Suchmodellbasiertes Content-based Image Retrieval

Ähnlichkeitsdefinition, Anwendung und Automatisierung

eingereicht von:

Ing. Mag. Horst Eidenberger

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum socialium oeconomicarumque
(Dr. rer. soc. oec.)
Doktor der Sozial- und Wirtschaftswissenschaften

Sozial- und wirtschaftswissenschaftliche Fakultät
Universität Wien

Erstgutachter: Univ.-Prof. Dr. Christian Breiteneder
Zweitgutachter: ao. Univ.-Prof. Dr. Dieter Merkl

Wien im Juli 2000

Inhalt

1 Einleitung

Die explosive Entwicklung des Internets in den letzten Jahren ermöglichte es, relativ kostengünstig Datenbestände von Museen, Archiven, etc. multimedial aufbereitet einem größeren Publikum als bisher anzubieten. Daher werden von vielen Einrichtungen derzeit großen Anstrengungen unternommen, um vorhandene Bestände zu digitalisieren und in entsprechenden multimedialen Datenbanken, sei es für die Verwendung in Multimedia-Informationssystemen (z. B. als Lernprogramme, etc.) oder mit einer entsprechenden Suchschnittstelle versehen dem Endbenutzer direkt zur Verfügung zu stellen. Ganz besonders gilt das für Bilder, für die man auf Konferenzen beziehungsweise beim Studium von Publikationen, etc. den Eindruck gewinnen kann, daß im deutschen Sprachraum derzeit jede Institution ihr eigenes (zumindest eines!) Digitalisierungsprojekt betreibt. Der Reiz, die eigenen Bestände zumindest teilweise im weltumspannenden Netz – also, wie einigermaßen optimistisch gefolgert wird, weltweit – anzubieten, ist hier (vielleicht aufgrund des weitgehend sprachunabhängigen Mediums) besonders groß. Allerdings ist es, um sich in den so angelegten weiträumigen Informationssümpfen zurechtfinden zu können, notwendig, die Daten auch suchbar zu machen. Dies geschieht heute noch durchwegs durch textuelles Retrieval nach mit Datenobjekten verknüpften Schlagworten. Daraus ergeben sich die üblichen Probleme: (die Anwendbarkeit einschränkende) Sprachabhängigkeit, hoher Kostenaufwand zur Datenerfassung durch den Menschen, zwangsläufig Fehler in der Indizierung und geringe Chancen, falsch verknüpfte Daten wieder zu finden.

Aufgrund dieser Nachteile und der dieser Idee innewohnenden Faszination wäre es sehr interessant, über eine Möglichkeit zur inhaltsorientierten Suche, das heißt nach qualitativen Bildmerkmalen, zu verfügen. Man sucht also z. B. nach Bildern mit Urwaldlandschaften und nimmt dafür ein Beispielphoto aus dem letzten Urlaub. Diese Suchmethode wird als Content-based Image Retrieval (CBIR) bezeichnet, also die inhaltsbasierte Suche nach Bildern. Dadurch sollen die Möglichkeiten der reinen Textsuche um eine neue Komponente erweitert werden. Dem CBIR haften derzeit folgende Hauptnachteile an:

- Die Ergebnisse sind nicht so gut, wie sie sein sollten. Im Bereich künstlich hergestellter Bilder lassen sich zwar akzeptable Ergebnisse erzielen, dort jedoch, wo natürliche Bilder verwendet werden, verhindern technische Einflußfaktoren (Aufnahmewinkel, Lichteinstrahlung, etc.) oft noch gute Ergebnisse. Teilweise Mitschuld an der schlechten Qualität der Ergebnisse hat auch das bisher nicht sehr intensiv reflektierte Ähnlichkeitskonzept des CBIR.

- Schlechte Performance bei der Suche. Bilder werden anhand sogenannter Distanzfunktionen verglichen, die bisweilen sehr rechenintensiv sind. Daher kommt es beim Durchsuchen großer Datenmengen oft zu hohen Antwortzeiten. In der CBIR-Literatur

finden sich Lösungsansätze, die versuchen, die bestehenden Probleme durch Verringerung der Zahl der Distanzvergleiche zu verkleinern. Wesentliche Verbesserungen wurden dadurch aber bisher nicht erreicht.

- Komplizierte Benutzerschnittstellen. Die inhaltsorientierte Suche ist ein Vorgang, der von einer traditionellen Suche nach Stichworten sehr weit abweicht. Der Benutzer, der eine normale Suchmaske erwartet, ist, wenn er eine CBIR-Suchmaske sieht, zunächst verwirrt und versucht oft erst gar nicht, die Vorteile des CBIR zu nutzen, weil er es scheut, sich in die neuen Suchfunktionen einzuarbeiten.

In dieser Arbeit wird gezeigt, wie CBIR modellbasiert ablaufen kann. Dabei bezieht sich das Adjektiv „modellbasiert“ nicht auf die untersuchten Daten, sondern auf die Form der Ähnlichkeitsdefinition. Das heißt der gesamte Suchprozeß wird untersucht, um Gemeinsamkeiten im Ablauf zu finden, und es wird ein neues, einheitliches CBIR-Konzept zugrunde gelegt. Die bisher gültige Ähnlichkeitsdefinition wird verallgemeinert und erweitert, um Ergebnisse von höherer Qualität zu erzielen. Daneben wird mit der gleichzeitigen Suche nach vielen Bildeigenschaften mit einfachen, robusten Funktionen eine neue Vorgangsweise für CBIR vorgestellt. Das zentrale in dieser Arbeit vorgestellte Konzept sind die sogenannten Suchmodelle (Query models). Jeder Suche wird ein Modell zugrunde gelegt, das die für diese Suche spezifische Ähnlichkeit definiert. Die Abarbeitung erfolgt für alle Suchmodelle gleich. Zur Vereinfachung der Benutzerschnittstellen wurde das sogenannte Click & Refine-Modell entwickelt, das es dem Benutzer ermöglicht, allein durch sein binäres Relevanzurteil (relevant / nicht relevant) zu gefundenen Bildern seine Suchergebnisse iterativ zu verbessern. Die dadurch möglich werdende einfachere Gestaltung der Benutzerschnittstellen soll zusammen mit besseren Ergebnissen helfen, die Akzeptanz von CBIR-Systemen zu erhöhen.

In den folgenden Kapiteln wird gezeigt, wie alle diese Konzepte umgesetzt wurden. Manche Ideen, wie z. B. ein Teil der entwickelten Feature-Funktionen, sind spezifisch für den gewählten Anwendungsbereich: eine Bilddatenbank mit Wappenbildern, die überwiegende Mehrzahl ist jedoch allgemein für alle Arten von Bildern gültig. Im folgenden Kapitel werden die Grundlagen für diese Arbeit behandelt. Neben einer Einführung in das Information Retrieval für Bilder und in die Wappenkunde besteht es hauptsächlich aus einem Literaturüberblick über das Forschungsgebiet des CBIR. In der Einführung in das Information Retrieval für Bilder wird zuerst die Vorgangsweise beim Information Retrieval generell skizziert und dann auf die Besonderheiten des CBIR eingegangen. Im Literaturüberblick zu CBIR wird versucht, sowohl klassische Lösungsansätze als auch moderne, derzeit in der Erprobung befindliche Ansätze nebeneinander darzustellen. Im Kapitel 3 wird dargestellt, wie CBIR für Wappen realisiert wurde. Dazu werden die Methoden vorgestellt, mit denen in der Testdatenbank Bildeigenschaften (Features) aus Wappenbildern extrahiert wurden sowie die Vorgangsweise bei der suchmodellorientieren Recherche beschrieben. Außerdem werden in

einem eigenen Abschnitt die verwendeten Testdaten und Features statistisch analysiert, um die Qualitätsangaben bei den Auswertungen besser beurteilen zu können.

Im Kapitel 4 werden Algorithmen beschrieben, die aufbauend auf dem Konzept der Suchmodelle die oben angegebenen Probleme deutlich verkleinern. In Abschnitt 4.1 wird ein Algorithmus zur optimalen Gewichtung von Suchergebnissen vorgestellt, der die Qualität der Suchergebnisse signifikant verbessert. Abschnitt 4.2 beschreibt einen Algorithmus, der durch die geschickte Verarbeitung von Suchmodellen die Antwortzeiten von CBIR-Suchen stark senkt. Im Kapitel 5 werden Algorithmen zur automatischen Erzeugung von Suchmodellen aus einem beziehungsweise mehreren Suchbildern vorgestellt, die so gute Ergebnisse liefern, daß die Realisierung des Click & Refine-Konzeptes realistisch wird. Die vorliegende Dissertation versucht, ausgehend vom derzeitigen Entwicklungsstand des CBIR durch das Suchmodell–Konzept und die modellbasierte Vorgangsweise, operationalisiert in einer Anzahl von Lösungsalgorithmen für verschiedene CBIR-Probleme, ein höheres Niveau bei CBIR-Systemen zu erreichen.

Die in dieser Arbeit vorgestellten Konzepte wurden großteils in den folgenden Publikationen veröffentlicht:

- Features für CBIR nach Wappen in [7]
- Verwendung von Suchmodellen zur Suche in [5]
- Automatische Gewichtung von Suchmodellen in [8]
- Performance-optimierte Reihung von Suchmodellen in [9]
- Generierung von Suchmodellen für Bilder und Gruppen in [6]

Abschließend bleibt dem Autor noch die erfreuliche Aufgabe, sich bei allen zu bedanken, die (wissentlich oder nicht) dazu mitgeholfen haben, daß diese (hoffentlich nicht allzu entbehrliche) Arbeit geschrieben sowie die dazu nötigen praktischen Arbeiten in den beiden letzten Jahren durchgeführt werden konnten.

2 Grundlagen

In diesem Kapitel werden die Grundlagen für diese Arbeit behandelt. Es wird beschrieben, wie Information Retrieval grundsätzlich und speziell für Bilder (CBIR) abläuft, welche Methoden im CBIR angewendet werden und nach welchen Regeln die Wappenkunde (Heraldik) strukturiert ist.

2.1 Information Retrieval für Bilder

Infolge der Entwicklung moderner Computer- und Multimediatechnik sowie neuer Technologien, wie des World Wide Web, entwickelte sich das Bedürfnis, multimediale Objekte (Bilder, Videosequenzen, etc.) zu sammeln und suchbar zu machen. Traditionell erfolgt eine Datenbanksuche nach Stich- oder Schlagworten, die bei der Datenerfassung mit dem eingelagerten Objekt durch einen menschlichen Erfasser verknüpft werden. Diese Methode hat zwei wesentliche Nachteile ([38]):

- Die Verknüpfung durch einen Menschen ist subjektiv und fehleranfällig. Hinzu kommt, daß dort, wo sie nicht von einem Experten vorgenommen wird, die Beschreibung nur formal sein kann.

- Die Erfassung durch Menschen ist kostenintensiv.

Die Methoden des Content-based Image Retrieval (CBIR) dienen dazu, Bilder anhand ihrer Inhalte suchbar zu machen; CBIR ist weniger als Substitut sondern eher als Ergänzung zur herkömmlichen Schlagwortsuche zu sehen. Der folgende Abschnitt 2.1.1 führt in die Grundlagen des Information Retrievals ein und Abschnitt 2.1.2 behandelt die Grundlagen des CBIR (Bereiche, Begriffe, etc.).

2.1.1 Information Retrieval

Information Retrieval (IR; [15]) widmet sich dem, durch die fortschreitende Perfektionierung der weltweiten Informationserzeugungs- und -weiterleitungsmaschinerie immer bedeutender werdenden, Problem des Auffindens relevanter Informationen. Dabei zeichnet sich IR gegenüber den herkömmlichen Abfragen in Datenbanken vor allem durch nicht-exakte Frageformulierung und unsicheres Wissen aus. Der Begriff Information läßt sich folgendermaßen definieren ([15]): "Information ist jene Teilmenge von Wissen, die von jemanden in einer konkreten Situation zur Lösung von Problemen benötigt wird". Dabei werden unter Wissen Daten mit semantischen Bezügen verstanden.

Zentrales Element des (textorientierten) IR ist das Dokument. Hierbei werden üblicherweise drei verschiedene Sichtweisen auf Dokumente angeführt:

- Layout-Sicht: beschreibt die Darstellung eines Dokumentes.

