandsberechnung sowie die Gewichte der verschiedenen Repräsentanten gelernt. Zusätzlich wurde ein Verfahren zur Adaption der Repräsentantengewichte vorgestellt, das neben positiven auch negative Trainingsbeispiele zum Lernen verwendet. Motiviert durch die Problematik der gewöhnlich geringen Trainingsmenge wurde außerdem ein Verfahren entwickelt, das auf der Kombination von überwachtem und unüberwachtem Lernen basiert. In diesem Ansatz werden neben den Stichprobenelementen auch unklassifizierte Bilder der Datenbank zur Systemadaption verwendet. In einer abschließenden Evaluation konnte die Lernfähigkeit des INDI Systems eindrucksvoll nachgewiesen werden.

Neben den Aspekten der adaptiven Bildersuche wurde in dieser Arbeit die Notwendigkeit von multidimensionalen Indizierungsverfahren betont. Diese ermöglichen, dass die vorgestellten inhaltsbasierten Techniken auch auf umfangreiche Datenbestände skalierbar sind. Einleitend wurden zunächst verschiedene Techniken vorgestellt, die die Organisation hochdimensionaler Daten ermöglichen. Unter diesen Techniken sind zum einen Verfahren wie z.B. Quadtrees oder R-Trees. Zum anderen sind dies Methoden wie z.B. die Vektorquantisierung oder der Neural-Gas-Algorithmus. Aufbauend auf den theoretischen Ausführungen wurde schließlich der für das INDI System entwickelte Indizierungsmechanismus vorgestellt. Dabei wurde nach einer Aufwandsanalyse des Suchprozesses die technische Umsetzung des Indizierungsverfahrens beschrieben. Die zentralen Komponenten dieses Ansatzes stellen dabei zwei Datenbanktabellen dar, die Informationen beinhalten, um den Suchraum eines Suchschritts einschränken zu können. Die zugehörige Evaluation demonstrierte schließlich, dass trotz der Einschränkung des Suchraumes gute Suchergebnisse in der Evaluationsdatenbank erzielt werden können. Allerdings haben die Untersuchungen auch gezeigt, dass es durchaus problematisch ist, ein adaptives Verfahren auf eine unüberwacht erzeugte Datenbankorganisation zu stützen.

Abschließend betrachtet stellt das INDI System einen hervorragenden Systemansatz zur inhaltsbasierten Bildersuche in einer digitalen Bildsammlung dar. Seine multimodale Schnittstelle ermöglicht einerseits eine einfache und natürliche Datenbanknavigation. Andererseits ist ausgehend von dem entwickelten Suchverfahren die adaptive

inhaltsbasierte Suche in einer Menge von digitalen Bildern möglich, sodass Benutzer und System auf der visuellen Ebene miteinander interagieren.

6.2 Ausblick

Da das Forschungsfeld der inhaltsbasierten Bildersuche sehr umfangreich ist und dementsprechend im INDI System nur einige Aspekte der inhaltsbasierten Bildersuche bearbeitet werden konnten, existieren verschiedene Aspekte, die die Grundlage weiterführender Arbeiten darstellen könnten. Da der Schwerpunkt dieser Arbeit auf dem Systemlernen innerhalb der Bildersuche liegt, sollten Anschlußarbeiten gegebenfalls weitere elementare Bestandteile eines Bildsuchsystems betrachten. Mögliche Themengebiete sind z.B. die Extraktion bildspezifischer Merkmale oder der Entwurf neuer Abstandsmaße. Diesbezüglich existiert mit dem INDI System ein flexibler Systemansatz, der sich hervorragend zum Testen solcher Merkmale und Abstandsmaße eignet.

Auch das Problem der in der Regel geringen Trainingsmenge sollte verstärkt untersucht werden. Dabei bietet es sich einerseits an, ähnlich wie in dieser Arbeit, unklassifizierte Bilder der Datenbank in den Lernprozess zu integrieren. Andererseits könnte durch eine spezielle Interaktion mit dem Benutzer versucht werden, die aktuelle Stichprobe zu erweitern. Vorstellbar wäre dabei, dass einem Anwender vor einem Lernschritt gezielt weitere Bilder zur Bewertung präsentiert werden. Da dieser Vorgang eine einfache Interaktion zwischen Mensch und System erfordert, sollten dabei die im INDI System entwickelten Funktionalitäten zur multimodalen Interaktion genutzt werden. Allerdings muss darauf geachtet werden, dass diese zusätzliche Interaktion keine Belastung für einen Anwender darstellt.

Neben weiteren Verfahren zum Kurzzeitlernen könnten in weiterführenden Arbeiten auch spezielle Methoden des Langzeitlernens entwickelt werden (vgl. z.B. [Min96] oder [Kos03]). Damit kann eine Organisation der Datenbank erreicht werden, die im besten Fall einer semantischen Kategorisierung entspricht, sodass die inhaltsbasierten Techniken auf umfangreiche Datenbestände skalierbar sind. Aufgrund der semantischen Lücke bietet es sich außerdem an, hybride Bildsuchsysteme zu entwerfen, in denen die Vorteile der text- und inhaltsbasierten Bildersuche miteinander kombiniert werden (vgl. z.B. [Zho02]).

A Triviale Lösung der Distanzminimierung

Das in Kapitel 4.3.1 beschriebene Adaptionsschema basiert auf der Minimierung von Distanzwerten. Dabei werden die Parameter $W = \{\mathbf{W}_j\}$ und $V = \{v_j\}$ so adaptiert, dass die Abstände der Beispielobjekte $\{\mathcal{O}_h | h = 1, 2, \ldots, H\}$ zum idealen Anfrageobjekt $\mathcal{Q}^*$ minimiert werden:

$$\min(F), \text{ mit } F = \boldsymbol{\pi}^T \mathbf{f}$$

Bei genauerer Betrachtung lässt sich zeigen, dass diese Gleichung ohne die zusätzliche Formulierung von Nebenbedingungen lediglich die triviale Lösung besitzt.

Es gilt:

$$\begin{aligned} F &= \boldsymbol{\pi}^T \mathbf{f} \\ &= \sum_{h=1}^{H} \pi_h f_h \\ &= \sum_{h=1}^{H} \pi_h \sum_{j=1}^{J} v_j \, d_j(\mathbf{r}_j^h, \mathbf{q}_j^*, \mathbf{W}_j) \\ &= \sum_{h=1}^{H} \pi_h \sum_{j=1}^{J} v_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*)^T \, \mathbf{W}_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*) \end{aligned}$$

Die Minimierung der Funktion F erfordert jeweils die partiellen Ableitungen nach v_j und $w_{j_{mn}}$. Für die Ableitung nach dem j-ten Repräsentantengewicht v_j gilt:

$$\begin{aligned} \frac{\partial F}{\partial v_j} &= \frac{\sum_{h=1}^{H} \pi_h \sum_{j=1}^{J} v_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*)^T \, \mathbf{W}_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*)}{\partial v_j} \\ &= \sum_{h=1}^{H} \pi_h \, (\mathbf{r}_j^h - \mathbf{q}_j^*)^T \, \mathbf{W}_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*) \end{aligned}$$

Der Ausdruck wird genau dann für alle Elemente der Stichprobe minimal, wenn die Gewichtsmatrix $\mathbf{W}_j$ ausschließlich Nullen beinhaltet, $w_{j_{mn}} = 0$ für $m, n = 1, 2, \ldots, N_j$.

Entsprechend gilt für die partielle Ableitung nach den Komponenten $w_{j_{mn}}$ der Gewichtsmatrix $\mathbf{W}_j$:

$$\begin{aligned}
\frac{\partial F}{\partial w_{j_{mn}}} &= \frac{\sum_{h=1}^{H} \pi_h \sum_{j=1}^{J} v_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*)^T \, \mathbf{W}_j \, (\mathbf{r}_j^h - \mathbf{q}_j^*)}{\partial w_{j_{mn}}} \\
&= \frac{\sum_{h=1}^{H} \pi_h \sum_{j=1}^{J} v_j \; \sum_{m=1}^{N_j} \sum_{n=1}^{N_j} w_{j_{mn}} (r_{j_m}^h - q_{j_m}^*)(r_{j_n}^h - q_{j_n}^*)}{\partial w_{j_{mn}}} \\
&= \sum_{h=1}^{H} \pi_h \, v_j \, (r_{j_m}^h - q_{j_m}^*) \, (r_{j_n}^h - q_{j_n}^*)
\end{aligned}$$

Dieser Ausdruck besitzt für alle Elemente der Stichprobe genau dann ein Minimum, wenn $v_j = 0$ für $j = 1, 2, \ldots, J$. Damit ist gezeigt, dass der Ansatz der Distanzminimierung ohne die Formulierung von Nebenbedingungen für die Parameter $W = \{\mathbf{W}_j\}$ und $V = \{v_j\}$ lediglich die triviale Lösung besitzt.

B Farbräume

Farbe ist eine Sehempfindung, die jedem Objekt gegeben ist. Sie ist nach DIN-Norm 5033 als der perzeptive Eindruck definiert, der uns die Unterscheidung zweier strukturloser Flächen gleicher Helligkeit ermöglicht. Die Grundlage der Farbwahrnehmung sind „sichtbare“ Lichtstrahlen. Dabei handelt es sich um elektromagnetische Strahlung, deren Wellenlängen sich auf einen Bereich von ca. 400 bis 700 Nanometer erstrecken. Wenn diese als einfallendes Licht auf das menschliche Auge trifft, wird sie von verschiedenen Rezeptoren absorbiert. Die Photorezeptoren wiederum können in zwei Arten klassifiziert werden: Stäbchen und Zapfen. Während die Stäbchen nur beim Dämmerungs- und Nachtsehen aktiv sind, dienen die Zapfen zum Farbensehen. Die Zapfenrezeptoren werden entsprechend ihrer Empfindlichkeit in drei Arten gruppiert, die als lang-, mittel- und kurzwellenlänge-sensitiv bezeichnet werden.

Neben der physikalischen Beschreibung anhand der Wellenlänge existieren weitere Varianten, die die Farbbeschreibung ermöglichen. Ein System, das die mathematische Repräsentation von Farbe ermöglicht wird als Farbraum bezeichnet. Mit jedem Farbraum ist ein Farbmodell assoziiert, das die Parameter des Farbraumes definiert. Diese spannen ein Koordinatensystem auf, in dem Farben als Punkte repräsentiert werden. Die in der Literatur verfügbaren Farbsysteme sind zahlreich (vgl. z.B. [Wys82], [Gev01] oder [Gon02]). Jedes von ihnen besitzt Eigenschaften, die aus der Berücksichtigung spezieller Aspekte motivierbar sind und sich daher für bestimmte Anwendungen als besonders praktisch darstellen. Da Farbe ein wichtiges Charakteristikum der inhaltsbasierten Bildersuche ist, haben sich die verschiedenen Farbräume auch in diesem Anwendungsfeld etabliert. Sie dienen als Grundlage der Merkmalsextraktion und ermöglichen eine vielfältige Repräsentation der verschiedenen Farben eines Bildes. In den folgenden Abschnitten werden die für diese Arbeit relevanten Farbräume näher erläutert.

B.1 RGB Farbraum

Der RGB Farbraum ist ein additives Farbsystem, in dem Farben als Überlagerung der Grundfarben **R**ot, **G**rün, und **B**lau dargestellt werden. Das Prinzip wird beispielsweise bei Farbbildschirmen zur Darstellung eines Bildes eingesetzt. Die drei Grundfarben spannen ein kartesisches Koordinatensystem auf, das wie in Abbildung B.1 dargestellt als Würfel repräsentiert werden kann. Jeder Farbkanal ist dabei auf ein fixes Intervall,

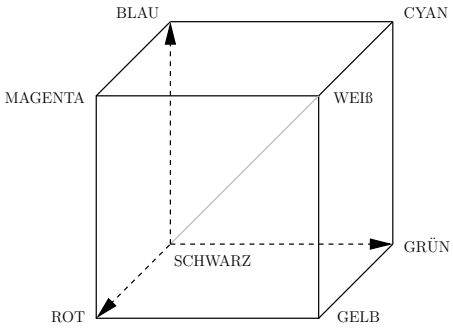


Abb. B.1: RGB Farbwürfel

z.B. $[0, 255]$, begrenzt. Grautöne (R=G=B) befinden sich in dem Koordinatensystem auf der eingezeichneten Würfeldiagonalen.

Die Nachteile der RGB Farbrepräsentation sind allerdings, dass die Farben weder intuitiv noch perzeptuell linear sind. Unter intuitiv wird eine Farbdarstellung verstanden, die der des Menschen entspricht. Da dieser Farben eher in der Form von Farbton, Sättigung und Intensität handhabt, ist diese Bedingung im RGB Raum nicht erfüllt. Auch die perzeptuelle Linearität ist nicht gegeben. Diese erfordert nämlich, dass kleine Farbänderungen im Farbraum von einem menschlichen Betrachter auch als kleine Farbänderungen wahrgenommen werden.

B.2 HSI Farbraum

Die Farbdarstellung im HSI Farbraum ist im Gegensatz zur RGB Repräsentation intuitiv. In ihm werden Farben durch Farbton (engl. **H***ue*), Sättigung (engl. **S***aturation*) und Intensität (engl. **I***ntensity*) beschrieben. Der in Abbildung B.2 dargestellte Doppelkegel ermöglicht eine schematische Veranschaulichung des Farbsystems.

Die Mittelachse des Doppelkegels repräsentiert die Intensität I. Ihre Werte stammen aus dem Intervall $[0, 1]$ und nehmen von unten nach oben hin zu. Dies entspricht einem Intensitätsverlauf von Schwarz nach Weiß. Farbtöne H werden als Winkel um die Intensitätsachse repräsentiert. Komplementärfarben sind dabei um $180°$ versetzt. Bildet man einen Schnitt durch den Kegel, so lassen sich die verschiedenen Farben als Sektoren in den resultierenden Kreis einzeichnen. Die Sättigungsachse steht senkrecht auf der Intensitätsachse und verläuft von innen nach außen. Der Sättigungswert S nimmt mit dem Abstand zur Mittelachse zu. Dies ist gleichbedeutend mit einer Verringerung des Weißanteils in dem entsprechende Farbton. Im Zentrum besitzt die Sättigung ihr Minimum $S_{\text{min}} = 0$ und auf der Kegeloberfläche ihr Maximum $S_{\text{max}} = 1$.

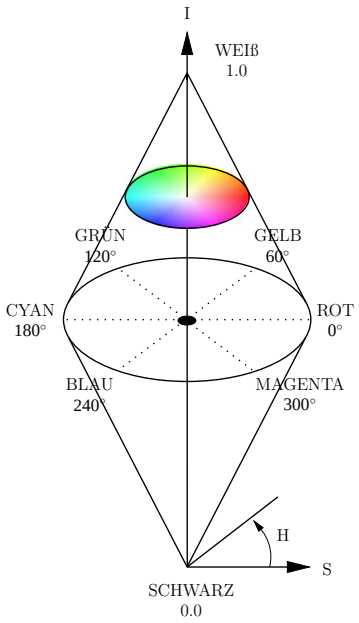


Abb. B.2: HSI Farbmodell

Ausgehend vom RGB Farbraum sind die Werte des HSI Farbraumes nach Gonzales und Woods [Gon02, S.299] wie folgt definiert:

$$
\begin{aligned}
I &= \frac{R+G+B}{3} \\
S &= 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \\
H &= \begin{cases} \theta & , \text{ falls } B \leq G \\ 360-\theta, & \text{sonst} \end{cases} ,
\end{aligned}
$$

mit

$$
\theta = \arccos\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}} \right\}
$$

Bei der Verwendung des HSI Farbraumes als Grundlage der Merkmalsextraktion ist zu beachten, dass die Repräsentation des Farbtons zirkulär ist. Dies erfordert daher entweder eine spezielle Handhabung während der Merkmalsberechnung oder den Einsatz spezieller Abstandsmaße.

B.3 CIE $L^*u^*v^*$ Farbraum

Ein weiterer häufig zur Berechnung von Bildcharakteristika verwendeter Farbraum ist der CIE[1] $L^*u^*v^*$ Raum (vgl. z.B. [Mog99] oder [Fau02]). Er unterscheidet sich von den bisher betrachteten Modellen dadurch, dass er nahezu perzeptuell linear ist und sich daher besonders gut für die Anwendung der inhaltsbasierten Bildersuche eignet.

Der $L^*u^*v^*$ Farbraum ist auf der Grundlage des CIE XYZ Farbsystems definiert. In diesem werden Farben als additive Mischung der drei Primitiva X, Y und Z dargestellt. Sie ermöglichen die Darstellung aller reellen Farben, ohne ein negatives Vorzeichen benutzen zu müssen. Ihre Berechnung erfolgt auf der Grundlage der RGB Darstellung (mit D_{65} als Referenzweiß) nach folgender Berechnungsvorschrift [Poy97]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ausgehend von diesen Primärvalenzen sind die Komponenten des $L^*u^*v^*$ Farbraumes durch folgende Gleichungen definiert [Rub01, S. 30ff]:

$$\begin{aligned} L^* &= \begin{cases} 116\,(Y/Y_n)^{\frac{1}{3}} - 16, & \text{falls } (Y/Y_n) > 0.008856 \\ 903.3\,(Y/Y_n) \quad\;\; , & \text{sonst} \end{cases} \\ u^* &= 13\,L^*(u' - u'_n), \\ v^* &= 13\,L^*(v' - v'_n, \end{aligned}$$

mit

$$u' = \frac{4X}{X + 15Y + 3Z}, v' = \frac{9Y}{X + 15Y + 3Z}$$

u'_n und v'_n sind ebenso wie u' und v' definiert. Ihre Berechnung basiert jedoch auf den Normfarbwerten X_n, Y_n und Z_n, die durch die verwendete Standardbeleuchtung gegeben sind. Beispiele dafür sind die CIE Beleuchtungsstandards A, B, C und D_{65} (vgl. [Wys82, S.142ff]).

B.4 CIE $L^*a^*b^*$ Farbraum

Ein weiterer ebenfalls nahezu perzeptuell linearer Farbraum ist der $L^*a^*b^*$ Farbraum. Dieser stimmt in der Helligkeits- bzw. Luminanzkomponente L^* mit dem $L^*u^*v^*$ Farbraum überein. Im Gegensatz zu diesem berechnen sich die beiden Farbkomponenten

[1] Commission Internationale de l'Eclairage: Institution zur Standardisierung im Bereich der Kolorimetrie und Photometrie, die 1913 die Aufgaben der *Commision Internationale de Photométrie* übernommen hat.

jedoch direkt aus denen des XYZ Farbraumes. Während a^* mit dem Rot-Grün-Anteil der Farbe korreliert, repräsentiert b^* den Gelb-Blau-Anteil. Beide Komponenten werden wie folgt berechnet [Rub01, S. 30ff]:

$$\begin{aligned} a^* &= 500\,(f(X/X_n) - f(Y/Y_n)), \\ b^* &= 200\,(f(Y/Y_n) - f(Z/Z_n)), \end{aligned}$$

mit

$$f(t) = \begin{cases} t^{1/3} & , \quad \text{falls } Y/Y_n > 0.008856 \\ 7.787\,t + 16/116, & \quad \text{sonst} \end{cases}$$

C Dimensionsreduktion durch Hauptachsentransformation

Bei der Hauptachsentransformation[1] handelt es sich um ein Verfahren, das eine gegebene Datenmenge $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$, mit $\mathbf{x}_i \in \mathbb{R}^N$, in einen neuen Merkmalsraum abbildet, in dem die Komponenten der transformierten Datenvektoren dekorreliert sind. Dabei sind die Achsen des neuen Koordinatensystems gewöhnlich so angeordnet, dass die Streuung der Daten in der ersten Komponente am größten ist und bis zur letzten Komponente abnimmt (vgl. z.B. [Nie83], [Jol86] oder [Fin03]).

Die Grundlage der Hauptachsentransformation bildet die Analyse der Streuungseigenschaften einer Datenmenge X. Diese lassen sich durch die Gesamtstreuungsmatrix (engl. *Total Scatter Matrix*) $\mathbf{S}_T \in \mathbb{R}^N \times \mathbb{R}^N$ wie folgt beschreiben:

$$\mathbf{S}_T = \frac{1}{I} \sum_{i=1}^{I} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

Dabei symbolisiert $\boldsymbol{\mu}$ den Mittelwertsvektor der betrachteten Datenmenge:

$$\boldsymbol{\mu} = \frac{1}{I} \sum_{i=1}^{I} \mathbf{x}_i$$

Für die Streuungseigenschaften der Datenmenge X gilt nach Anwendung einer Transformation $\mathbf{T}$:

$$\tilde{\mathbf{S}}_T = \frac{1}{I} \sum_{i=1}^{I} \mathbf{y}_i \mathbf{y}_i^T = \frac{1}{I} \sum_{i=1}^{I} \mathbf{T}(\mathbf{x}_i - \boldsymbol{\mu})[\mathbf{T}(\mathbf{x}_i - \boldsymbol{\mu})]^T = \mathbf{T}\,\mathbf{S}_T\mathbf{T}^T$$

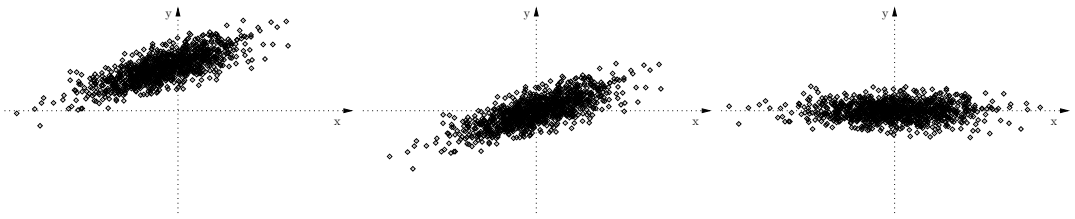
Eine Dekorrelation der Daten wird somit dann erreicht, wenn eine Transformation $\mathbf{T}$ gefunden wird, die $\mathbf{S}_T$ diagonalisiert, sodass $\tilde{\mathbf{S}}_T$ Diagonalform besitzt (vgl. z.B. [Fin03, S.142]).