- Logische Sicht: beschreibt die Struktur eines Dokumentes.

- Semantische Sicht: beschreibt den Inhalt eines Dokumentes.

Die semantische Sicht ist die für IR vor allem relevante. Im folgenden werden kurz einige IR-Modelle beschrieben sowie Überlegungen zu möglichen Meßmethoden für IR-Methoden angestellt.

2.1.1.1 IR-Modelle

IR-Modelle sind generell danach zu unterscheiden, ob sie auf statistischen Methoden (probabilistische IR-Modelle) beruhen oder nicht (nicht-probabilistische IR-Modelle). Bei den nicht-probabilistischen IR-Modellen sind vor allem folgende Ansätze hervorzuheben:

- Boolesches Retrieval: dabei besteht eine Frage aus Wörtern, die durch logische Operatoren verknüpft werden. Der Vorteil dieser Methode liegt darin, daß man unmittelbar nach dem Untersuchen eines Dokumentes beurteilen kann, ob es zur Ergebnismenge gehört oder nicht; nachteilig ist, daß keine Reihung der gefundenen Dokumente erfolgt und sich die Anzahl der Treffer für eine Frage nur schwer abschätzen läßt.

- Fuzzy-Retrieval: erweitert das Boolesche Retrieval um die Möglichkeit der Gewichtung, wodurch ein Ranking der Ergebnismenge möglich wird.

- Vektorraummodell (VRM): im VRM werden Dokumente und Fragen als Punkte in einem Vektorraum aufgefaßt, wobei die Basis dieses Raumes durch die einzelnen Terme der Datenbasis gebildet wird. Die Retrievalfunktion hat die Aufgabe, ähnliche Vektoren zum Fragevektor zu finden, wobei sich die gefundenen Dokumente nach ihrer Distanz ordnen lassen. Das Vektorraummodell ist unter anderem die methodische Grundlage von Content-based Image Retrieval.

- Dokumentenclusterung: diese Methode faßt ähnliche Dokumente zu Gruppen, sogenannten Clustern, zusammen und findet zu einer Frage passende Dokumente, indem in Clustern gesucht wird, deren Eigenschaften der Frage entsprechen. Die Clustermethode beruht auf der beweisbaren Überlegung, daß die Ähnlichkeit von für eine bestimmte Frage relevanten Dokumenten im allgemeinen größer ist als von zufällig gewählten.

Im Gegensatz zu den oben dargestellten Ansätzen versucht man durch probabilistische IR-Modelle, den verschiedenen Unsicherheitsfaktoren des IR durch statistische Methoden zu begegnen. Zentral ist das Konzept der Relevanz, wobei der Anwender die Relevanz ausgewählter Dokumente festlegt und das IR-System mithilfe dieser Informationen alle relevanten Dokumente in einer Datenbank sucht.

2.1.1.2 Evaluierung

Ziel der Evaluierung von IR-Methoden ist die Messung ihrer Effektivität, worunter vor allem die Qualität des Ergebnisses in Relation zum eingesetzten Aufwand verstanden wird. Dieser Abschnitt beschäftigt sich mit möglichen Bewertungsmaßen, wobei für jedes Maß zu klären ist, von welchem Standpunkt aus die Qualität einer Methode betrachtet wird:

1. Benutzerstandpunkt: Sichtweise eines einzelnen Benutzers. Benutzerorientierte Maße beziehen sich folglich auf die Erwartungen und Präferenzen des Benutzers.

2. Systemstandpunkt: globale Sichtweise. Systemorientierte Maße versuchen daher, eine von einzelnen Benutzerstandpunkten unabhängige Beurteilung vorzunehmen.

Retrievalergebnisse lassen sich zu Klassen, den sogenannten Distributionen, verallgemeinern, die dann zur Bewertung einer Methode herangezogen werden können. Dazu wird zuerst eine Rangordnung der Ergebnisse nach ihrem Gewicht gebildet, diese Ordnung dann aufgrund eines allgemeinen Relevanzurteils beurteilt und daraus eine Menge von Äquivalenzklassen von Distributionen abgeleitet.

Von den benutzerorientieren Maßen werden meist Precision und Recall verwendet:

$$precision = \frac{|REL \cap GEF|}{|GEF|} \tag{1}$$

$$recall = \frac{|REL \cap GEF|}{|REL|} \tag{2}$$

wobei GEF die gefundenen Dokumente und REL die in der Datenbank vorhandenen relevanten sind. Außerdem kann man noch das Fallout-Maß verwenden, das die Fähigkeit eines Systems bewertet, dem Benutzer irrelevante Dokumente vorzuenthalten:

$$fallout = \frac{|GEF - REL|}{|ALL - REL|} \tag{3}$$

wobei ALL die Gesamtzahl aller Dokumente ist. Das Hauptproblem der Berechnung dieser Kennzahlen ist die Bestimmung der Anzahl der relevante Dokumente. Dazu ist entweder

exakte Kenntnis der verwendeten Datenbank oder die Anwendung einer der folgenden Näherungsmethoden nötig:

- Verwendung einer repräsentativen Stichprobe und Hochrechnung auf die gesamte Datenbasis.

- Document-Source-Methode: Auswahl von für bestimmte Fragen relevanten Dokumenten und Bestimmung, wie oft diese Dokumente in der Ergebnismenge vorkommen. Aus der relativen Häufigkeit läßt sich der Recall ableiten.

- Frageerweiterung: dadurch werden weitere Dokumente gefunden, die vielleicht für die ursprüngliche Frage relevant sind.

- Abgleich mit externen Quellen: z. B. Heranziehung von Experten, etc.

Als Beispiel für ein systemorientiertes Maß sei das Nützlichkeitsmaß von Frei und Schäuble erwähnt ([13]). Es dient dem Vergleich zweier Retrievalverfahren anhand ihrer Antwortmengen und liefert neben einer Beurteilung der Effizienz auch noch eine Aussage über die statistische Signifikanz dieser Beurteilung.

2.1.2 Content-based Image Retrieval

CBIR beruht auf dem Vektorraummodell, wobei der Vektorraum in diesem Fall durch sogenannte Features aufgespannt wird. Unter dem zentralen Begriff des Features versteht man die zahlenmäßige Repräsentation von wesentlichen Bildeigenschaften oder -inhalten. Ein Feature eines Farbbildes könnte z. B. ein Farbhistogramm sein. Abbildung 1 zeigt beispielhaft einen dreidimensionalen Vektorraum der durch die Farben rot, grün und blau eines Farbfeatures gebildet wird.

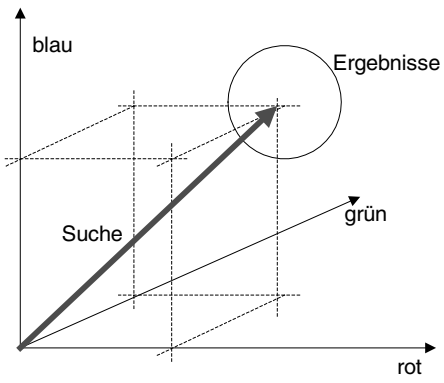

Abbildung 1: Dreidimensionaler Farbvektorraum

Die drei wesentliche Teilbereiche des CBIR sind (vgl. auch Abbildung 2):

- Extraktion von Bildinhalten (visual feature extraction)

- Indexierung der Featuredaten (multi-dimensional indexing)

- Entwurf von Suchsystemen (retrieval system design)

Sie werden detailliert im Literaturüberblick (Abschnitt 2.2) behandelt. Auf die Extraktion von Bildinhalten wird im Literaturüberblick im Abschnitt 2.2.2.1 eingegangen, Indizierung wird im Abschnitt 2.2.2.3 und Entwurf von Suchsystemen im Abschnitt 2.2.3 behandelt. Im allgemeinen werden folgende Arten von Suchen unterschieden:

- Retrieval by browsing (RBR): Suche anhand von Beispielen.

- Retrieval by objective attributes (ROA): Suche nach logischen oder Meta-Attributen beziehungsweise Kombinationen daraus.

- Retrieval by spatial constraints (RSC): solche Abfragen basieren auf den räumlichen Relationen von Bildobjekten.

- Retrieval by semantic attributes (RSA): Suchen nach explizit angegebenen Eigenschaften der gesuchten Bilder. Die Attribute können z. B. aus einer Liste ausgewählt oder als Stichworte eingegeben werden.

- Retrieval by feature similarity (RFS): Suchen nach Bildeigenschaften (Features)

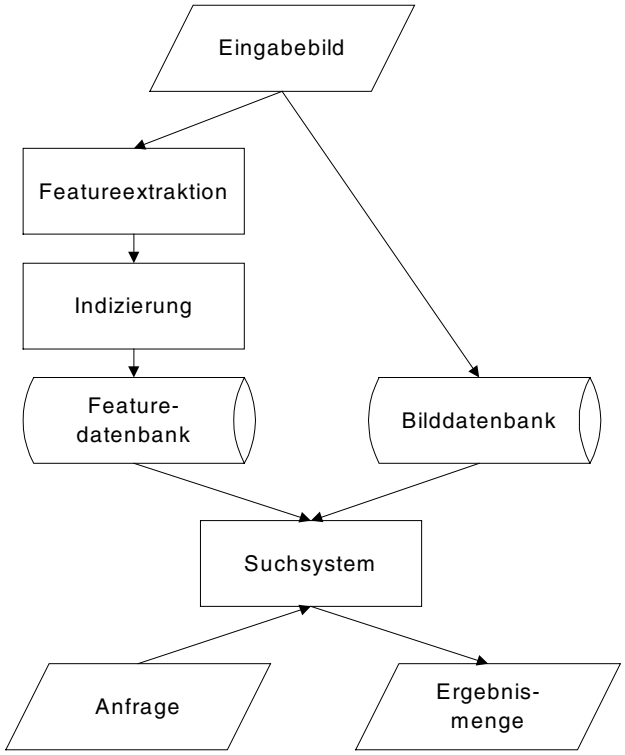

Abbildung 2: Elemente des CBIR

Bildmengen können im Hinblick auf CBIR anhand folgender Eigenschaften klassifiziert werden:

- Homogenität / Heterogenität ([32])

  Homogene Bildmengen (beziehungsweise Datenbanken) sind solche, für die es einen gemeinsamen inhaltlichen Kontext (ground truth) gibt. Sie beinhalten Bilder der gleichen Klasse. Bei Suchen in solchen Datenbanken ist die Annahme wahrnehmbarer Ähnlichkeit (perceptual similarity) zweier Bilder implizit klar. Der Designer eines Systems für solche Bilder wird auf die gemeinsamen Eigenschaften Rücksicht nehmen und entsprechend modellbasiert vorgehen.

  Datenbanken mit heterogenem Bildmaterial haben keinen gemeinsamen inhaltlichen Kontext. Retrieval Systeme für solche Datenbanken sollten daher ein großes Maß an Flexibilität aufweisen.

- Bilder mit natürlichem / künstlichem Ursprung ([48])

  Bilder, die nicht zumindest teilweise aus Symbolen bestehen, werden als natürliche Bilder bezeichnet. Der derzeitige Stand der CBIR-Technik erlaubt zufriedenstellende Ergebnisse vor allem für künstliche Bilder.

2.2 Literaturüberblick

Dieser Abschnitt liefert einen Überblick über die für CBIR verwendeten Techniken, etablierte Systeme und Anwendungsgebiete. Es wurde versucht, sowohl ein umfassendes Literaturverzeichnis zu erstellen als auch die verwendeten Techniken kurz, aber vollständig zu beschreiben. Im Abschnitt 2.2.1 wird auf den Begriff Ähnlichkeit eingegangen und einige Voraussetzungen für seine Messung abgehandelt. Abschnitt 2.2.2 faßt die Mehrzahl der derzeit für CBIR eingesetzten Methoden zusammen und Abschnitt 2.2.3 beschreibt die wichtigsten CBIR-Systeme.