Unter der Voraussetzung, dass die symmetrische Matrix $\mathbf{S}_T$ nicht singulär[2] ist, lässt sie sich durch eine orthonormale Transformation $\mathbf{T} = \boldsymbol{\Phi}^T$ diagonalisieren. $\boldsymbol{\Phi}$ repräsentiert dabei die Eigenvektormatrix, die wie folgt strukturiert ist:

$$\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N]$$

[1] Das Verfahren wird auch als Hauptkomponentenanalyse (engl. *Principle Component Analysis*, PCA) oder Karhunen-Loève-Transformation (KLT) bezeichnet. Die KLT basiert jedoch nicht auf der Zentrierung der zu transformierenden Datenmenge und ist somit nur bei mittelwertfreien Daten zur Hauptachsentransformation äquivalent.

[2] Singuläre Matrizen erfordern in der Praxis eine spezielle Handhabung, wie z.B. die von Friedman [Fri89] vorgestellte Regularisierung.



(a) Originaldaten, korreliert (b) Zentr. Daten, korreliert (c) Transf. Daten, dekorreliert

Abb. C.1: Beispiel für die Hauptachsentransformation einer zweidimensionalen Verteilung: (a) Originaldaten, (b) zentrierte Daten und (c) Zentrierte Datenmenge nach Anwendung der Hauptachsentransformation mit dekorrelierten Merkmalskomponenten und der größten Varianz entlang der ersten Koordinatenachse.

An dieser Stelle wird angenommen, dass die Eigenvektoren ϕ_n entsprechend ihrer Eigenwerte $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ sortiert sind. Obwohl diese Ordnung für den Vorgang der Diagonalisierung nicht zwingend ist, werden die Eigenvektoren unter Berücksichtigung der angestrebten Dimensionsreduktion entsprechend der Größe der korrespondierenden Eigenwerte angeordnet. Für die Eigenvektoren der Gesamtstreuungsmatrix $\mathbf{S}_T$ gilt:

$$\mathbf{S}_T \phi_n = \phi_n \lambda_n, \text{ für } n = 1, \ldots, N \tag{C.1}$$

Aus der Verallgemeinerung dieser Eigenwertgleichung folgt:

$$\mathbf{S}_T \mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda} \Leftrightarrow \mathbf{S}_T = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T, \text{ mit } \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_N \end{bmatrix}$$

Wendet man die Transformation $\mathbf{T} = \mathbf{\Phi}^T$ auf mittelwertfreie Daten an, so gilt für die Streuungseigenschaften der Bildpunkte $\mathbf{y}_i = \mathbf{\Phi}^T(\mathbf{x}_i - \boldsymbol{\mu})$:

$$\tilde{\mathbf{S}}_T = \mathbf{\Phi}^T \mathbf{S}_T \mathbf{\Phi} = \mathbf{\Phi}^T \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{\Lambda}$$

Bei der Hauptachsentransformation handelt es sich somit um eine Abbildung, die eine gegebene Datenmenge X in den durch die Eigenvektoren aufgespannten Eigenraum transformiert. Da diese Transformation außerdem orthonormal ist, bleibt die relative Lage der Datenvektoren zueinander erhalten [Fin03, S. 141]. Abbildung C.1 veranschaulicht die Hauptachsentransformation einer zweidimensionalen Verteilung.

Die Varianz der Daten, und somit der Informationsgehalt, nimmt mit den höheren Komponenten ab. Eine Nichtberücksichtigung der Eigenvektoren, deren Eigenwerte klein sind, würde daher den gesamten Informationsgehalt nur wenig verringern und zusätzlich zu einer Reduktion der Vektordimension führen. Zur Dimensionsreduktion

werden daher anstelle aller Eigenvektoren nur die Eigenvektoren der k größten Eigenwerte betrachtet. Der durch die k Eigenvektoren aufgespannte Eigenraum ist niedrigdimensionaler als der ursprüngliche Merkmalsraum und die Abbildung eines Datenvektors $\mathbf{x}_i$ in diesen Eigenraum ist wie folgt definiert[3]:

$$\tilde{\mathbf{x}}_i = \mathbf{\Phi}_k^T \mathbf{x}_i, \text{ mit } \mathbf{\Phi}_k = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_k] \text{ und } \tilde{\mathbf{x}}_i \in \mathbb{R}^k$$

Natürlich resultiert aus dem Verzicht auf bestimmte Eigenvektoren ein gewisser Fehler, der durch genauere Betrachtung der Rücktransformation auch qualitativ erfasst werden kann:

$$\mathbf{x}_i' = \sum_{n=1}^{k} \tilde{x}_{i_n} \boldsymbol{\phi}_n$$

mit Approximationsfehler

$$\Delta \mathbf{x}_i = \sum_{n=k+1}^{N} \tilde{x}_{i_n} \boldsymbol{\phi}_n, \tag{C.2}$$

wobei für eine Vektorkomponente im Eigenraum $\text{span}\{\boldsymbol{\phi}_n | n = 1, 2, \ldots, N\}$ gilt:

$$\tilde{x}_{i_n} = \boldsymbol{\phi}_n^T \mathbf{x}_i \tag{C.3}$$

Es kann gezeigt werden, dass der mittlere quadratische Approximationsfehler, der durch die Rekonstruktion einer Datenmenge X entsteht, der Summe der unberücksichtigten Eigenwerte entspricht (vgl. Herleitung in Abbildung C.2):

$$E = \frac{1}{I} \sum_{i=1}^{I} (\Delta \mathbf{x}_i)^2 = \sum_{n=k+1}^{N} \lambda_n$$

Der prozentuale Anteil des Gesamtinformationsgehalts, der durch die Projektion einer Datenmenge in den niedrigdimensionaleren Eigenraum erhalten bleibt, lässt sich qualitativ wie folgt beschreiben (vgl. z.B. [Jol86, S. 93f] oder [Jai88, S. 27]):

$$\rho = 100 \, \frac{\sum_{n=1}^{k} \lambda_n}{\sum_{m=1}^{N} \lambda_m}$$

Somit kann die Anzahl der Eigenvektoren abgeschätzt werden, die erforderlich sind, um einen bestimmten Informationsgehalt der ursprünglichen Datenmenge zu bewahren.

[3] Für die folgenden Ausführungen gehen wir oBdA von mittelwertbereinigten Daten aus, sodass für die Datenmenge $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$ gilt: $\boldsymbol{\mu} = \frac{1}{I} \sum_{i=1}^{I} \mathbf{x}_i = \mathbf{0}$. Demzufolge entspricht die Gesamtstreuungsmatrix der Autokorrelationsmatrix der Datenmenge X.

Ausgehend vom Eigenraum span$\{\phi_n | n = 1, 2, \ldots, k\}$ mit $k < N$, wobei N die Dimension der Merkmalsvektoren $\mathbf{x}_i$ der Datenmenge $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I\}$ repräsentiert, gilt für den mittleren quadratischen Approximationsfehler:

$$\begin{aligned} E &= \frac{1}{N}\sum_{i=1}^{I}(\Delta\mathbf{x}_i)^2 \\ &\overset{Gl.\ C.2}{=} \frac{1}{N}\sum_{i=1}^{I}\sum_{m,n>k}\tilde{x}_{i_m}\phi_m\tilde{x}_{i_n}\phi_n \\ &\overset{\text{NR}}{=} \sum_{m,n>k}\delta_{mn}\lambda_n\phi_m\phi_n \\ &= \sum_{n=k+1}^{N}\lambda_n \end{aligned}$$

Nebenrechnung (NR):

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{I}\tilde{x}_{i_m}\tilde{x}_{i_n} &\overset{Gl.\ C.3}{=} \frac{1}{N}\sum_{i=1}^{I}\phi_m^T\mathbf{x}_i\phi_n^T\mathbf{x}_i \\ &= \frac{1}{N}\sum_{i=1}^{I}\phi_m^T(\mathbf{x}_i\mathbf{x}_i^T)\phi_n \\ &= \phi_m^T\,\mathbf{S}_T\,\phi_n \\ &\overset{Gl.\ C.1}{=} \phi_m^T\,\phi_n\,\lambda_n \\ &= \delta_{mn}\,\lambda_n \end{aligned}$$

Da von mittelwertbereinigten Daten ausgegangen wird, entspricht die Autokorrelationsmatrix $\mathbf{C} = \frac{1}{I}\sum_{i=1}^{I}\mathbf{x}_i\mathbf{x}_i^T$ der Gesamtstreuungsmatrix $\mathbf{S}_T$.

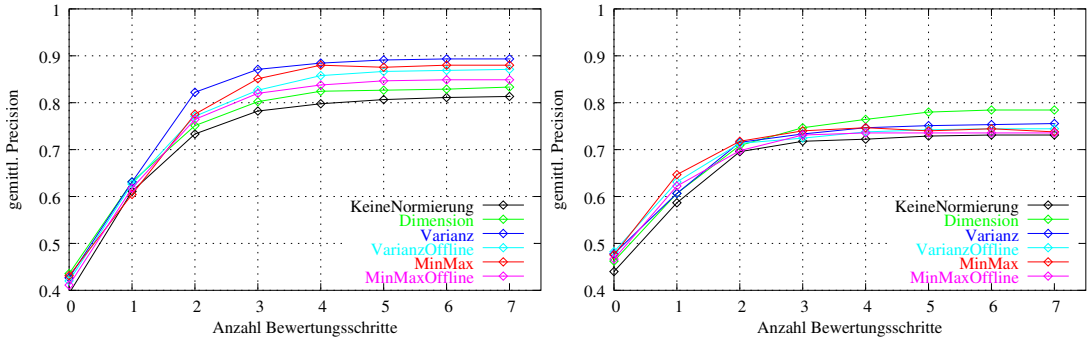
Abb. C.2: Herleitung des mittleren quadratischen Approximationsfehlers

D Ergebnisse der experimentellen Untersuchungen

Dieser Anhang enthält weitere Ergebnisse der in Kapitel 4 durchgeführten Evaluation des in dieser Arbeit vorgestellten adaptiven Suchprozesses. Dabei wurden die verschiedenen Experimente anstatt mit einer Ergebnismenge von 30 Bildern mit einer Ergebnismenge von 45 Bildern durchgeführt. Die Details der hier aufgelisteten Experimente sind den Ausführungen in Abschnitt 4.4.4 zu entnehmen:

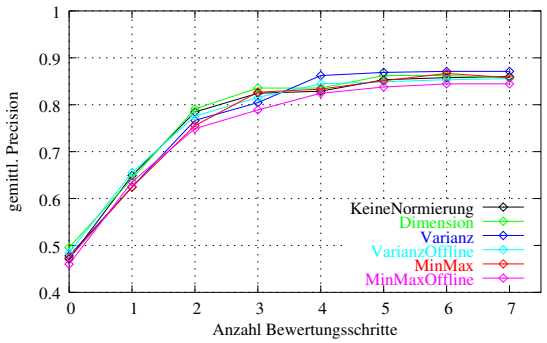
Experiment A:	Vergleich von Verfahren zur Distanznormierung
Experiment B:	Vergleich von Bildersuchen mit gewichtetem und generalisiertem euklidischen Abstand sowie mit und ohne negativen Beispielen
Experiment C:	Vergleich der Verfahren zur Adaption der Repräsentantengewichte
Experiment D:	Analyse von Regularisierung und Co-Training