2.2.1 Ähnlichkeit

Die Suche nach Bildinhalten läuft in der Regel so ab, daß der Benutzer bestimmt nach welchen Features er suchen will und die relative Bedeutung dieser Features durch Gewichte festlegt. Diesen Ansatz nennt man den computer-zentrischen Ansatz (computer centric approach; vgl. [39]). Mit ihm verbunden sind zwei wesentliche Probleme: die sogenannte semantische Lücke (semantic gap) zwischen den Konzepten der inhaltsorientierten Suche (high-level) und Features (low-level) ([39], [40], [42]) und die Subjektivität des Anwenders (subjectivity of human perception; [39]) Die Autoren von [40] gehen davon aus, daß durch einfache Features Ähnlichkeit nie ausreichend definiert werden kann und daher durch CBIR nur Korrelationen zwischen Suchbildern und Ergebnismengen herstellt werden können.

Das Problem der Subjektivität des Anwenders stellt sich so dar, daß verschiedene Personen (Rezipienten) oder dieselbe Person in verschiedenen Situationen visuelle Inhalte verschieden beurteilen. Dieses Problem existiert in mehreren Formen: verschiedene Personen können unterschiedlichen Wert auf bestimmte Eigenschaften, wie Farbe oder Textur, etc. legen; oder wenn sie auf dieselbe Eigenschaft Wert legen, können sie diese doch unterschiedlich wahrnehmen und beurteilen ([39]).

Die Bewertung von Ähnlichkeit kann danach unterschieden werden, ob sie ohne Fokussierung (pre-attentive) oder mit Fokussierung (attentive) erfolgt (siehe [41]). Im ersteren Fall erfolgt die Beurteilung der Ähnlichkeit von Bildern, ohne auf bestimmte Eigenschaften zu fokussieren und ohne die Miteinbeziehung des menschlichen Erkennungsapparates. Im zweiten Fall wird das Bild erkannt und auf besonders wichtige Bereiche (z. B. in Bildern von Gesichtern auf die Augen) fokussiert. Geringe Abweichungen (bezogen auf das Gesamtbild) reichen dann bereits aus, damit solche Bilder als überdurchschnittlich unähnlich beurteilt werden. Die CBIR-Forschung konzentriert sich (im Bereich von general purpose - Ansätzen) vor allem auf Ähnlichkeitsmessung ohne Fokussierung.

Was aber ist Ähnlichkeit (oder eigentlich, da meist das Ausmaß der Differenz gemessen wird, Unähnlichkeit)? Dafür gibt es verschiedene Ansätze, von denen der wohl bedeutendste das Konzept der sogenannten geometrischen Ähnlichkeit (geometric similarity; [43]) ist. Im

folgenden werden kurz die Axiome dargestellt, auf denen es beruht (metrische Axiome). Grundsätzlich ist bei der Ähnlichkeitsmessung zwischen der wahrgenommenen (perceived similarity) d und der bewerteten Ähnlichkeit (judged similarity) D zu unterscheiden. Wenn A und B die Featurevektoren der Bilder a und b sind, dann ist d(A,B) die wahrgenommene Distanz während die bewertete Distanz D(A,B) = g(d(A,B)), wobei g() eine passende monoton steigende Funktion ist. Es ist zu beachten, daß nur die bewertete Ähnlichkeit experimentell überprüfbar ist.

Nach dem geometrischen Ähnlichkeitsmodell müssen für eine Distanzfunktion d() folgende Axiome erfüllt sein:

1. Identität: d(A,A) = d(B,B); das kann überprüft werden, da es impliziert: D(A,A) = D(B,B)

2. Minimalität: d(A,B) >= d(A,A); impliziert (da g() monoton ist) D(A,B) >= D(A,A). Verschiedene Experimente mit gegensätzlichen Ergebnissen (siehe [43]) haben dazu geführt, daß die Bedeutung dieses Axioms angezweifelt wird.

3. Symmetrie: d(A,B) = d(B,A); impliziert D(A,B) = D(B,A). Das Symmetrie-Axiom wird angezweifelt, da Ausprägungen von Features (sogenannte Stimuli) oft unterschiedlich deutlich hervortreten. Im allgemeinen gilt, daß ein weniger deutlicher Stimulus einem deutlicheren gegenüber als ähnlicher wahrgenommen wird als umgekehrt.

4. Dreiecksungleichung (triangle inequality): d(A,B) + d(B,C) >= d(A,C); das schwächste Axiom, das, wie experimentell festgestellt wurde, zumindest für einige Stimuli sicher nicht erfüllt ist.

Eine Art von Distanzfunktionen, die diese Axiome erfüllt, sind die sogenannten Minkowski Distanzen:

$$d_p(A,B) = \left[\sum_i (A_i - B_i)^p \right]^{\frac{1}{p}} \tag{4}$$

wobei $A = \{A_1, ..., A_n\}$, $B = (B_1, ..., B_n)$ die Featurevektoren sind und $p>0$ eine Konstante ist, welche die Distanzfunktion charakterisiert.

Andere Distanzmaße sind die euklidische Distanz und die Mahalanobis-Distanz ([11], [16], [38]): Die euklidische Distanz ist das Quadrat der Minkowski-Distanz mit p = 2 und erfüllt also die Axiome der geometrischen Ähnlichkeit:

$$D_{euklid} = \sum_{i=1}^{n} (X_i - Y_i)^2 \tag{5}$$

Hierbei sind X und Y zwei Featurevektoren mit n Stellen. Die Mahalanobis-Distanz berücksichtigt die Kovarianz zwischen den Elementen der Featurevektoren (siehe z. B. [12]):

$$D_{mahal} = (X - Y)^t C^{-1} (X - Y) \tag{6}$$

X, Y sind Featurevektoren und C^{-1} ist die Kovarianzmatrix. Da die Berechnung dieser Distanz zum Recherchezeitpunkt sehr aufwendig ist, wird in [47] vorgeschlagen, sie zu zerlegen und anstelle der Featurevektoren die einzelnen Terme abzuspeichern:

$$D_{mahal} = X^t C^{-1} X + Y^t C^{-1} Y - 2 X^t C^{-1} Y \tag{7}$$

Die metrischen Axiome sind nicht die einzige Möglichkeit zur Definition von Ähnlichkeitsmaßen. Sie werden jedoch unter anderem in der Psychologie sehr häufig angewendet und sollen auch für uns gelten.

2.2.2 CBIR-Techniken

Im folgenden werden die verschiedenen Techniken kurz dargestellt, die für CBIR verwendet werden. Es wurde versucht, neben etablierten Methoden auch aktuelle erfolgversprechende Ansätze darzustellen.

2.2.2.1 Feature- und Distanzfunktionen

2.2.2.1.1 Arten von Features

Features werden eingeteilt in allgemeine (general purpose; z. B. Farbhistogramm, Texturen, etc.) und anwendungsspezifische (domain specific) ([48], [16]). Beispiele für letztere sind Features zur Gesichtserkennung (human face recognition), Erkennung von Fingerabdrücken, aber auch die Features, die im Rahmen dieser Arbeit speziell für Wappen entwickelt wurden. Außerdem lassen sich Features danach unterscheiden, ob sie sich auf ein Bild als Ganzes (scene; z. B. Farbhistogramm) oder auf Bildteile (objects) beziehen ([16]).

2.2.2.1.2 Farbfeatures

Farben sind wohl die meistverwendeten Features im CBIR. Zu ihren Vorteilen zählen unter anderem die Unabhängigkeit von der Bildgröße und Perspektive. Normalerweise (z. B. in [12]) wird Farbe durch das Trippel (rot, grün, blau) dargestellt. Für einige Anwendungen wie z. B. die Zerlegung von Bildern (image segmentation; [18]) ist es aber besser, ein anderes Farbmodell (color set) zu verwenden. Häufig wird das HVC-Modell verwendet, in dem eine Farbe durch hue (Farbwert), value (luminance; Intensität) und chroma (saturation; das Ausmaß in dem eine Farbe vorhanden ist) dargestellt wird. Die Umrechnung von RGB in HVC kann durch folgenden Algorithmus durchgeführt werden (aus [19], [47]):

```
max = max(R,G,B)
min = min(R,G,B)

R1 = (max-R) / (max-min)
G1 = (max-G) / (max-min)
B1 = (max-B) / (max-min)

V = max
C = (max-min) / max

if (R = max and G = min) H = 5 + B1
else if (R = max and G <> min) H = 1 - G1
else if (G = max and B = min) H = R1 + 1
else if (G = max and B <> min) H = 3 - B1
else if (R = max) H = 3 + G1
else H = 5 - R1

H = H*60
```

Ein intensives Gelb würde bei additiver Farbmischung im RGB-Modell ca. mit dem Vektor RGB = $(230, 230, 50)^t$ dargestellt werden (wobei die Farbwerte auf das Intervall [0, 255] normiert sind). Das entspricht im HVC-Modell dem Vektor HVC = $(60, 230, 0.72)^t$ (wobei H auf [0, 360], V auf [0,255] und C auf [0, 1] normiert ist). Die Farbe hat also eine hohe Intensität und Sättigung.

Das häufigste Farbfeature ist vermutlich ein Farbhistogramm (vgl. [16], [38], [55]) in Kombination mit einem Distanzmaß, das Ähnlichkeiten zwischen verschiedenen Farben berücksichtigt (z. B. Mahalanobis-Distanz, vgl. Abschnitt 2.2.1) Gegen ein Farb-histogramm spricht, daß es keine räumliche (spatial) Information berücksichtigt. Diese negative Eigenschaft kann dadurch gemildert werden, daß nicht nur ein globales Histogramm, sondern auch für Subregionen lokale Histogramme erzeugt werden (histogram localization; [18], [16]). Eine Alternative zum Histogramm ist die Verwendung von Farb-Momenten (z. B. Mittelwert, Varianz und Schiefe). Andere Farbfeatures sind die vorherrschende Farbe (dominant color; [35]) oder die Anzahl der Farben, etc.

2.2.2.1.3 Texturen

Die Bedeutung von Texturen als wesentlichen Bildeigenschaften sind schon lange bekannt, daher gibt es zu diesem Gebiet eine Fülle von Arbeiten sowohl allgemeiner als auch spezieller Natur. Im folgenden geben wir, da Texturfeatures für künstliche Bilder und unseren Anwendungsbereich nur eine untergeordnete Rolle spielen, nur einen kurzen Überblick über die verwendeten Methoden zur Featureextraktion und Distanzberechnung sowie einen Qualitätsvergleich der verschiedenen Methoden.

Textur-Features können in zwei Gruppen eingeteilt werden: strukturelle und statistische ([16], [38]). Erstere definieren Texturen anhand ihrer Grundstrukturen, letztere anhand der geometrischen Verteilung der Bildpunkte. Das Problem struktureller Methoden liegt darin, daß

sie nur bei wirklich gleichmäßigen Texturen gute Ergebnisse erzielen und daher selten anwendbar sind. Die bekanntesten Techniken zur statistischen Texturerkennung sind:

- Tamura features: abgeleitet aus psychologischen Studien, sah Tamura als wesentliche Textureigenschaften Grobheit (coarseness), Kontrast und Richtung (directionality) an (vgl. [16], [38]).

- Simultaneous Autoregressive (SAR), rotation-invariant SAR (RISAR), multiresolution (MRSAR) Modell: hier werden die Eigenschaften der Pixel eines Bildes anhand ihrer Nachbarn beschrieben. Das RISAR-Modell stellt eine Erweiterung zu SAR dar, die es drehungs-invariant macht. Das MRSAR-Modell schließlich kann Texturen unabhängig von ihrer Granularität beschreiben (vgl. [16], [38]).

- Wold Features: in [29] wird dargestellt, wie Texturen mithilfe der Zerlegung des als Farbrelief interpretierten Bildmaterials (wold decomposition) dargestellt werden können. Beschrieben werden hier: Periode (periodicity), Richtung und Willkürlichkeit (randomness; siehe auch [37]).

In [16], [29] werden die oben angegebenen Texturmethoden nach ihrer Leistungsfähigkeit verglichen. Dazu wird die sogenannte Brodatz-Datenbank von 112 Textur-Bildern verwendet, die unter anderem in [36] beschrieben ist. Der Vergleich ergibt eine eindeutige Reihung nach Leistungsfähigkeit: wold Zerlegung, MRSAR, Tamura. Da aber auch in diesen Tests die Definition von Ähnlichkeit subjektiv ist, bedeutet das nicht, daß die wold Zerlegung in jedem Fall die beste Alternative darstellt.

2.2.2.1.4 Form / Gestalt (shape)

Form-Features beziehen sich üblicherweise auf Objekte einer Szene und geben allgemeine geometrische Eigenschaften derselben an (Größe, Lage, etc.) Sie können danach unterschieden werden, ob sie sich auf den Rand (boundary-based) oder ein Objekt als Ganzes (region-based) beziehen ([38]). Ein Beispiel für ersteres sind Fourier Deskriptoren (vgl. [38], [55]); für letzteres Moment-Invarianten (invariant gegen Verschiebung, Drehung und Skalierung).

Bei Verwendung der Fast Fourier Transformation (FFT) werden normalerweise die ersten Koeffizienten herangezogen, um Eigenschaften von Objekten zu beschreiben. Die Vorteile der FFT sind Schnelligkeit und Aussagekräftigkeit.

Von den Moment-Invarianten sind vor allem die folgenden besonders interessant:

- Zentrale Momente:

$$\mu_{p,q} = \sum_{(x,y)\in R} (x - x_c)^p (y - y_c)^q \tag{8}$$

  Hierbei sind p und q natürliche Zahlen und x_c, y_c der Massenschwerpunkt des Objektes R. Die zentralen Momente lassen sich durch folgende Umformung auch skalierungsinvariant machen:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, wobei\ \gamma = \frac{p+q+2}{2} \tag{9}$$

  Aufbauend auf den zentralen Momenten entwickelte Hu (vgl. [16]) ein System von 7 Momenten, das unter anderem in [55] zur Form-Beschreibung verwendet wird.

- Zirkularität (das Verhältnis von Fläche zu Umfang eines Objektes):

$$\gamma_i = \frac{4\pi S_i}{P_i^2} \tag{10}$$

  Hierbei ist S_i die Größe des Objektes in Pixel und P_i die Anzahl der Pixel am Rand. Dieser Wert bewegt sich zwischen 0 (Linie) und 1 (Kreis).

- Hauptachsenorientierung (major axis orientation):

$$\Theta = \frac{1}{2}\arctan\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \tag{11}$$

Neuere Ansätze im Bereich Form-Features sind die Verwendung von Eigenvektoren und die Kombination von inhalts- und randorientierten Methoden ([38], [37]).

2.2.2.1.5 Layout (sketch)

Das Suchen von Bildern nach einer Skizze basiert im wesentlichen auf der Extraktion von Kanten (edges) und geeigneten Methoden zu ihrem Vergleich. Techniken zur Kantenextraktion (Vektorisierung) beruhen normalerweise auf folgenden Ablaufmodell ([16]):

1. Normalisierung des Bildes auf eine bestimmte Größe und Farbanzahl

2. Berechnung des Gradienten (Richtung, Stärke) für jedes Pixel (z. B. durch den Sobel-Operator (vgl. [49])

3. Durchführen eines Schwellwert-Verfahrens zum Herausfinden der globalen/lokalen Kanten-Kandidaten

4. Nachbearbeiten des Bildes durch Verdünnen (thinning), Verknüpfen von Kanten (merging), etc.

Danach können die Kanten aus dem Bild extrahiert und in Vektorform gebracht werden, um für Objekterkennung und/oder Ähnlichkeitsvergleich zur Verfügung zu stehen (siehe auch [30]). Neuere Ansätze zum Retrieval nach Skizzen verwenden auch Hidden Markov Models (HMM) für elastisches Auffinden von Formen ([34]).

2.2.2.1.6 Signaturen

Immer wieder wird im Rahmen des CBIR versucht, die Komplexität eines zweidimensionalen Bildes durch die Transformation in eine eindimensionale Kurve (der sogenannten Signatur) zu reduzieren. In [52] geschieht das durch die sogenannte Radon Transformation. Die Autoren wollen dadurch sowohl geometrische als auch statistische Information in die Signatur integrieren. Außerdem hat die Radon Transformation folgende Eigenschaften: sie ist unter anderem drehungs-invariant, nicht anfällig für Rauschen und invariant gegen Intensität und Kontrast, aber nicht skalierungs-invariant.

In [23] wird eine wavelet Zerlegung vorgeschlagen, der unter anderem folgende Vorteile angerechnet werden: genaue Beschreibung durch wenige Koeffizienten, skalierungs-invariant, schnell und einfach zu berechnen. In [21] schließlich wird eine Signatur dadurch generiert, daß zuerst "normale" Features (Farben, Texturen etc.) berechnet werden und mithilfe dieser Werte dann eine Wahrscheinlichkeitsdichte (Gauß-Kurve) erzeugt wird.

Das Messen der Distanz zweier Objekte erfolgt bei Signatur-Features normalerweise durch Integration des Zwischenraumes zweier Signaturen (S_x, S_y):

$$d(x, y) = \int (S_x(i) - S_y(i))\, di \tag{12}$$

Abschließend ist kritisch anzumerken, daß Signaturen oft nur eine unstrukturierte Datenmatrix um eine Dimension auf eine wiederum unstrukturierte Datenserie reduzieren, ohne wesentliche Bildeigenschaften zu extrahieren.

2.2.2.1.7 Merging von Features

Wenn im Rahmen eines Suchvorganges mehrere Features verwendet werden, werden unabhängige Anfragen für alle Features durchgeführt und die Ergebnisse durch lineare Kombination der gewichteten Einzeldistanzwerte zu einem Gesamt-Distanzwert verschmolzen (Positionswert). Diesen Prozeß nennt man Merging ([46]):

$$Positionswert_{Objekt} = \sum_{i=1}^{F} w_i d_i \quad (13)$$

Hierbei ist F die Anzahl der verwendeten Features, w_i ist das Gewicht und d_i der Distanzwert vom Suchbild (query example beziehungsweise search image) zum verglichenen Bild (candidate image) für Feature i. Diese Methode setzt voraus, daß alle Distanzfunktionen auf denselben Wertebereich standardisiert sind. Lineares Merging verursacht zwei wesentliche Probleme:

1. Die Autoren von [54] weisen darauf hin, daß manche Feature-Klassen nicht in linearer Beziehung zueinander stehen und die Kombination daher nicht in linearer Form erfolgen kann. Sie schlagen statt linearem Merging die Verwendung eines neuronalen Netzes vor, das in Abschnitt 2.2.2.2 beschrieben wird.

2. Üblicherweise muß der Benutzer die Gewichte selbst bestimmen. In manchen Systemen (z. B. [12]) sind die Gewichte auf bestimmte Werte fixiert, in den meisten aber wird nach der Präferenz des Benutzers gefragt. Die Autoren von [38] argumentieren, daß das Festlegen der Gewichte eine zu große Belastung für den Benutzer darstellt, da er dafür ein umfassendes Wissen über die Features besitzen sollte, was normalerweise nicht angenommen werden kann.

Die Gewichte dienen dazu, die Reihung der Ergebnismenge zu optimieren. In den meisten Systemen (z. B. [12]) wird die absolute Größe der Ergebnismenge beim Formulieren der Suche mit angegeben. Es ist dann die Aufgabe der Gewichte, die ähnlichsten Bilder zuerst (und damit in der Ergebnismenge) zu platzieren. Da die Gewichte diese Aufgabe kaum erfüllen können, wird in dieser Arbeit das Konzept der Suchmodelle und Schwellwerte entwickelt.

2.2.2.2 Neuronale Netzwerke

Im folgenden werden kurz einige Anwendungen von neuronalen Netzen (besonders Self-organizing Maps (SOM; [24])) im Bereich CBIR dargestellt: ikonische Indizes durch SOM, Merging, etc.

Unter einem ikonischen Index (iconic index; [16], [1]) versteht man einen Menge von Bildgruppen, wobei eine Gruppe (Cluster) jeweils durch ein Beispielbild beschrieben wird. Ein solcher Index (bestehend aus tatsächlich vorhandenen oder virtuellen Bildern) kann z. B. mithilfe von Self-organizing Maps hergestellt werden.

Eine SOM (vgl. [24]) ist ein zweischichtiges neuronales Netz, wobei jedes Neuron der Eingabeschicht mit jedem Neuron der Ausgabeschicht verbunden ist. Es kann durch die Gewichte dieser Verbindungen vollständig beschrieben werden. Die Eingabeschicht wird als ein n-stelliger Vektor und die Ausgabeschicht als eine zweidimensionale Matrix (die

sogenannte Map) interpretiert (siehe Abbildung 3). Das Lernen erfolgt, nachdem die Gewichte zu Beginn zufällig initialisiert wurden, unüberwacht in drei Schritten ([16], [24]):

1. Finden eines Eingabevektors f

2. Finden des Knotens i in der Ausgabeschicht, dessen Referenzvektor (Mittelwert) die geringste Distanz (Distanzmaß: z. B. euklidische Distanz) zu f hat. Diesen Knoten bezeichnet man als gewinnenden Knoten (winning node).

3. Anpassen des Referenzvektors des gewinnenden Knotens und seiner Nachbarn durch die sogenannte Delta-Regel (beziehungsweise Widrow-Hoff Lernregel).

Dieser Prozeß wird fortgesetzt, bis die Anpassungen der Vektoren gegen null gehen. Es erfolgt also eine Clusterung der Eingabevektoren in einer zweidimensionalen Map. SOMs sind aufgrund ihrer einfachen Handhabung und Akzeptanz ein beliebtes Werkzeug in vielen Forschungsbereichen und es überrascht daher, daß, wie die Autoren von [27] feststellen, sie bisher kaum für CBIR herangezogen wurden.

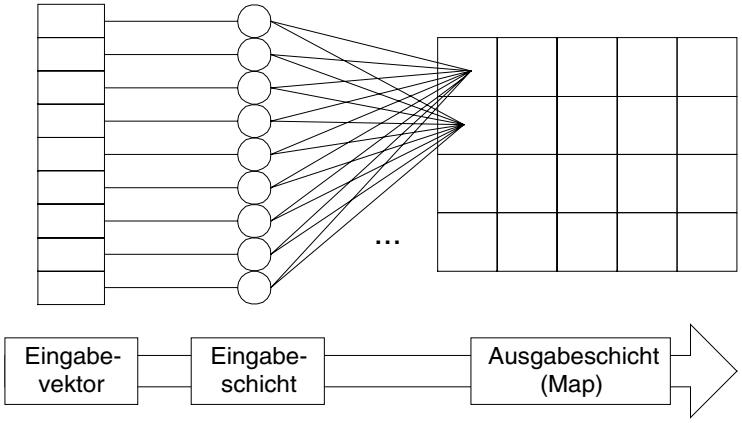

Abbildung 3: Schematische Darstellung einer SOM

Ein ikonischer Index kann so hergestellt werden, daß jeder Cluster auf unterster Ebene durch das dem Referenzvektor am nächsten liegende Bild repräsentiert wird und auf höheren Ebenen jeweils mehrere Cluster zusammengezogen werden und ein passendes Beispielbild ausgewählt wird.

Einen ähnlichen Gedanken verfolgen die Autoren von PicSOM ([28], [27]), einem CBIR-System, das auf SOMs basiert. Allerdings werden hier keine klassischen, sondern Tree Structured SOMs (TS-SOMs) verwendet, bei denen ein Cluster einer SOM auf höherer Ebene je auf eine Map auf der nächst tieferen Ebene verweist, die wiederum diesen Cluster strukturiert. In PicSOM werden nun mehrere TS-SOMs dazu verwendet, eine Bildmenge zu clustern und somit suchbar zu machen.

In [54] und [55] wird, aufbauend auf der Argumentation, daß lineares Merging (vgl. Abschnitt 2.2.2.1.7) bei Featureklassen mit nicht-linearer Relation keine geeignete Methode darstellt, ein neuronales Netz auf Perzeptron-Basis vorgestellt, das die Distanzwerte von Suchen nach mehreren Features passend verknüpft. Dabei handelt es sich um Netze mit drei (Eingabe, verdeckt, Ausgabe) beziehungsweise vier Schichten (Eingabe, Vergleich, verdeckt, Ausgabe), die durch überwachtes Lernen (back propagation Algorithmus) trainiert werden.