Experiment A: Vergleich der Verfahren zur Distanznormierung

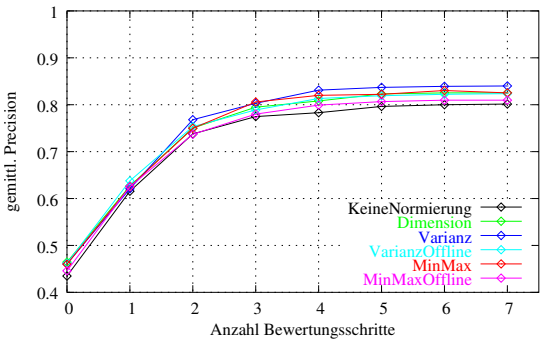


(a) Autorennen

(b) Blumen



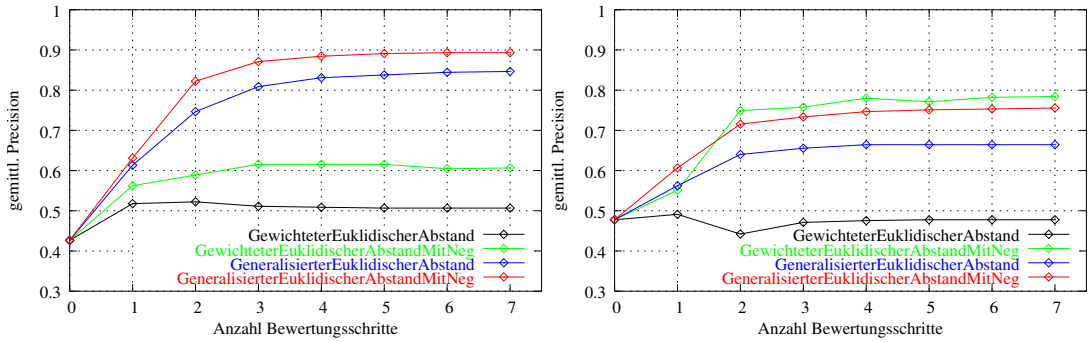
(c) Golf



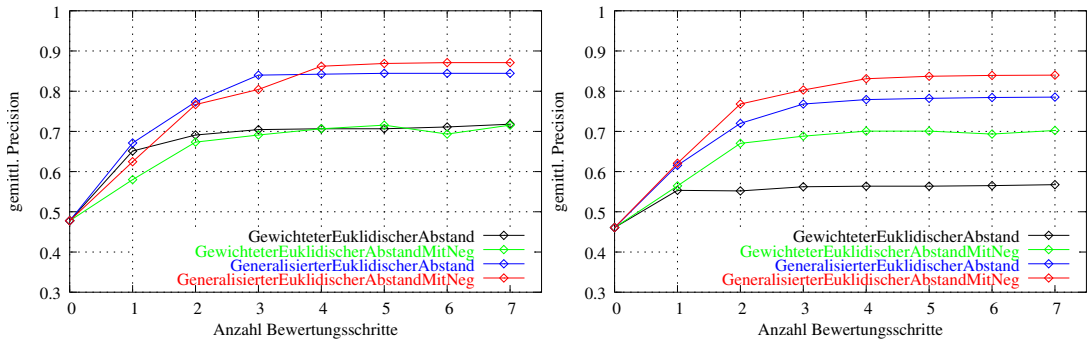
(d) Gesamt

Abb. D.1: Vergleich der unterschiedlichen Verfahren zur Normierung der Distanzwerte bei einer Ergebnismenge von 45 Bildern.

Experiment B:Vergleich von Bildersuchen mit gewichtetem und generalisiertem euklidischen Abstand sowie mit und ohne negativen Beispielen (erster Teil)



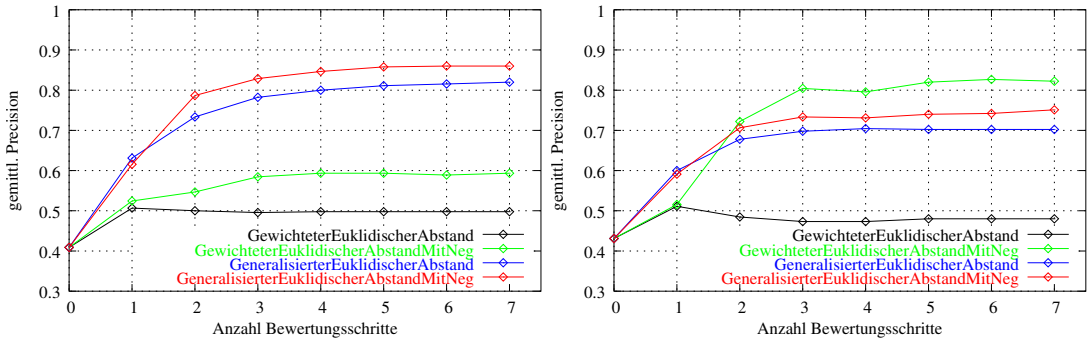
(a) Autorennen (b) Blumen



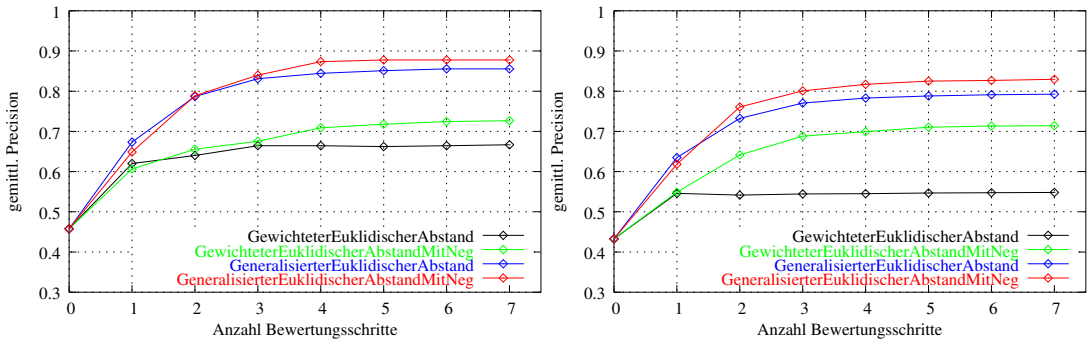
(c) Golf (d) Gesamt

Abb. D.2: Distanzbasierte Bildersuche mit gewichtetem und generalisiertem euklidischen Abstand mit und ohne negative Trainingsbeispiele bei einer Ergebnismenge von 45 Bildern.

Experiment B:Vergleich von Bildersuchen mit gewichtetem und generalisiertem euklidischen Abstand sowie mit und ohne negativen Beispielen (zweiter Teil)



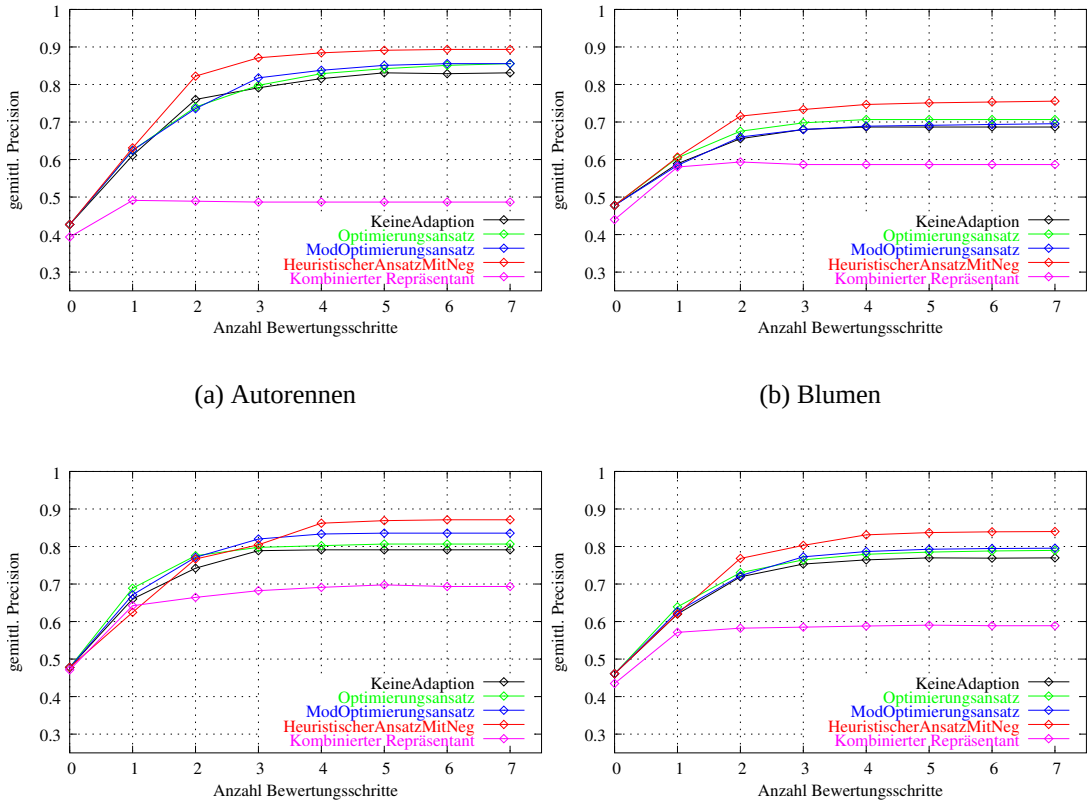
(a) Autorennen (b) Blumen



(c) Golf (d) Gesamt

Abb. D.3: Rangbasierte Bildersuche mit gewichtetem und generalisiertem euklidischen Abstand mit und ohne negative Trainingsbeispiele bei einer Ergebnismenge von 45 Bildern.

Experiment C: Vergleich der Verfahren zur Adaption der Repräsentantengewichte (erster Teil)



(a) Autorennen

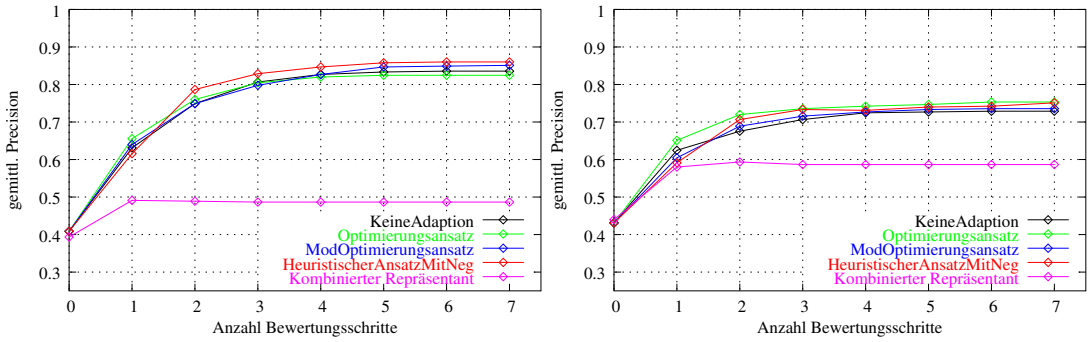
(b) Blumen

(c) Golf

(d) Gesamt

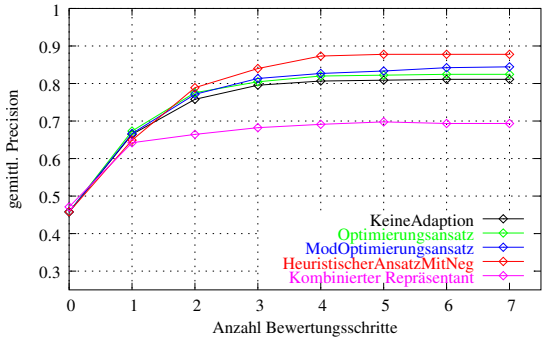
Abb. D.4: Vergleich der Verfahren zur Adaption der Repräsentantengewichte bei einer Ergebnismenge von 45 Bildern (distanzbasiert).