Schließlich sei nochmals auf [54] verwiesen, wo Learning Vektor Quantifizierung (LVQ; [25]) verwendet wird, um anhand von Benutzerrankings eine Bildmenge für die iterative Verfeinerung der Suche (vgl. Abschnitt 2.2.2.5) zu clustern. Die verschiedenen LVQ-Algorithmen bilden anhand von Eingabevektoren, die in Klassen angeordnet sind, Gruppen von sogenannten Codebook-Vektoren, die diese Klassen beschreiben.

2.2.2.3 Indizierung

Das Gebiet des Indizierens hochdimensionaler Featurevektoren ([11], [16]) soll im folgenden nur gestreift werden, da der Fokus dieser Arbeit auf den beiden anderen Kernbereichen des CBIR liegt (Extraktion von Bildinhalten, Entwurf von Suchsystemen). Es sei aber darauf hingewiesen, daß die Entwicklung geeigneter Methoden zur Indizierung der hochdimensionalen Featurevektoren ein wesentliches Kriterium für schnelle Zugriffssysteme ist ([26]).

Zur Indizierung der Featurevektoren bieten sich im wesentlichen vier Gruppen von Methoden an: R*-Bäume, k-d-Bäume, Quadtrees und Gridfiles ([11], [16], [38]). Bei allen dieser Gruppen handelt es sich um intervall-basierte Methoden, das heißt es erfolgt eine Teilraumabbildung der Indexeinträge auf Datenblöcke. Hash-basierte Methoden, bei denen eine Punktabbildung erfolgt, spielen für die Speicherung von hochdimensionalen Daten keine Rolle. R*-Bäume und k-d-Bäume basieren auf dem binären Baum (B^+-Baum).

Beim R*-Baum werden Gruppen mehrdimensionaler Objekte durch ein Rechteck mit minimalem Umfang (minimum bounding rectangle; MBR) repräsentiert. Verzeichnisknoten enthalten Einträge der Form (R, ptr), wobei ptr ein Zeiger auf einen Knoten auf einen Kind-Knoten und R die MBR ist, die alle MBRs im Kind-Knoten umspannt. Blattknoten sind Einträge der Form (obj-id, R), wobei obj-id ein Zeiger auf ein Objekt und R die MBR dieses Objektes ist. Die wesentliche Verbesserung des R*-Baums gegenüber dem B^+-Baum ist, daß sich Verzeichniseinträge überlappen dürfen. Bei Intervallsuchen im R*-Baum werden alle Zweige untersucht, deren MBR sich mit dem MBR der Suche überschneiden.

Das Indizierungsverfahren beim k-d-Baum ist eine Verallgemeinerung des binären Baumes für n-dimensionale Daten:

1. Jeder eingefügte Datenvektor (x_1, ..., x_n) partitioniert den Datenraum ein weiteres Mal in einer Dimension

2. Die Verzeichnisebene 1 wird nach der ersten Dimension der Datenvektoren (Element x_1), Ebene n nach der n-ten Dimension (Element x_n) nd die Ebene n+1 wieder nach der ersten Dimension organisiert.

Beim k-d-Baum werden die ersten Elemente der Datenvektoren bevorzugt.

Der Quadtree ist ein Indizierungsverfahren, mit dem ein zweidimensionaler Datenraum organisiert werden kann. Dabei findet keine Bevorzugung eines Attributes statt. Das Quadtree-Verfahren besteht darin, daß jedes eingefügte Attribut-Paar den Datenraum (oder einen Unterraum) in vier Teile teilt.

Beim Gridfile werden die n-dimensionalen Datenvektoren als Elemente in einem n-dimensionalen Datenraum aufgefaßt, ohne daß ein Attribut bevorzugt wird. Dazu wird der Datenraum in Teilräume der Größe von Massenspeichertransfereinheiten aufgeteilt. Die Begrenzungsinformation der Teilräume wird in n Listen (sogenannte Skalen) abgelegt. Ist ein Teilraum voll, wird eine neue Teilraumgrenze eingefügt und der Inhalt des Teilraums mit einem neuen Teilraum geteilt. Welche Skala zum Einfügen einer neuen Teilraumgrenze verwendet wird, wird meist zyklisch bestimmt. D. h. beim ersten Einfügen wird die Skala für die erste Dimension verwendet, beim zweiten Einfügen die Skala für die zweite Dimension, usw.

Allen diesen Indizierungsmethoden sind zwei Nachteile gemein:

- Mit steigender Dimensionalität verlieren die Methoden ihre Wirksamkeit (so steigt z. B. bei Gridfiles die Größe des Verzeichnisses für die Skalen mit steigender Dimensionalität stark an).

- Sie setzen oft die Annahme euklidischer Distanzen voraus.

Eine Methode zur Verminderung des ersten Problems ist grobe Repräsentation (coarse representation) von Features ([16], [38]). Dazu gibt es je nach Featureklasse verschiedene Methoden. Für Farbhistogramme bietet sich z. B. eine Reduktion der Einträge an, zur Feststellung von Positionsinformation z. B. eine Einteilung des Bildes in einen 3x3-Raster. Die Größe eines Objektes kann von der Anzahl der Pixel p_{Objekt} auf einen Wert o, für den folgendes gilt, verringert werden:

$$x^{o-1} \leq p_{Objekt} \leq x^{o} \tag{14}$$

Ausgehend von der Überlegung, daß trotz einer hohen Dimension der Featurevektoren die Anzahl linear unabhängiger Dimensionen der Vektoren vermutlich geringer ist, kann durch eine Transformation eine Dimensionsreduktion erreicht werden. Verwendet werden unter anderem die diskrete Fourier-Transformation und die Karhunen-Loeve-Transformation (KLT; [38], [37]) Die KLT benutzt dazu die Eigenvektoren der Kovarianzmatrix der Featurevektoren.

2.2.2.4 Qualitätsmessung

Die Bewertung von CBIR-Techniken in Multimedia-IR-Systemen erfolgt normalerweise anhand von Precision und Recall (siehe Abschnitt 2.1.1.2; vgl. [11], [16]), die für Bilder folgendermaßen definiert sind:

$$\text{Precision} = \frac{\text{zurückgegebene passende Bilder}}{\text{zurückgegebene Bilder}} \tag{15}$$

$$\text{Recall} = \frac{\text{zurückgegebene passende Bilder}}{\text{passende Bilder}} \tag{16}$$

Daraus wird klar, daß diese beiden Maßzahlen interdependent sind; d. h. eine Verbesserung der einen Kennzahl verursacht möglicherweise eine Verschlechterung der anderen. In der Literatur (z. B. [11]) wird der Recall üblicherweise höher als die Precision bewertet, da vor allem sichergestellt sein soll, daß alle passenden Bilder zurückgegeben werden (Vermeidung von falschen Zurückweisungen); Browsen ist dem Anwender eher zumutbar.

Problematisch an der Anwendung von Precision und Recall ist, daß einerseits zur Berechnung des Recall eine genaue Kenntnis der Datenbank nötig ist und andererseits diese beiden Maße in gewissem Umfang subjektiv sind. Dennoch werden andere Maße, wie z. B. das oben dargestellte Nützlichkeitsmaß, kaum angewendet. Oft erfolgt die Evaluierung von CBIR-Methoden sogar nur anhand von Suchbeispielen. Der Vergleich einzelner CBIR-Methoden ist derzeit aufgrund des Fehlens passender Benchmarks eigentlich kaum möglich; Lösungs-ansätze existieren hier vor allem im Bereich Texturen mit der Brodatz-Datenbank. Für andere Bereiche (Farbbilder, Gestalterkennung für natürliche beziehungsweise künstliche Bilder) gibt es noch keine passenden Standards.

2.2.2.5 Iterative Verfeinerung der Suche durch das Relevanzurteil des Benutzers

Die Probleme, die sich aus dem computer-zentrischen Ansatz (CCA) ergeben (semantische Lücke, Subjektivität der menschlichen Wahrnehmung), wurden in Abschnitt 2.2.1 behandelt. Ein Ansatz, um im Bereich Design von Suchsystemen die Rechercheergebnisse zu verbessern, ist, an die Stelle des CCA einen iterativen Prozeß zu setzen, der den Benutzer miteinbindet (iterative refinement durch relevance feedback). Relevance feedback ([39], [32]) bedeutet, daß der Benutzer beim Verfeinern seiner Suche anhand der Ergebnisse seiner letzten Abfrage sein Relevanzurteil mit einfließen läßt (siehe Abbildung 4). Dadurch wird versucht, die semantische Lücke (so weit wie möglich) zu schließen.

In [39] wird für iterative Refinement ein System mit mehreren Features verwendet (unter anderem Farbhistogramm, Fourierkoeffizienten, Tamura Features, etc.), wobei das Merging durch lineare Gewichtung erfolgt. Die Gewichte werden anhand der Informationen des

Benutzers, welche Bilder der letzten Ergebnismenge den gesuchten sehr beziehungsweise gar nicht entsprechen, dynamisch adaptiert.

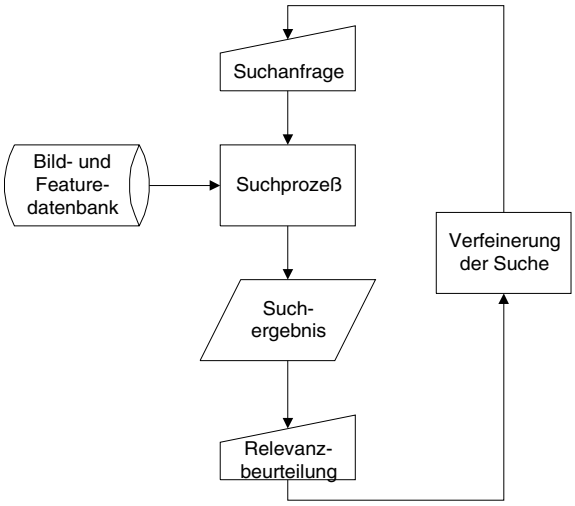

Abbildung 4: CBIR als iterativer Prozeß

In [32] erfolgt die iterative Verfeinerung der Suche anhand der Beispiele, die der Benutzer als besonders positiv bewertet. Bilder werden auch hier durch Featurevektoren repräsentiert (das heißt Bild X wird durch den Vektor $x = \{x_1, ..., x_n\}$ repräsentiert). Dabei ist die Vorgangsweise folgende:

- Gesucht wird für jede Suche die Wahrscheinlichkeitsfunktion für den Featurevektor x, mit der das Bild X relevant ist. Dabei wird davon ausgegangen, daß diese Wahrscheinlichkeitsfunktion gauß-verteilt ist.

- Weiters wird vereinfachend angenommen, daß die Elemente des Featurevektors voneinander unabhängig sind, sodaß die gesuchte Wahrscheinlichkeitsfunktion das Produkt der Wahrscheinlichkeitsfunktionen der Vektorelemente ist. Das wird im allgemeinen zwar nicht zutreffen, ist aber eine häufige Annahme in der Theorie des Information Retrieval.

- Die Parameter der Gauß-Verteilung (Mittelwert und Varianz) der Elemente der Featurevektoren werden durch iterative Verfeinerung angenähert.

2.2.2.6 Zukunftsaussichten: MPEG-7

Derzeit wird an einer Norm gearbeitet, die die Beschreibung audiovisueller Daten anhand ihrer Inhalte vereinheitlichen soll: MPEG-7 (vgl. [31], [35]). Dieser Standard wird bestehen aus sogenannten Deskriptoren (D), Beschreibungsschemata (descriptor scheme, DS) und einer Beschreibungssprache (descriptor definition language, DDL) zur Beschreibung von Deskriptoren und DS. Unter einem Deskriptor versteht man eine Repräsentation für ein

Feature, also z. B. für das Farbfeature ein Farbhistogramm. Ein Beschreibungsschema besteht aus Deskriptoren und wiederum Beschreibungsschemas und definiert die Struktur und Inhalte der Beziehungen seiner Komponenten.

Anwendungsbereiche für MPEG-7 werden unter anderem digitale Bibliotheken, multimediale Verzeichnisdienste, etc. sein. Die Fertigstellung des Standards ist für September 2001 geplant.