Experiment C: Vergleich der Verfahren zur Adaption der Repräsentantengewichte (zweiter Teil)

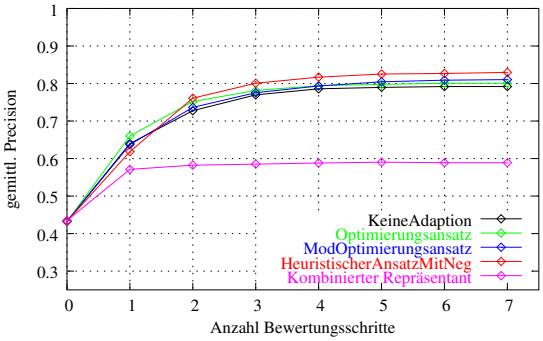


(a) Autorennen

(b) Blumen



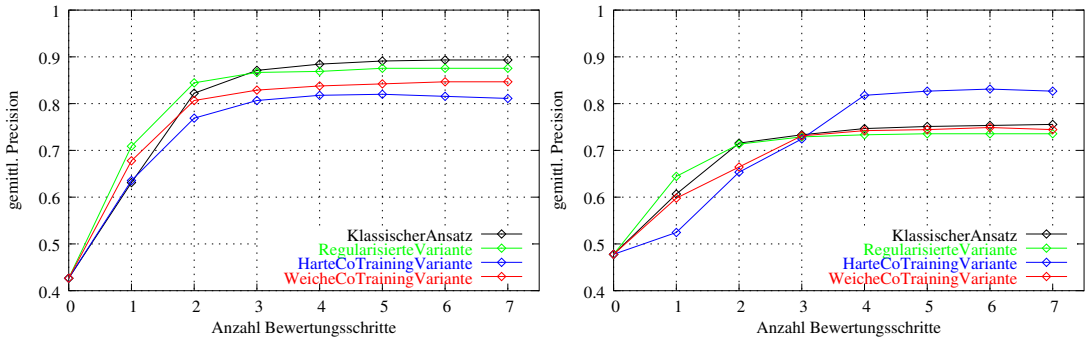
(c) Golf



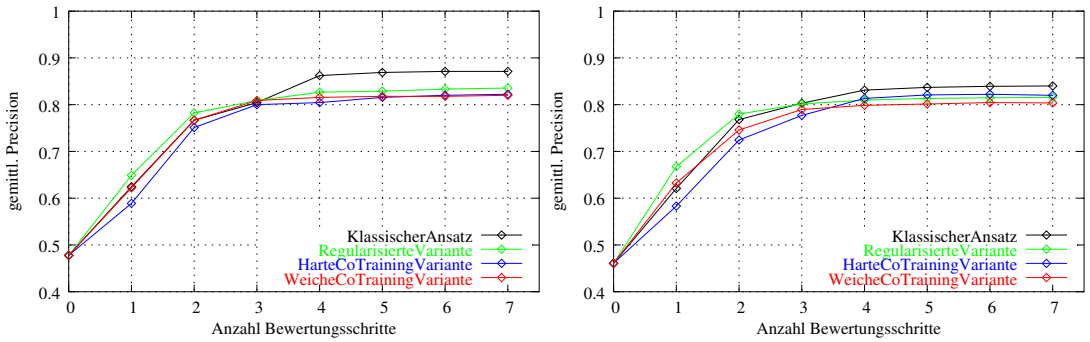
(d) Gesamt

Abb. D.5: Vergleich der Verfahren zur Adaption der Repräsentantengewichte bei einer Ergebnismenge von 45 Bildern (rangbasiert).

Experiment D: Analyse von Regularisierung und Co-Training (erster Teil)



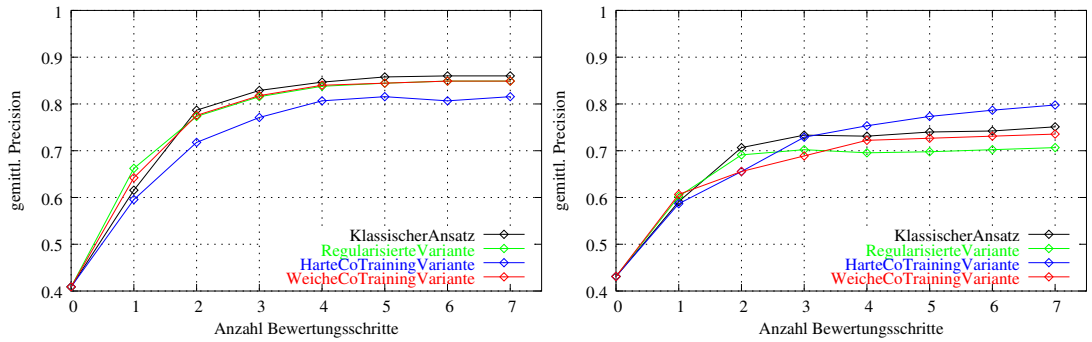
(a) Autorennen (b) Blumen



(c) Golf (d) Gesamt

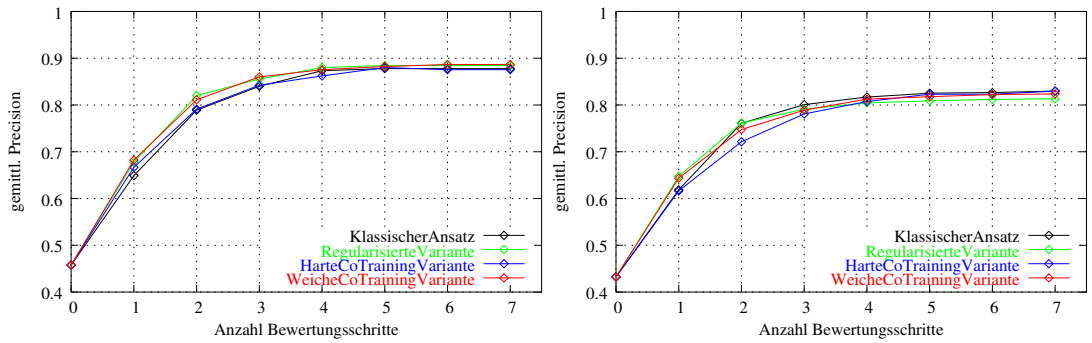
Abb. D.6: Co-Training und Regularisierung bei einer Ergebnismenge von 45 Bildern (distanzbasiert).

Experiment D: Analyse von Regularisierung und Co-Training (zweiter Teil)



(a) Autorennen

(b) Blumen



(c) Golf

(d) Gesamt

Abb. D.7: Co-Training und Regularisierung bei einer Ergebnismenge von 45 Bildern (rangbasiert).

Literatur

[Aks01] S. Aksoy, R. Haralick: *Feature Normalization and Likelihood-Based Similarity Measures for Image Retrieval*, *Pattern Recognition Letters*, Bd. 22, Nr. 5, April 2001, S. 563–582.

[AM98] M. Abdel-Mottaleb, S. Krishnamachari, N. Mankovich: *Performance Evaluation of Clustering Algorithms for Scalable Image Retrieval*, in K. Bowyer, P. Phillips (Hrsg.): *Empirical Evaluation Techniques in Computer Vision*, IEEE Computer Society Press, Juni 1998.

[Bac96] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, C. Shu: *The Virage Image Search Engine: An Open Framework for Image Management*, in *Proc. of Storage and Retrieval for Image and Video Databases IV*, Bd. 2670 von *SPIE*, San Diego/La Jolla, CA, USA, Jan./Feb. 1996, S. 76–87.

[Bau02] C. Bauckhage, J. Fritsch, K. Rohlfing, S. Wachsmuth, G. Sagerer: *Evaluating Integrated Speech- and Image Understanding*, in *Proc. of IEEE International Conference on Multimodal Interfaces*, Pittsburgh, PA, Okt. 2002, S. 9–14.

[Bau03] C. Bauckhage, T. Käster, M. Pfeiffer, G. Sagerer: *Content-Based Image Retrieval by Multimodal Interaction*, in *Proc. of 29th Annual Conference of the IEEE Industrial Electronics Society*, Roanoke, VA, Nov. 2003, S. 1865–1870.

[Bec90] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger: *The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles*, in *Proc. of ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, Mai 1990, S. 322–331.

[Ben02] A. Ben-Hur, D. Horn, H. Siegelmann, V. Vapnik: *Support Vector Clustering*, *The Journal of Machine Learning Research*, Bd. 2, März 2002, S. 125–137.

[Blu98] A. Blum, T. Mitchell: *Combining Labeled and Unlabeled Data with Co-Training*, in *Proc. of the Eleventh Annual Conference on Computational Learning Theory*, Madison, Wisconsin, USA, Juli 1998, S. 92–100.

[Bob01] M. Bober: *MPEG-7 Visual Shape Descriptors*, *IEEE Transactions on Circuits and Systems for Video Technology*, Bd. 11, Nr. 6, Juni 2001, S. 716–719.

[Bou00] N. Boujemaa: *Generalized Competitive Clustering for Image Segmentation*, in *Proc. of 19th International Meeting of the North American Fuzzy Information Processing Society*, Atlanta, USA, Juli 2000, S. 133–137.

[Bra99] S. Brandt: *Use of Shape Features in Content-Based Image Retrieval*, Diplomarbeit, Helsinki Universität der Technologie, Finnland, Aug. 1999.

[Bra00] S. Brandt, J. Laaksonen, E. Oja: *Statistical Shape Features in Content-Based Image Retrieval*, in *Proc. of IEEE Internatinal Conference on Pattern Recognition*, Bd. 2, Barcelona, Spain, Sep. 2000, S. 1062–1065.

[Bro66] P. Brodatz: *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.

[Bro98] T. Brondsted, L. Larsen, M. Manthey, P. McKevitt, T. Moeslund, K. Olesen: *The Intellimedia Workbench – A Generic Environment for Multimodal Systems*, in *Proc. of International Conference on Spoken Language Processing*, Sydney, Australia, Nov. 1998, S. 273–276.

[Car99] C. Carson, M. Thomas, S. Belongie, J. Hellerstein, J. Malik: *Blobworld: A System for Region-Based Image Indexing and Retrieval*, in *Proc. of Third International Conference on Visual Information and Information Systems*, Nr. 1614 in Lecture Notes In Computer Science, Springer-Verlag, Amsterdam, Juni 1999, S. 509–516.

[Car02] C. Carson, S. Belongie, H. Greenspan, J. Malik: *Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 24, Nr. 8, Aug. 2002, S. 1026–1038.

[Cas01] V. Castelli: *Multidimensional Indexing Structures for Content-based Retrieval*, RC 22208 (98723), IBM Research Division, Feb. 2001.

[Cel99] A. Celentano, S. Sabbadin: *Multiple Feature Indexing in Image Retrieval Systems*, in *Proc. of First European Workshop on Content-Based Multimedia Indexing*, Toulouse, France, Okt. 1999.

[Cha80] N.-S. Chang, K.-S. Fu: *Query-by-Pictorial-Example*, *IEEE Transactions on Software Engineering*, Bd. 6, Nr. 6, Nov. 1980, S. 519–524.

[Cha88] S. Chang, C. Yan, D. Dimitroff, T. Arndt: *An Intelligent Image Database System*, *IEEE Transactions on Software Engineering*, Bd. 14, Nr. 5, Mai 1988, S. 681–688.

[Che97] J.-Y. Chen, C. Bouman, J. Allebach: *Fast Image Database Search Using Tree-Structured VQ*, in *Proc. of IEEE International Conference on Image Processing*, Bd. 2, Santa Barbara, CA, Okt. 1997, S. 827–830.

[Che00] J.-Y. Chen, C. Bouman, J. Dalton: *Hierarchical Browsing and Search of Large Image Databases*, *IEEE Transactions on Image Processing*, Bd. 9, Nr. 3, März 2000, S. 442–455.

[Che01] Y. Chen, X. Zhou, T. Huang: *One-Class SVM for Learning in Image Retrieval*, in *Proc. of IEEE International Conference on Image Processing*, Bd. 1, Thessaloniki, Greece, Okt. 2001, S. 34–37.

[Com97] D. Comaniciu, P. Meer: *Robust Analysis of Feature Spaces: Color Image Segmentation*, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, Juni 1997, S. 750–755.

[Cox98] I. Cox, M. Miller, T. Minka, P. Yianilos: *An Optimized Interaction Strategy for Bayesian Relevance Feedback*, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, Juni 1998, S. 553–558.

[Cox00] I. Cox, M. Miller, T. Minka, T. Papathomas, P. Yianilos: *The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments*, *IEEE Transactions on Image Processing*, Bd. 9, Nr. 1, Jan. 2000, S. 20–37.

[Dav79] D. Davies, D. Bouldin: *A Cluster Separation Measure*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 1, Nr. 2, April 1979, S. 224–227.

[Dud73] R. Duda, P. Hart: *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[Elm02] R. Elmasri, S. Navathe: *Grundlagen von Datenbanksystemen*, Pearson Studium, München, 3. Ausg., 2002.

[Fal94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz: *Efficient and Effective Querying by Image Content*, *Journal of Intelligent Information Systems*, Bd. 3, Nr. 3/4, Juli 1994, S. 231–262.

[Fau02] J. Fauqueur, N. Boujemaa: *Image Retrieval by Regions: Coarse Segmentation and Fine Color Description*, in *Proc. of IEEE International Conference on Image Processing*, Bd. II, Rochester, NY, Sep. 2002, S. 609–612.

[Fin74] R. Finkel, J. Bentley: *Quad-Trees: A Data Structure for Retrieval on Composite Keys*, *ACTA Informatica*, Bd. 4, Nr. 1, Nov. 1974, S. 1–9.

[Fin99] G. Fink: *Developing HMM-based Recognizers with ESMERALDA*, in V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (Hrsg.): *Lecture Notes in Artificial Intelligence*, Bd. 1692, Springer-Verlag, 1999, S. 229–234.

[Fin03] G. Fink: *Mustererkennung mit Markov-Modellen*, Leitfäden der Informatik, B.G. Teubner, Stuttgart – Leipzig – Wiesbaden, 2003.

[Fli95] M. Flickner, H. Sawhney, W. Niblack: *Query by Image and Video Content: The QBIC System*, *IEEE Computer*, Bd. 28, Nr. 9, Sep. 1995, S. 23–32.

[Fre99] Y. Freund, R. Schapire: *A Short Introduction to Boosting*, *Journal of Japanese Society for Artificial Intelligence*, Bd. 14, Nr. 5, Sep. 1999, S. 771–780.

[Fri89] J. Friedman: *Regularized Discriminant Analysis*, *Journal of American Statistical Association*, Bd. 84, Nr. 405, März 1989, S. 165–175.

[Fuk90] K. Fukunaga: *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, 2. Ausg., 1990.

[Ger92] A. Gersho, R. Gray: *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

[Gev01] T. Gevers: *Color-Based Retrieval*, in M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 2, Springer-Verlag, London, 2001, S. 11–49.

[Gol00] S. Goldman, Y. Zhou: *Enhancing Supervised Learning with Unlabeled Data*, in *Proc. of the Seventeenth International Conference on Machine Learning*, Stanford, CA, USA, Juni 2000, S. 327–334.

[Gon02] R. Gonzalez, R. Woods: *Digital Image Processing*, Prentice-Hall, New Jersey, 2. Ausg., 2002.

[Gou01] V. Gouet, N. Boujemaa: *Object-Based Queries Using Color Points of Interest*, in *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 2001)*, Kauai, Hawaii, USA, Dez. 2001, S. 30–36.

[Gud95] V. Gudivada, V. Raghavan: *Content-Based Image Retrieval Systems*, *IEEE Computer*, Bd. 28, Nr. 9, Sep. 1995, S. 18–22.

[Gut84] A. Guttman: *R-Trees: A Dynamic Index Structure for Spatial Searching*, in *Proc. of ACM Sigmond International Conference on Management of Data*, Boston, MA, Juni 1984, S. 47–57.

[Har73] R. Haralick, K. Shanmugam, I. Dinstein: *Textural Features for Image Classification*, *IEEE Transactions on Systems, Man and Cybernetics*, Bd. 3, Nr. 6, Nov. 1973, S. 610–621.

[Har79] R. Haralick: *Statistical and Structural Approaches to Texture*, *Proc. of the IEEE*, Bd. 67, Nr. 5, Mai 1979, S. 786–804.

[Har88] C. Harris, M. Stephens: *A Combined Corner and Edge Detector*, in *Proc. of 4th Alvey Vision Conference*, Manchester, Aug. 1988, S. 147–151.

[Hon00] P. Hong, Q. Tian, T. Huang: *Incorporate Support Vector Machines to Content-Based Image Retrieval with Relevant Feedback,* in *Proc. of IEEE International Conference on Image Processing*, Vancouver, Canada, Sep. 2000, S. 750–753.

[Hu62] M. Hu: *Visual Pattern Recognition by Moment Invariants*, *IRE Transaction on Information Theory*, Bd. 8, Nr. 2, Feb. 1962, S. 179–187.

[Hua96] T. Huang, S. Mehrotra, K. Ramchandran: *Multimedia Analysis and Retrieval System (MARS) Project*, in *Proc. of the 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, März 1996, S. 0.

[Hua97] J. Huang, S. Kumar, M. Mitra, W. Zhu, R. Zabih: *Image Indexing Using Color Correlograms*, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, Juni 1997, S. 762–768.

[Hua02] T. Huang, X. Zhou, M. Nakazato, I. Cohen, Y. Wu: *Learning in Content-Based Image Retrieval*, in *Proc. of 2nd International Conference on Development and Learning*, Cambridge, MA, Juni 2002, S. 155–164.

[Ish98] Y. Ishikawa, R. Subramanya, C. Faloutsos: *MindReader: Querying Databases Through Multiple Examples*, in *Proc. of 24th International Conference on Very Large Data Bases, VLDB*, New York, NY, Aug. 1998, S. 218–227.

[ISO02] ISO/IEC JTC1/SC29/WG11: *MPEG-7 Overview*, Doc. N4980, Juli 2002.

[Jai88] A. Jain, R. Dubes: *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[Jai96] A. Jain, A. Vailaya: *Image Retrieval Using Color and Shape, Pattern Recognition*, Bd. 29, Nr. 8, Aug. 1996, S. 1233–1244.

[Jol86] I. Jolliffe: *Principal Component Analysis*, Springer-Verlag, New York, 1986.

[Jol01] J.-M. Jolion: *Feature Similarity*, in M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 3, Springer-Verlag, London, 2001, S. 51–85.

[Jun98] N. Jungclaus: *Integration verteilter Systeme zur Mensch-Maschine-Kommunikation*, Dissertation, Universität Bielefeld, Technische Fakultät, Okt. 1998.

[Käm02] T. Kämpfe, T. Käster, M. Pfeiffer, H. Ritter, G. Sagerer: *INDI – Intelligent Database Navigation by Interactive and Intuitive Content-Based Image Retrieval*, in *Proc. of IEEE International Conference on Image Processing*, Bd. III, Rochester, NY, Sep. 2002, S. 921–924.

[Käs01] T. Käster: *Konzeption und Implementierung eines SQL-Datenbank-Backends zur Speicherung von Multimediadaten*, Diplomarbeit, Angewandte Informatik, Technische Fakultät, Universität Bielefeld, 2001.

[Käs03a] T. Käster, M. Pfeiffer, C. Bauckhage, G. Sagerer: *Combining Speech and Haptics for Intuitive and Efficient Navigation through Image Databases*, in *Proc. of International Conference on Multimodal Interfaces*, ACM, Vancouver, Canada, Nov. 2003, S. 180–187.

[Käs03b] T. Käster, V. Wendt, G. Sagerer: *Comparing Clustering Methods for Database Categorization in Image Retrieval*, in B. Michaelis, G. Krell (Hrsg.): *Pattern Recognition; 25th DAGM Symposium, Magdeburg, Sep. 2003. Proceedings*, Lecture Notes in Computer Science 2781, Springer-Verlag, Heidelberg, Deutschland, 2003, S. 228–235.

[Käs04] T. Käster, M. Pfeiffer, C. Bauckhage, G.Sagerer: *Intelligent Navigation in Image Databases*, *KI Zeitschrift*, Bd. 18, Nr. 4, Nov. 2004, S. 24–43.

[Khe02] M. Kherfi, D. Ziou, A. Bernardi: *Learning from Negative Example in Relevance Feedback for Content-Based Image Retrieval*, in *Proc. of IEEE International Conference on Pattern Recognition*, Bd. 2, Québec, Canada, Aug. 2002, S. 933–936.

[Knu73] D. Knuth: *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.

[Koh00] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela: *Self Organization of a Massive Document Collection*, *IEEE Transactions on Neural Networks*, Bd. 11, Nr. 3, Mai 2000, S. 574–585.

[Kos03] M. Koskela, J. Laaksonen: *Using Long-Term Learning to Improve Efficiency of Content-Based Image Retrieval*, in *Proc. of 3rd International Workshop on Pattern Recognition in Information Systems*, Angers, France, April 2003, S. 72–79.

[Kri99] S. Krishnamachari, M. Abdel-Mottaleb: *Hierarchical Clustering Algorithm for Fast Image Retrieval*, in *Proc. of IS&T/SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, Jan. 1999, S. 427–435.

[La 98] M. La Cascia, S. Sethi, S. Sclaroff: *Combining Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web*, in *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries*, Santa Barbara, CA, Juni 1998, S. 24–28.

[Laa00] J. Laaksonen, M. Koskela, S. Laakso, E. Oja: *PicSOM – Content-Based Image Retrieval with Self-Organizing Maps*, *Pattern Recognition Letters*, Bd. 21, Nr. 13-14, Dez. 2000, S. 1199–1207.

[Laa01] J. Laaksonen, M. Koskela, S. Laakso, E. Oja: *Self-Organising Maps as a Relevance Feedback Technique in Content-Based Image Retrieval*, *Pattern Analysis and Applications*, Bd. 4, Nr. 2-3, Juni 2001, S. 140–152.