2.2.3 Systeme

Moderne Multimedia-IR-Systeme zeichnen sich ihren Vorgängern gegenüber dadurch aus, daß nunmehr größerer Wert auf klassische Eigenschaften eines DBMS gelegt wird: Modellierung, Speichermanagement, Indizierung spielen jetzt eine größere Rolle. Die meisten Systeme verfügen zudem über ein offenes API, das eine Weiterentwicklung (zusätzliche Features, etc.) des Systems erlaubt ([17]).

2.2.3.1 Query by Image Example - QBIC

QBIC ([12], [17], [38]) war das erste kommerzielle CBIR-System und hatte damit großen Einfluß auf spätere Entwicklungen. Ziel von QBIC ist es, als Informationsfilter zu dienen und dem Benutzer als Ergebnis einer Suche alle Objekte zu zeigen, die eventuell in Frage kommen könnten, das heißt der Fokus liegt auf dem Recall. Es unterstützt folgende Arten von Abfragen: Suche nach Beispielen und Suche nach Skizzen. Als Features werden verschiedene Arten von Farbhistogrammen sowie Tamura Features für Texturen und einige Form-Features (Zirkularität, Hauptachsenorientierung, etc.) verwendet. Zur Distanzmessung wird mehrheitlich (Texturen, Formen) eine gewichtete euklidische Distanz herangezogen. Zum Farbhistogramm-Vergleich wird eine Mahalanobis-Distanz mit einer Kovarianzmatrix der Farbähnlichkeiten verwendet.

Im Suchprozeß erfolgt keine automatische Verfeinerung und der Benutzer hat keine Möglichkeit, die Gewichte beim linearen Merging zu beeinflussen. Zur Komplexitätsreduktion wurde erstmals die Karhunen-Loeve-Transformation benutzt und als Indexstrukturen erstmals R*-Bäume, wodurch QBIC das erste System wurde, das auch Wert auf effiziente Zugriffsmechanismen legte.

2.2.3.2 Photobook

Photobook ([37], [38]) wurde am MIT Media Lab entwickelt und stellt eine Menge interaktiver Werkzeuge zum Browsen und Durchsuchen von Bilddatenbanken bereit. Als Features werden Shape-Features, Texturfeatures (wold features) und Features zur Gesichtserkennung verwendet. Photobook nutzt exzessiv die Karhunen-Loeve-Transformation zur Dimensionsreduktion und in der neueren Version FourEyes Methoden zur iterativen Verfeinerung der Suche.

2.2.3.3 VisualSEEk

VisualSEEk (sowie die Web-Variante WebSEEk; [47], [38]) wurde entwickelt an der Columbia University; wesentlichste Eigenschaft ist die Kombination von visuellen Features und Positionsinformation, das heißt es wird zu jedem Feature geometrische Information miterfaßt. In Abfragen können diese Eigenschaften miteinander kombiniert werden. Verwendete Features sind Farbhistogramme und Textur-Features (wavelet Zerlegung). Zur Beschleunigung von Abfragen werden binäre Bäume zur Indizierung der Daten verwendet.

2.2.3.4 Virage

Das kommerzielle Virage ([2], [38]) erlaubt Suchen nach Farben, Farb-Layout, Texturen und Strukturen (Objektrand-Information) sowie Kombinationen dieser Basis-Features. Zur Kombination der Teilabfragen wird lineares Merging verwendet, wobei der Benutzer die Gewichte selbst bestimmen darf. Außerdem verfügt Virage über ein API, sodaß weitere Features hinzugefügt werden können.

2.2.3.5 MARS

MARS ist ein Entwicklungsprojekt an der Universität von Illinois in Urbana ([38]), wobei neben der Weiterentwicklung der Methoden des CBIR auch die Integration von Information Retrieval in Datenbankmanagement-Systeme, die Nutzung der Indizierungsstruktur beim Retrieval und die Benutzerfreundlichkeit der Schnittstelle von CBIR-Systemen verbessert werden sollen. Ziel von MARS ist es, durch die Verwendung mehrerer Features und von iterativer Verfeinerung der Suche CBIR-Systeme allgemeiner anwendbar zu machen.

2.2.3.6 Andere allgemeine CBIR-Systeme

Eines der ältesten CBIR-Systeme ist Art Museum. Als einziges Feature wird hier der Rand von Objekten verwendet. Das System CANDID (Comparison Algorithm for Navigating Digital Image Databases; [23]) schließlich generiert für jedes Bild in der Datenbank eine globale Signatur, anhand derer Ähnlichkeiten festgestellt werden.

2.2.4 Zusammenfassung

Dieser Abschnitt gibt einen kurzen Überblick über das sich gegenwärtig schnell entwickelnde Gebiet des CBIR. Es werden erprobte Methoden ebenso wie zukunftsweisende Ansätze dargestellt sowie derzeit vorhandene Systeme und Einsatzgebiete skizziert. Trotz einiger Mängel wird sich CBIR zweifellos in vielen Bereichen als Ergänzung zur traditionellen Stichwortsuche etablieren.

2.3 Einführung in die Heraldik

Der Ursprung der Heraldik liegt im Mittelalter, als Könige und Feldherren ihre Truppen zur Identifikation mit Symbolen kennzeichneten, die von Rittern vorzugsweise auf dem Schild getragen wurden. Ihren Namen verdankt die Heraldik den Herolden, die, ursprünglich zur Nachrichtenübermittlung eingesetzt, durch ihre weiten Reisen im Laufe der Zeit zu Kennern der vielen unterschiedlichen Symbole wurden. Das ging so weit, daß bisweilen Herolde nötig waren, um nach einer geschlagenen Schlacht festzustellen, wer nun eigentlich gewonnen hatte (wie z. B. von Shakespeare in [45] dargestellt). Herolde begannen damit, Wappen-Beschreibungen zu sammeln und genaue Regeln festzulegen, wie ein Wappen aufgebaut sein muß, welche Elemente es enthalten darf, etc. und gründeten sogenannte Colleges of Heralds zur Registrierung neuer Wappen. Im folgenden werden die wichtigsten Regeln der Heraldik kurz dargestellt.

Der wesentliche Teil eines Wappens, dessen Aussehen durch Regeln bestimmt ist, ist das Schild. Dieses kann eventuell noch mit Helm, Helmdecke und Helmzier versehen sein und muß eine der in Abbildung 5 dargestellten vier Formen haben (vgl. [20]).

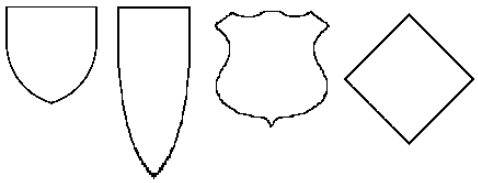

Abbildung 5: Erlaubte Schildformen für Wappen

Im Schild dürfen nur bestimmte Farben, die sogenannten Tinkturen, verwendet werden. Es wird unterschieden zwischen Metallen: Gold (gelb) und Silber (weiß), Farben: rot, blau, schwarz, grün und purpurn sowie Pelzen (dargestellt als weiß/gelb/schwarzes Muster). Ist es z. B. für einen Siegeldruck nötig, Tinkturen in schwarz/weiß-Darstellung zu bringen, müssen die in Abbildung 6 gezeigten Schraffuren verwendet werden (vgl. [10]).

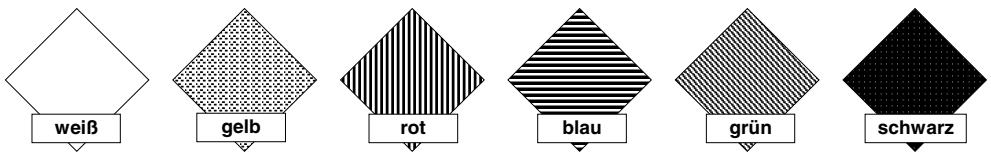

Abbildung 6: Schraffuren für Farben in Siegeln

Des weiteren gelten für die Verwendung der Tinkturen folgende Regeln:

- Metalle dürfen nicht auf Metallen und Farben nicht auf Farben plaziert werden.
- Metalle müssen immer auf Farben plaziert werden und umgekehrt.
- Pelze können entweder den Platz eines Metalls oder einer Farbe einnehmen.

Ein Schild besteht normalerweise aus mehreren Schichten: Hintergrund (field), Heroldsstücken (ordinaries) und Objekten (charges). In der Hintergrund-Schicht wird die Unterteilung des Schildes in Regionen festgelegt (heraldische Schnitte beziehungsweise field divisions). Erlaubt sind unter anderem die in Abbildung 7 dargestellten Schnitte.

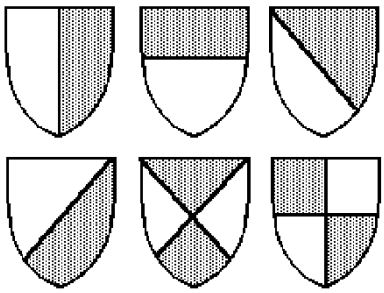

Abbildung 7: Heraldische Schnitte

Es ist zwar nicht erlaubt, bestimmte Tinkturen auf andere zu legen, aber durchaus, sie in unterschiedlichen Feldern nebeneinander anzubringen. Die Form der Unterteilungslinien (division lines) ist ebenfalls auf wenige Typen beschränkt (gerade, gezackt, sinusförmig, etc.).

Die Heroldsstücke sind eigentlich eine Untergruppe der heraldischen Objekte, die sehr häufig verwendet werden: Kreuzformen, Balken, etc. Es wird unterschieden zwischen großen (major ordinaries) und kleinen Heroldsstücken (sub-ordinaries), die weniger häufig und deshalb nicht exakt definiert sind (vgl. Abbildung 8 und 9).

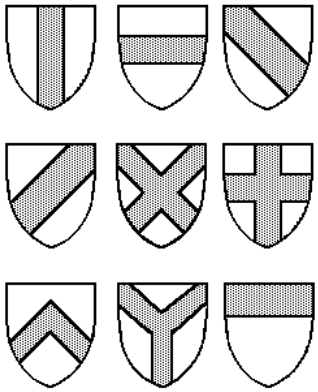

Abbildung 8: Große Heroldsstücke

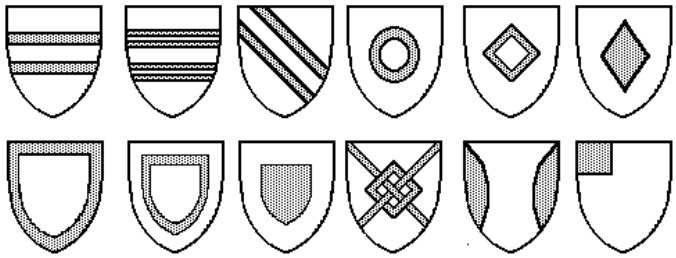

Abbildung 9: Kleine Heroldsstücke

Über dem Hintergrund und den Heroldsstücken können nun verschiedene Arten von Objekten angebracht werden. Es wird unterschieden zwischen: Menschen, Tieren (Löwen, Vögel, Fische, etc.), mythischen Kreaturen (Einhörner, Drachen, etc.), Blumen (vor allem Rosen und Lilien) sowie, bei bürgerlichen Wappen, Geräten und Marken (Buchstaben, etc.).

Durch die genaue Festlegung, aus welchen Elementen ein Wappen bestehen darf und wie die einzelnen Teile miteinander in Beziehung stehen dürfen, ist es möglich, den Inhalt eines Wappens durch eine kurze Beschreibung so wiederzugeben, daß es ohne das Bild zu kennen sehr genau nachgebildet werden kann. Diese textuellen Beschreibungen nennt man Blazon. Angegeben wird für jedes Objekt, in welchem Feld eines heraldischen Schnittes es liegt, aus welchen Tinkturen es besteht und was es darstellt. Die Blazonierung entwickelte sich wie die Heraldik selbst im Mittelalter, wo es zu den Aufgaben der Herolde gehörte, bei Turnieren die teilnehmenden fremden Ritter dem weit entfernt sitzenden Publikum anhand ihrer Wappen vorzustellen.