[Lan95] S. Lang, P. Lockemann: *Datenbankeinsatz*, Springer-Verlag, Berlin Heidelberg New York, 1995.

[Le 02] B. Le Saux, N. Boujemaa: *Unsupervised Robust Clustering for Image Database Categorization*, in *Proc. of IEEE International Conference on Pattern Recognition*, Québec, Canada, Aug. 2002.

[Lew01] M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Springer-Verlag, London, 2001.

[Lin80] Y. Linde, A. Buzo, R. Gray: *An Algorithm for Vector Quantizer Design*, *IEEE Transactions on Communications*, Bd. 28, Nr. 1, Jan. 1980, S. 84–95.

[Llo82] S. Lloyd: *Least Squares Quantization in PCM*, *IEEE Transactions on Information Theory*, Bd. 28, Nr. 2, März 1982, S. 129–137.

[Löm02] F. Lömker, G. Sagerer: *A Multimodal System for Object Learning*, in L. V. Gool (Hrsg.): *Pattern Recognition, 24th DAGM Symposium, Zurich, Switzerland*, Lecture Notes in Computer Science 2449, Springer-Verlag, Berlin, Sep. 2002, S. 490–497.

[Lon98] S. Loncaric: *A Survey of Shape Analysis Techniques*, *Pattern Recognition*, Bd. 31, Nr. 8, Aug. 1998, S. 983–1001.

[Lou00] E. Loupias, N. Sebe, S. Bres, J.-M. Jolion: *Wavelet-Based Salient Points for Image Retrieval*, in *Proc. of IEEE International Conference on Image Processing*, Bd. 2, Vancouver, Canada, Sep. 2000, S. 518–521.

[Luc01] L. Lucchese, S. Mitra: *Color Image Segmentation: A State-Of-The-Art Survey*, in *Proc. of The Indian National Science Academy*, Bd. 67, A, New Delhi, India, März 2001, S. 207–221.

[Ma97a] W. Ma, B. Manjunath: *Edge Flow: A Framework of Boundary Detection and Image Segmentation*, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, Juni 1997, S. 744–749.

[Ma97b] W. Ma, B. Manjunath: *NETRA: A Toolbox for Navigating Large Image Databases*, in *Proc. of IEEE International Conference on Image Processing*, Bd. 1, Washington, DC, Okt. 1997, S. 568–571.

[Mac67] J. MacQueen: *Some Methods for Classification and Analysis of Multivariate Observations*, in L. M. L. Cam, J. Neyman (Hrsg.): *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Bd. 1, 1967, S. 281–296.

[Mal89] S. Mallat: *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 11, Nr. 7, Juli 1989, S. 674–693.

[Mal03] D. Maltoni, D. Maio, A. Jain, S. Prabhakar: *Handbook of Fingerprint Recognition*, Springer Professional Computing, Springer-Verlag, New York, 1. Ausg., Juni 2003.

[Man96] B. Manjunath, W. Ma: *Texture Features for Browsing and Retrieval of Image Data*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 18, Nr. 8, Aug. 1996, S. 837–842.

[Man01] B. Manjunath, J.-R. Ohm, V. Vasudevan, A. Yamada: *Color and Texture Descriptors*, *IEEE Transactions on Circuits and Systems for Video Technology*, Bd. 11, Nr. 6, Juni 2001, S. 703–715.

[Mar91] T. Martinetz, K. Schulten: *A Neural Gas Network Learns Topologies*, in T. Kohonen, K. Mäkisara, O. Simula, J. Kangas (Hrsg.): *Proc. of IEEE International Conference on Artificial Neural Networks*, Bd. 1, Elsevier, Amsterdam, Holland, 1991, S. 397–407.

[Mar93] T. Martinetz, S. Berkovich, K. Schulten: *"Neural Gas" Network for Vector Quantization and its Application to Time-Series Prediction*, *IEEE Transactions on Neural Networks*, Bd. 4, Nr. 4, Juli 1993, S. 558–569.

[Mar01] D. Martin, C. Fowlkes, D. Tal, J. Malik: *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, in *Proc. of IEEE International Conference on Computer Vision*, Bd. 2, Vancouver, Canada, Juli 2001, S. 416–423.

[Mat97] G. Matthiessen, M. Unterstein: *Relationale Datenbanken und SQL*, Addison-Wesley, München, 1. Ausg., 1997.

[Mau02] U. Maulik, S. Bandyopadhyay: *Performance Evaluation of Some Clustering Algorithms and Validity Indices*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 24, Nr. 12, Dez. 2002, S. 1650–1654.

[Meh97] B. Mehtre, M. Kankanhalli, W. Lee: *Shape Measures for Content-Based Image Retrieval: A Comparison*, *Information Processing and Management*, Bd. 33, Nr. 3, Mai 1997, S. 319–337.

[Mei02] T. Meiers, T. Sikora, I. Keller: *Hierarchical Image Database Browsing Enviroment with Embedded Relevance Feedback,* in *Proc. of IEEE International Conference on Image Processing*, Bd. II, Rochester, NY, Sep. 2002, S. 593–596.

[Mes99] K. Messer, J. Kittler: *A Region-Based Image Database System Using Colour and Texture*, *Pattern Recognition Letters*, Bd. 20, Nr. 11-13, Nov. 1999, S. 1323–1330.

[Mil85] G. Milligan, M. Cooper: *An Examination of Procedures for Determining the Number of Clusters in a Dataset*, *Psychometrika*, Bd. 50, Nr. 2, Juni 1985, S. 159–179.

[Min96] T. Minka, R. Picard: *Interactive Learning Using a "Society of Models"*, in *Proc. of International Conference on Computer Vision and Pattern Recognition*, San Francisco, Ca, Juni 1996, S. 447–452.

[Mog99] B. Moghaddam, H. Biermann, D. Margaritis: *Defining Image Content with Multiple Regions-of-Interest*, in *Proc. of IEEE Workshop on Content-Based Access of Image and Video Libraries*, Fort Collins, Colorado, Juni 1999, S. 89–96.

[Mok92] F. Mokhtarian, A. Mackworth: *A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 14, Nr. 8, Aug. 1992, S. 789–805.

[Mok96] F. Mokhtarian, S. Abbasi, J. Kittler: *Efficient and Robust Retrieval by Shape Content through Curvature Scale Space*, in *Proc. of International Workshop on Image Databases and Multimedia Search*, Amsterdam, Aug. 1996, S. 35–42.

[Mon98] P. Montesinos, V. Gouet, R. Deriche: *Differential Invariants for Color Images*, in *Proc. of IEEE 14th International Conference on Pattern Recognition*, Bd. 1, Brisbane, Australia, Aug. 1998, S. 838–840.

[Mül01] H. Müller: *Suchen ohne Worte – Wie inhaltsbasierte Suche funktioniert, c't*, Bd. 15, 2001, S. 162–167.

[Mül02] H. Müller, S. Marchand-Maillet, T. Pun: *The Truth about Corel – Evaluation in Image Retrieval*, in M. Lew, N. Sebe, J. Eakins (Hrsg.): *Image and Video Retrieval*, Lecture Notes in Computer Science 2383, Springer-Verlag, 2002, S. 38–49.

[Nen96] S. Nene, S. Nayar, H. Murase: *Columbia Object Image Library (COIL-100)*, CUCS-006-96, Department of Computer Science, Columbia University, New York, NY, Feb. 1996.

[Ng96] R. Ng, A. Sedighian: *Evaluating Multi-Dimensional Indexing Structures for Images Transformed by Principle Component Analysis*, in *Proc. of SPIE/IS&T Conference Storage Retrieval Image Video Databases IV*, Bd. 2670, San Jose, CA, Feb. 1996, S. 50–61.

[Nib93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos: *The QBIC Project: Querying Images by Content Using Color, Texture and Shape*, in *Proc. of SPIE Conference on Storage and Retrieval for Image and Video Databases*, Bd. 1908, April 1993, S. 173–187.

[Nie83] H. Niemann: *Klassifikation von Mustern*, Springer-Verlag, Berlin, 1983.

[Ort97] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, T. Huang: *Supporting Similarity Queries in MARS*, in *Proc. of ACM Conference on Multimedia*, Seattle, Washington, USA, Nov. 1997, S. 403–413.

[Pas96] G. Pass, R. Zabih, J. Miller: *Comparing Images Using Color Coherence Vectors*, in *Proc. of ACM International Conference on Multimedia*, Boston, MA, Nov. 1996, S. 65–73.

[Pen94] A. Pentland, B. Moghaddam, T. Starner: *View-Based and Modular Eigenspaces for Face Recognition*, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, Seattle, Juni 1994, S. 84–91.

[Pen96] A. Pentland, R. Picard, S. Sclaroff: *Photobook: Content-Based Manipulation of Image Databases*, *International Journal of Computer Vision*, Bd. 18, Nr. 3, Juni 1996, S. 233–254.

[Pfe06] M. Pfeiffer: *Architektur eines multimodalen Forschungssystems zur iterativen inhaltsbasierten Bildersuche*, Dissertation, Universität Bielefeld, Technische Fakultät, erscheint voraussichtlich 2005/2006.

[Pic96] R. Picard, T. Minka, M. Szummer: *Modeling User Subjectivity in Image Libraries*, in *Proc. of IEEE International Conference on Image Processing*, Lausanne, Sep. 1996, S. 777–780.

[Por99] K. Porkaew, M. Ortega, S. Mehrotra: *Query Reformulation for Content Based Multimedia Retrieval in MARS*, in *Proc. of IEEE International Conference on Multimedia Computing and Systems*, Bd. 2, Florence, Italy, Juni 1999, S. 747–751.

[Poy97] C. Poynton: *Frequently Asked Questions about Color*, 1997, Verfügbar unter: http://www.poynton.com/ColorFAQ.html.

[Puz99] J. Puzicha, Y. Rubner, C. Tomasi, J. Buhmann: *Empirical Evaluation of Dissimilarity Measures for Color and Texture*, in *Proc. of IEEE International Conference on Computer Vision*, Bd. 2, Sep. 1999, S. 1165–1172.

[Qia02] F. Qian, M. Li, L. Zhang, H.-J. Zhand, B. Zhang: *Gaussian Mixture Model for Relevance Feedback in Image Retrieval*, in *Proc. of IEEE International Conference on Multimedia and Expo*, Bd. 1, Lausanne, Switzerland, Aug. 2002, S. 229–232.

[Reh98] V. Rehrmann, L. Priese: *Fast and Robust Segmentation of Natural Color Scenes*, in *Proc. of the Third Asian Conference on Computer Vision*, Bd. 1, Hong Kong, Jan. 1998, S. 598–606.

[Roc71] J. Rocchio: *Relevance Feedback in Information Retrieval*, in G. Salton (Hrsg.): *The SMART Retrieval System*, Automatic Computation, Prentice Hall, Englewood Cliffs, New Jersey, 1971, S. 313–323.

[Rub01] Y. Rubner, C. Tomasi: *Perceptual Metrics for Image Database Navigation*, Kluwer Academic Publisher, 2001.

[Rui97] Y. Rui, T. Huang, S. Mehrotra: *Content-Based Image Retrieval With Relevance Feedback in MARS*, in *Proc. of IEEE International Conference on Image Processing*, Santa Barbara, CA, Okt. 1997, S. 815–818.