Im Laufe der Jahrhunderte wurden Wappen neben ihrem ursprünglichen Zweck zu einem Statussymbol: aristokratische Familien trugen Wappen, Städte, Regionen und Länder kreierten eigene Wappen und nach den bürgerlichen Revolutionen im 18. und 19. Jahrhundert begannen auch Bürgerfamilien (vor allem in Deutschland) Wappen zu tragen. Die damit verbundene enorme Zunahme an Wappenbildern machte es notwendig, sie systematisch zu ordnen, um sie suchbar zu machen. Heute gibt es im deutschen Sprachraum unterschiedliche Ordnungssysteme für aristokratische, bürgerliche und hoheitliche Wappen. In [33] wird für bürgerliche Wappen folgendes Ordnungssystem verwendet:

- Heroldsbilder (zusammengesetzt aus geometrischen Symbolen)
- Kosmos (Mond, Sonne, Sterne, etc.)
- Lebewesen (Menschen, Tiere, mythische Kreaturen)
- Pflanzen (Blumen, Kränze, etc.)
- Geräte (Waffen, Werkzeugen, etc.)
- Buchstaben und Marken
- Mehrfeldrige Schilde (geteilt, gespalten, schräggeteilt, Balken, etc.)

Die Autoren von [33] weisen darauf hin, wie unzureichend ein solches Ordnungssystem für die tägliche Arbeit eines Heraldikers ist, die es in großen Bibliotheken, z. B. der Österreichischen Nationalbibliothek, durchaus noch gibt, stellen gleichzeitig aber bedauernd fest, daß derzeit (1995) ein computerbasiertes System zur Wappensuche noch als Utopie anzusehen ist. Ein solches System müßte idealerweise zu einer Vorlage (Suchbild,

Siegelabdruck) aus der Menge der bekannten Wappen jene (und die damit verknüpften Informationen) finden, die ihm am ähnlichsten sind. Dieses Problem ist heute (2000) durch die auf CBIR basierende Wappen-Testdatenbank weitgehend gelöst.

3 Suchmodellbasiertes CBIR für Wappen

In diesem Kapitel wird gezeigt, wie CBIR für Wappen realisiert wurde. Dazu werden die verwendeten Features beschrieben, das Konzept der suchmodellbasierten Recherche erklärt und statistische Untersuchungen der Datenbasis sowie der Features durchgeführt, um die gemachten Qualitätsaussagen bewerten zu können.

3.1 Features

Dieser Abschnitt beschreibt, welche Features zur Suche nach Wappenbildern entworfen und implementiert wurden, wobei in der Gliederung zwischen allgemeinen und anwendungsspezifischen Features unterschieden wird. Am Ende befindet sich eine kurze Beschreibung der Arbeits- und Testumgebung.

3.1.1 Allgemeine Features

3.1.1.1 Farbfeatures

Als allgemeine Features wurden entworfen und implementiert: ein Farbhistogramm für Wappenfarben, Features, die die Anzahl der Farben beziehungsweise Farbabstufungen in einem Bild zählen, ein Feature, das es erlaubt, Farbbedingungen zu formulieren, ein Form-Feature sowie einige Features zum Bestimmen von Symmetrien.

Das Farbhistogramm (Klassenname: QbWappFarbFeatureClass) realisiert ein Farbhistogramm, das speziell auf Wappenfarben (Tinkturen, vgl. Abschnitt 2.3) ausgerichtet ist. Es umfaßt nur sechs Farben: schwarz, weiß, rot, grün, blau und gelb. Die siebte Tinktur, purpur, wurde, da sie in zivilen Wappen (in der Testdatenbank vorwiegend verwendet; vgl. Abschnitt 3.1.3) nur sehr selten vorkommt, weggelassen. Die einzelnen Histogrammeinträge wurden aus den, auf Farbwerte abgebildeten, in QBIC im RGB-Schema angegebenen Pixelwerten aufsummiert. Dabei wurden die einzelnen Farben durch folgende rot/grün/blau - Intervalle definiert:

Farbe	Rot-Intervall	Grün-Intervall	Blau-Intervall
Schwarz	[0,80]	[0,80]	[0,80]
Weiß	[220,255]	[220,255]	[220,255]
Rot	[200,255]	[0,100]	[0,100]
Gelb	[200,255]	[200,255]	[0,110]
Blau	[0,130]	[0,255]	[150,255]
Grün	[0,110]	[100,255]	[0,110]

Diese Bereiche, die durch Auswertung der RGB-Werte für ein Farb-Testbild festgelegt wurden, umfassen alle typischerweise in Wappen verwendeten Farben. Die Vorgangsweise bei der Featureberechnung ist folgendermaßen:

1. Berechnung eines sechstelligen Histogramms aus dem in Form einer Bitmap an das Feature-Objekt übergebenen Bild.

2. Standardisierung des Featurevektors. Jedes Vektorelement wird durch folgende Umformung von der Bildgröße unabhängig gemacht:

$$FH_e = a\frac{FH_e}{mx.my} \tag{17}$$

Hierbei sind FH_e die Elemente des Featurevektors, mx und my die Ausdehnungen des Bildes und a ein beliebiger Parameter zur Abbildung der Histogrammeinträge auf einen bestimmten Wertebereich. In der Testumgebung wurde der Parameter a auf den Wert 30000 gesetzt, um die Elemente des Featurevektors als Integer-Werte abspeichern zu können. Abbildung 10 zeigt schematisch den Prozeß der Featureextraktion.

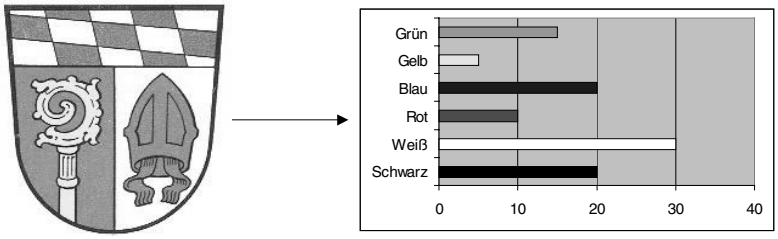

Abbildung 10: Ableitung eines Farbhistogramms

Statistische Auswertungen der Testdatenbank ergaben folgende durchschnittlichen Farbanteile (wobei nur das Wappenschild ohne Hintergrund betrachtet wurde):

Farbe	Fläche (Prozent)
Rot	46,80
Schwarz	21,30
Blau	12,19
Gelb	9,42
Grün	8,50
Weiß	1,78

Als Distanzmaß wurde im Gegensatz zum QBIC-Standard-Farbhistogramm, wo eine Mahalanobis-Distanzfunktion gewählt wurde, die euklidische Distanz gewählt (vgl. Abschnitt 2.2.2.1.3), da die Wappenfarben nur nach bestimmten, in der Heraldik festgelegten, Regeln

kombiniert werden dürfen und folglich Farbähnlichkeiten, auch weil nur sehr wenige Farben betrachtet werden, keine Rolle spielen:

$$D\left(FH^a, FH^b\right) = \frac{\sum_{i=0}^{5}\left(FH_i^a - FH_i^b\right)^2}{6} \tag{18}$$

Hierbei sind FH die Featurevektoren. Die Ordnung dieser Distanzfunktion O(n) ist quadratisch zur Anzahl der betrachteten Bilder und konstant für die Dimension der Vektoren. Das Farbhistogramm hat sich als besonders geeignet für eine schnelle Vorauswahl von passenden Bildern erwiesen. Es spielt folglich bei Clusterungen eine entscheidende Rollte (vgl. Abschnitt 3.3.3). Abbildung 11 zeigt ein typisches Suchergebnis für das Farbhistogramm. Das erste Bild ist dabei (und in allen folgenden Suchbeispielen) das gesuchte Bild.

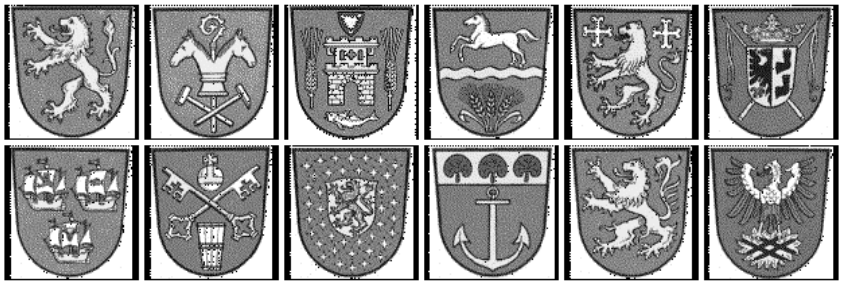

Abbildung 11: Suchbeispiel für das Farbhistogramm-Feature

Eine besondere Variante des Farbhistogramms, das bedingte Farbhistogramm (Klassenname: QbFarbBedFeatureClass), bildet denselben Featurevektor wie des Farbhistogramm, verwendet aber als Distanzmaß nicht die euklidische Distanz, sondern erlaubt dem Benutzer, Farbbedingungen zu formulieren und so alle jene Bilder zu selektieren, die diese Bedingung erfüllen. Bedingungen haben dieselbe Syntax wie im test-Kommando von Unix, da zur Auswertung der Bedingungen der Quelltext von GNU test verwendet wurde. Zusätzlich können die Farbwerte der sechs Wappenfarben als Parameter angegeben werden. Eine Bedingung könnte beispielsweise so aussehen:

$$R - gt\,10000 - a\,Y - lt\,100 \xrightarrow{?} select \tag{19}$$

Hier werden alle Bilder selektiert, die mehr als 10000 rote und weniger als 100 gelbe Pixel haben. Dabei ist die Gesamtanzahl der Pixel auf 30000 standardisiert. Für die verschiedenen Farben gelten folgende Platzhalter: "N": schwarz, "W": weiß, "R": rot, "Y": gelb, "B": blau und "G": grün. Erlaubt sind unter anderem folgende Operatoren: "-gt": größer als, "-lt": kleiner als, "-eq": gleich, "-a": und-Verknüpfung sowie "-o": oder-Verknüpfung (vgl. Unix-Dokumentation zum Kommando "test"). Die Distanzfunktion des bedingten Farbhistogramms ersetzt in der Bedingung alle Farben-Platzhalter durch die tatsächlichen Histogrammeinträge des jeweils betrachteten Bildes, prüft anhand der test-Funktion, ob sie zu "wahr" evaluiert und retourniert

"ähnlich", wenn das der Fall ist beziehungsweise "unähnlich", wenn nicht. Die Auswahl eines Suchbildes ist nicht erforderlich.

Zu den allgemeinen Farbfeatures gehören schließlich noch die Features zur Bestimmung der Anzahl von Farbabstufungen (Klassenname: QbWappNKFeatureClass) und der Zahl der Wappenfarben (Klassenname: QbWappFAnzFeatureClass). Ersteres stellt für ein beliebiges Bild die Anzahl der Farbübergänge, letzteres die Anzahl der verwendeten Wappenfarben fest. Der Klassenname QbWappNKFeatureClass rührt daher, daß dieses Feature sehr gut geeignet ist, um künstliche Bilder (wenige Farbabstufungen) von natürlichen (viele Farbabstufungen) zu unterscheiden. Die Distanzfunktionen beider Features berechnen die absolute Distanz zweier (einstelliger) Featurevektoren:

$$D_{x,y} = \frac{\left| f_x - f_y \right|}{m} \tag{20}$$

Dabei sind f_x und f_y die Featurevektoren (Zahl der Farben beziehungsweise Farbabstufungen) und m die maximal mögliche Distanz (sechs Farben beziehungsweise (bei 256 Farben) 64 Farbabstufungen). Da dies eine sehr schnelle Distanzfunktion ist, eignen sich diese Features hervorragend für eine schnelle Vorauswahl von Bildern.