[Rui98] Y. Rui, T. Huang: *Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, IEEE Transactions on Circuits and Video Technology, Special Issue on Segmentation Description, and Retrieval of Video Content*, Bd. 8, Nr. 5, Sep. 1998, S. 644–655.

[Rui99] Y. Rui, T. Huang, S. Chang: *Image Retrieval: Current Techniques, Promising Directions and Open Issues, Journal of Visual Communication and Image Representation*, Bd. 10, Nr. 4, April 1999, S. 39–62.

[Rui00] Y. Rui, T. Huang: *Optimizing Learning in Image Retrieval*, in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, Hilton Head, USA, Juni 2000, S. 236–243.

[Rui01] Y. Rui, T. Huang: *Relevance Feedback Techniques in Image Retrieval*, in M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 9, Springer-Verlag, London, 2001, S. 221–258.

[Sal83] G. Salton, M. McGill: *Introduction to Modern Information Retrieval*, McGraw-Hill Advanced Computer Science Series, 1983.

[Sal89] G. Salton (Hrsg.): *Automatic Text Processing*, Addison-Wesley, 1989.

[Sam95] H. Samet: *Spatial Data Structures*, in W. Kim (Hrsg.): *Modern Database Systems: The Object Model, Interoperability, and Beyond*, Addison Wesley/ACM Press, 1995, S. 361–385.

[Sch97] C. Schmid, R. Mohr: *Local Grayvalue Invariants for Image Retrieval*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 19, Nr. 5, Mai 1997, S. 530–535.

[Sch00] C. Schmid, R. Mohr, C. Bauckhage: *Evaluation of Interest Point Detectors*, *International Journal of Computer Vision*, Bd. 37, Nr. 2, Juni 2000, S. 151–172.

[Seb01] N. Sebe, M. Lew: *Texture Features for Content-Based Retrieval*, in M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 3, Springer-Verlag, London, 2001, S. 51–85.

[Seb02] N. Sebe, Q. Tian, E. Loupias, M. Lew, T. Huan: *Evaluation of Salient Point Techniques*, in M. Lew, N. Sebe, J. Eakins (Hrsg.): *Pattern Recognition*, Lecture Notes in Computer Science 2383, Springer-Verlag, 2002, S. 365–377.

[See01] M. Seeger: *Learning with Labeled and Unlabeled Data*, Institute for Adaptive and Neural Computation, University of Edinburgh, Feb. 2001.

[Sel87] T. Sellis, N. Roussopoulos, C. Faloutsos: *The R^+-Tree: A Dynamic Index for Multi-Dimensional Objects*, in *Proc. of the 13th International Conference on Very Large Data Bases*, Brighton, England, Sep. 1987, S. 507–518.

[Ser98] S. Servetto, Y. Rui, K. Ramchandran, T. Huang: *A Region-Based Representation of Images in MARS*, *VLSI Signal Processing Systems*, Bd. 20, Nr. 1-2, Okt. 1998, S. 137–150, Special Issue on Multimedia Signal Processing.

[Sig01] S. Siggelkow, M. Schael, H. Burkhardt: *SIMBA – Search IMages By Appearance*, *Lecture Notes in Computer Science*, Bd. 2191, 2001, S. 9–16.

[Sig02] S. Siggelkow: *Feature Histograms for Content-Based Image Retrieval*, Dissertation, Albert-Ludwigs-Universität Freiburg, Fakultät für Angewandte Wissenschaften, 2002.

[Sik01] T. Sikora: *The MPEG-7 Visual Standard for Content Description – An Overview*, *IEEE Transactions on Circuits and Systems for Video Technology*, Bd. 11, Nr. 6, Juni 2001, S. 696–702.

[Sme00] A. Smeulders, A. Gupta: *Content-Based Image Retrieval at the End of the Early Years*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 22, Nr. 12, Dez. 2000, S. 1349–1379.

[Smi94] J. Smith, S.-F. Chang: *Transform Features for Texture Classification and Discrimination in Large Image Databases*, in *Proc. of IEEE International Conference on Image Processing*, Bd. 3, Austin, Texas, Nov. 1994, S. 407–411.

[Squ99] D. M. Squire, W. Müller, H. Müller, J. Raki: *Content-Based Query of Image Databases, Inspirations from Text Retrieval: Inverted Files, Frequency-Based Weights and Relevance Feedback*, in *Proc. of 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, Juni 1999, S. 549–556.

[Str94] M. Stricker, M. Swain: *The Capacity of Color Histogram Indexing*, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, Juni 1994, S. 704–708.

[Str95] M. Stricker, M. Orengo: *Similarity of Color Images*, in *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)*, 1995, S. 381–392.

[Str97] M. Stricker, A. Dimai: *Spectral Covariance and Fuzzy Regions for Image Indexing*, *Machine Vision and Applications*, Bd. 10, 1997, S. 66–73.

[Swa91] M. Swain, D. Ballard: *Color Indexing*, *International Journal of Computer Vision*, Bd. 7, Nr. 1, 1991, S. 11–32.

[Tak98] T. Takahashi, S. Nakanishi, Y. Kuno, Y. Shirai: *Helping Computer Vision by Verbal and Nonverbal Communication*, in *Proc. of IEEE International Conference on Pattern Recognition*, Bd. 2, Brisbane, Australia, Aug. 1998, S. 1216–1218.

[Tam78] H. Tamura, S. Mori, T. Yamawaki: *Texture Features Corresponding to Visual Perception*, *IEEE Transactions on Systems, Man and Cybernetics*, Bd. 8, Nr. 6, Juni 1978, S. 460–473.

[Tam84] H. Tamura, N. Yokoya: *Image Database Systems: A Survey*, *Pattern Recognition*, Bd. 17, Nr. 1, 1984, S. 29–43.

[Tax99] D. Tax, R. Duin: *Support Vector Domain Description*, *Pattern Recognition Letters*, Bd. 20, Nr. 11-13, Nov. 1999, S. 1191–1199.

[Tea80] M. R. Teague: *Image Analysis Via the General Theory of Moments*, *Journal of Optical Society of America*, Bd. 70, Nr. 8, Aug. 1980, S. 920–930.

[Tia00] Q. Tian, T. Huang: *Combine User Defined Region-Of-Interest and Spatial Layout for Image Retrieval*, in *Proc. of IEEE International Conference on Image Processing*, Bd. III, Vancouver, Canada, Sep. 2000, S. 746–749.

[Tie00] K. Tieu, P. Viola: *Boosting Image Retrieval*, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, Juni 2000, S. 228–235.

[Tuc93] M. Tuceryan, A. Jain: *Texture Analysis*, in C. Chen, L. Pau, P. Wang (Hrsg.): *Handbook of Pattern Recognition and Computer Vision*, Kap. 2.1, World Scientific, Singapore, 1993, S. 235–276.

[Uns86] M. Unser: *Sum and Difference Histograms for Texture Classification*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 8, Nr. 1, Jan. 1986, S. 118–125.

[Vap95] V. Vapnik: *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[Wac98] S. Wachsmuth, G. Fink, G. Sagerer: *Integration of Parsing and Incremental Speech Recognition*, in *Proc. of the European Signal Processing Conference*, Bd. 1, Rhodes, Sep. 1998, S. 371–375.

[Wac02] S. Wachsmuth, G. Sagerer: *Bayesian Networks for Speech and Image Integration*, in *Proc. of 18th National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada, Juli 2002, S. 300–306.

[Wag99] T. Wagner: *Texture Analysis*, in B. Jähne, H. Haussecker, P. Geissler (Hrsg.): *Handbook of Computer Vision and Applications*, Bd. 2, Kap. 12, Academic Press, 1999, S. 275–308.

[Wan01] J. Wang, J. Li, G. Wiederhold: *SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 23, Nr. 9, Sep. 2001, S. 947–963.

[Whi96a] D. White, R. Jain: *Similarity Indexing: Algorithms and Performance*, in *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)*, 1996, S. 62–73.

[Whi96b] D. White, R. Jain: *Similarity Indexing with the SS-Tree*, in *Proc. of the 12th International Conference on Data Engineering*, New Orleans, USA, Feb. 1996, S. 516–523.

[Wol00] C. Wolf, J. Jolion, W. Kropatsch, H. Bischof: *Content-Based Image Retrieval Using Interest Points and Texture Features*, in *Proc. of IEEE International Conference on Pattern Recognition*, Bd. 4, Barcelona, Spain, Sep. 2000, S. 234–237.

[Wu00] Y. Wu, Q. Tian, T. Huang: *Integrating Unlabeled Images for Image Retrieval Based on Relevance Feedback*, in *Proc. of IEEE International Conference on Pattern Recognition*, Bd. 1, Barcelona, Spain, Sep. 2000, S. 1021–1024.

[Wys82] G. Wyszecki, W. Stiles: *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley & Sons, New York, 2. Ausg., 1982.

[Zah72] C. Zahn, R. Roskies: *Fourier Descriptors for Plane Closed Curves*, *IEEE Transactions on Computers*, Bd. C-21, Nr. 3, März 1972, S. 269–281.

[Zei96] E. Zeidler (Hrsg.): *Teubner – Taschenbuch der Mathematik*, B.G. Teubner, 1996.

[Zel94] M. Zeller: *Flinkes Wellenspiel – Signalverarbeitung mit Wavelets*, *c't*, Bd. 11, 1994, S. 258–264.

[Zha95] H. Zhang, Y. Gong, C. Low, S. Smollar: *Image Retrieval Based on Color Features: An Evaluation Study*, in *Proc. of SPIE Digital Image Storage and Archiving Systems*, Bd. 2606, Bellingham, Washington, 1995, S. 212–220.

[Zha01a] D. Zhang, G. Lu: *A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures*, in *Proc. of International Conference on Intelligent Multimedia and Distance Education*, Fargo, ND, USA, Juni 2001, S. 1–9.

[Zha01b] D. Zhang, G. Lu: *Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study*, in *Proc. of IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, Aug. 2001, S. 317–320.

[Zha03] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld: *Face Recognition: A Literature Survey*, *ACM Computing Surveys*, Bd. 35, Nr. 4, Dez. 2003, S. 399–458.

[Zho01] X. Zhou, T. Huang: *Small Sample Learning during Multimedia Retrieval Using BiasMap*, in *Proc. of Computer Vision and Pattern Recognition*, Bd. 1, Kauai, Hawaii, Dez. 2001, S. 11–17.

[Zho02] X. Zhou, T. Huang: *Unifying Keywords and Visual Contents in Image Retrieval*, *IEEE Multimedia*, Bd. 9, Nr. 2, April/März 2002, S. 23–33.

[Zho03a] X. Zhou, T. Huang: *Relevance Feedback in Image Retrieval: A Comprehensive Review*, *Multimedia Systems*, Bd. 8, Nr. 6, April 2003, S. 536–544.

[Zho03b] X. Zhou, Y. Rui, T. Huang: *Exploration of Visual Data*, Kluwer Academic, Boston/Dordrecht/New York/London, 2003.

[Zhu00] L. Zhu, A. Zhang: *Supporting Multi-Example Image Queries in Image Databases*, in *Proc. of IEEE International Conference on Multimedia and Expo*, New York City, NY, USA, Juli 2000, S. 697–700.