3.1.1.2 Objekt-Layout-Feature

Ein weiteres allgemeines, das heißt nicht nur für Wappen verwendbares, Feature ist das Feature zur Bestimmung des Objekt-Layouts eines Bildes (Klassenname: QbWappObjFeatureClass), das Bilder anhand des Layouts der in ihnen vorhandenen Objekte vergleicht. Dazu werden die Bilder zuerst vektorisiert und dann anhand der erhaltenen Kanteninformationen Objektbeschreibungen erstellt. Abbildung 12 zeigt schematisch, wie ein reales Bild auf eine Beschreibung abgebildet wird:

Im einzelnen werden im Rahmen der Featureextraktion folgende Schritte durchlaufen:

1. Normalisierung: jedes Bild wird auf eine standardisierte Größe umgewandelt.

2. Vektorisierung: dieser Schritt besteht aus Kantenextraktion und Objekterkennung. Zur Kantenerkennung wurden zwei Lösungsansätze implementiert:

   2.1 Kantenextraktion aus einem Grauwertbild aufgrund der Helligkeitswerte. Dazu wird jedes Bild aus dem RGB-Schema durch folgende Transformation pixelweise in ein Grauwertbild umgewandelt:

$$GW_p = 0.3R_p + 0.59B_p + 0.11G_p \tag{21}$$

Hierbei ist GW_p der Grauwert des Pixels p und R_p, B_p und G_p sind die Rot-, Blau- und Grün-Anteile des Pixels. Diese Transformation wird unter anderem auch zur Bildung des Luminanz-Signals beim Farbfernsehen ([53]) und bei der Bildbearbeitung in Photoshop verwendet. Für das erhaltene Grauwert-Bild wird durch den Sobel-Operator für jedes Pixel der Gradient (Richtung, Stärke) berechnet. Das Sobel-Verfahren berücksichtigt jeweils nur die Nachbarn der ersten Ordnung (bei Punkten, die nicht am Rand liegen: acht), wobei die vier direkten Nachbarn ein doppeltes Gewicht haben. Auf die zweidimensionale Gradienten-Map wird dann ein lokales Schwellwert-Verfahren mit globalem Einfluß angewendet, wobei sich ein innerer Bereich von 16x16 Bildpunkten bei einem äußeren von 64x64 Bildpunkten als am besten geeignet erwiesen hat. Alle Bildpunkte, deren Gradient den Schwellwert überschreiten, werden in der Kanten-Map mit "1" bewertet, alle anderen mit "0". Das Ergebnis ist eine binäre Matrix mit den gefundenen Kanten.

2.2 Das zweite Verfahren zur Kantenextraktion beruht auf der Einsicht, daß in Wappen ohnehin nur wenige Farben mit stufenweisen Übergängen verwendet werden. Vereinfachend kommt hinzu, daß jedes Objekt in einem Wappen nur eine Farbe hat. Deshalb wird bei dieser Methode nacheinander für jede Farbe eine Kanten-Matrix erzeugt (wobei ein Pixel immer dann mit "1" bewertet wurde, wenn es die gewünschte Farbe hatte) und auf diese das Verfahren zur Objekterkennung angewendet. Die Ergebnisse für die einzelnen Farben werden dann zu einer Gesamt-Beschreibung verschmolzen. Für Wappen erwies sich dieses Verfahren als wirkungsvoller, weshalb es unter anderem auch im Objekt-Layout-Feature verwendet wurde.

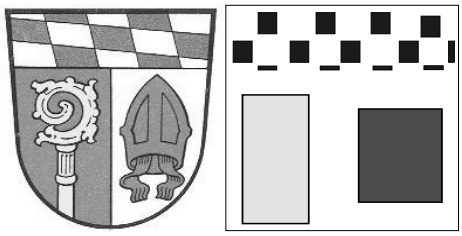

Abbildung 12: Beispiel für das Objekt-Layout-Feaure

Mit den Daten der Kanten-Matrix wird die Objekterkennung durchgeführt. Dazu wurde eine Reihe von Prozeduren erstellt, die aus Folgen von Bildpunkten Kantenbeschreibungen der Form (Startpunkt, Richtung, Länge) erzeugen und anschließend Objekte als zusammenhängende Folgen von Kanten erkennen. Dazu wird folgendermaßen vorgegangen: die Kantenmatrix wird von links, oben nach rechts, unten durchsucht. Immer, wenn ein gesetztes Pixel gefunden wird, startet eine rekursive Prozedur, die alle benachbarten gesetzten Pixel sucht. Für diese Pixel wird ebenfalls wieder rekursiv weitergesucht. Daraus wird die Kanten- und Objektinformation aufgebaut. Nachdem ein Pixel bearbeitet wurde, wird es in der Kantenmatrix auf "0" gesetzt, damit es

in der Folge nicht mehr gefunden wird. Für jedes Objekt wurden unter anderem folgende Daten aufgezeichnet:

- Größe in Bildpunkten sowie Größe des Umrisses in Pixel.
- Maximale Ausdehnung des Objektes, Mittelpunkt.
- Anzahl und Histogramm der verwendeten Wappenfarben.
- Mittelwert und Varianz der RGB-Farbwerte.
- Anzahl der Kanten im Objektrand, der Extrempunkte und der Löcher im Objekt.
- Rand als Liste von Kanten (inklusive aller Löcher = "innerer Rand") sowie der Innenwinkel zwischen zwei Kanten.

Unter einem Extrempunkt werden dabei Pixel verstanden, die am Rand liegen und in eine oder mehrere Richtungen (oben, unten, links, rechts) keinen weiter entfernten Nachbarn im Objekt haben. Folglich gibt es verschiedene Kategorien von Extrempunkten: solche, die nur in eine Richtung am weitesten vorstoßen (vgl. das Sechseck in Abbildung 13), solche, bei denen das für zwei Richtungen der Fall ist (z. B. mindestens eine Spitze eines Dreiecks; vgl. Abbildung 13) sowie solche, die im Hinblick auf alle vier Richtungen ein Extrempunkt sind (Objekt, das nur aus einem Pixel besteht).

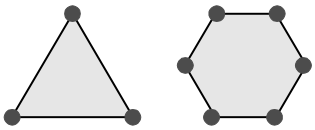

Abbildung 13: Arten von Extrempunkten

3. Auf der Basis dieser Objektdaten wurde versucht, höherwertige Objektbeschreibungen abzuleiten. Dabei wurden nur Objekte betrachtet, die zumindest aus 50 Pixeln bestanden. Ein Objekt wird durch folgende Struktur beschrieben:

   - Typ: möglich sind geometrische Grundformen: Linie, regelmäßiges konvexes Vieleck, unregelmäßiges konvexes Vieleck, Kreis, Zickzack (konkaves Vieleck).
   - Größe: angegeben wird nun nicht mehr die Anzahl der Pixel p, sondern der Wert x, für den folgende Relation gilt:

$$2^x \leq p < 2^{x+1} \tag{22}$$

   - Anzahl der Ecken (Anzahl der Kanten am Rand).
   - Verhältnis zwischen Höhe und Breite des Objektes.

- Mittelpunkt des Objektes bezogen auf die linke obere Ecke des Bildes.

- Durchschnittliche Farbe (RGB-Mittelwert).

Als Featurevektor wird eine Datenstruktur dynamischer Länge von allen Objekten, sortiert nach Typ, Größe und Eckenzahl abgelegt. Dieser Featurevektor kann als Beschreibung des Objekt-Layouts eines Bildes aufgefaßt werden.

Zum Distanzvergleich zweier Featurevektoren wurde folgender Algorithmus herangezogen:

1. Es wird die Anzahl der Objekte zweier Bilder bestimmt, die vom gleichen Typ sind, ähnliche Größe und eine ähnliche Anzahl von Ecken besitzen, sich in etwa an der selben Stelle im Bild befinden sowie von ähnlicher Farbe sind. Die Ähnlichkeitsmessung erfolgt aufgrund von Parametern, die die maximale Abweichung, bezogen auf den Typ der Variablen, angeben (derzeit sind generell 10 Prozent Abweichung erlaubt).

2. Der Distanzwert wird berechnet als:

$$D_{x,y} = 1 - \frac{a}{\min(o_x, o_y)} \tag{23}$$

   wobei a die Anzahl der ähnlichen Objekte ist und o_x beziehungsweise o_y die Anzahl der Objekte in den durch die Featurevektoren x und y repräsentierten Bilder sind. Dieses Distanzmaß, das aufgrund der Sortierung der Featurevektoren sehr effizient implementiert werden kann, ist nur unwesentlich langsamer als die zum Vergleich von Farbhistogrammen verwendete euklidische Distanz.

3.1.1.3 Symmetrie-Features

Der Bereich der allgemeinen Features wird abgeschlossen durch die Gruppe der Symmetrie-Features; sie besteht aus folgenden Features / Klassen:

Feature	Klassenname	Symmetrieachse
X-Achsen-Symmetrie	QbSymXFeatureClass	X-Achse
Y-Achsen-Symmetrie	QbSymYFeatureClass	Y-Achse
Inverse X-Achsen-Symmetrie	QbSymInvXFeatureClass	X-Achse
Inverse Y-Achsen-Symmetrie	QbSymInvYFeatureClass	Y-Achse
Diagonal ansteigende Symmetrie	QbSymDplusFeatureClass	Bilddiagonale mit Steigung 1

Feature	Klassenname	Symmetrieachse
Diagonal absteigende Symmetrie	QbSymDminusFeatureClass	Bilddiagonale mit Steigung -1
Kreuz-Symmetrie	QbSymKFeatureClass	X- und Y-Achse
Symmetrietyp	QbGetSymFeatureClass	

Diese Features messen, mit Ausnahme des Symmetrietyps, Symmetrien, indem sie ein Bild in zwei Hälften unterteilen, die zweite Hälfte spiegeln, über die erste legen und die Summe aller Bildpunkte mit ähnlichen Farbwerten bilden. Die X-Achsen-Symmetrie unterteilt ein Bild horizontal in der Mitte, die Y-Achsen-Symmetrie vertikal (vgl. Abbildung 14).

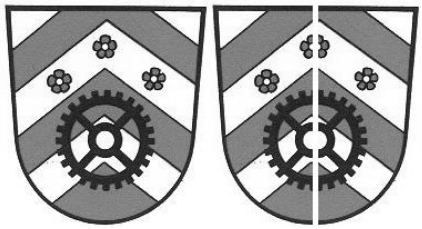

Abbildung 14: Beispiel für die Y-Achsen-Symmetrie

Die Features für die inversen Symmetrien führen dieselben Operationen wie X- und Y-Achsen-Symmetrie durch, zählen aber explizit die Anzahl der Pixel mit unterschiedlichen Farben. Dabei wird keine fixe Zuordnung von Farben beachtet, das heißt ein grünes Pixel in einem Feld kann einem blauen oder roten im anderen gegenüberliegen und wird beide Male als symmetrisch gezählt. Dadurch werden Bilder gefunden, bei denen sich die Symmetrie durch einheitliche Flächen in unterschiedlichen Farben äußert. Die diagonalen Symmetrien schneiden Bilder an den Diagonalen. Dazu werden die Bilder zuerst auf quadratische Größe gezerrt und anschließend durch eine entsprechende Vergleichsfunktion bewertet. Es zeigt sich (vgl. Abschnitt 3.3.3), daß diese beiden Symmetrien meist gemeinsam auftreten. Die Kreuz-Symmetrie schließlich viertelt ein Bild (horizontale und vertikale Teilung) und berechnet die Symmetrie aller vier Teile. Diese Features speichern als Featurevektor die Anzahl übereinstimmender Bildpunkte. Als Distanzmaß wird die absolute Differenz der einstelligen Featurevektoren, normiert auf das Intervall [0,1] verwendet:

$$D_{x,y} = \frac{\left| f_x - f_y \right|}{m} \tag{24}$$

Hierbei sind f_x und f_y die Featurevektoren und m die maximal mögliche Differenz zweier Vektoren. Abbildung 15 zeigt ein Suchbeispiel für die Y-Achsen-Symmetrie, wobei das erste Bild das Suchbeispiel ist.

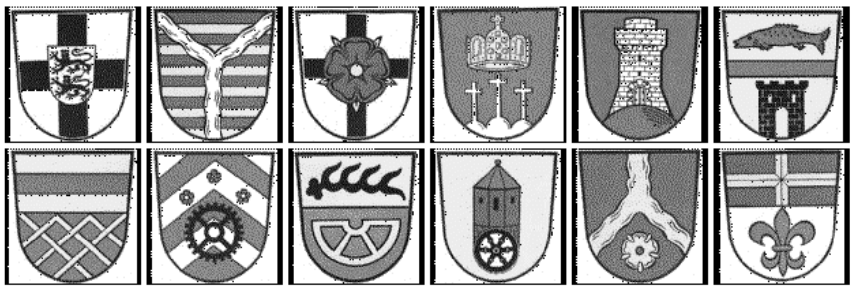

Abbildung 15: Suchbeispiel für das Feature zur Messung der Y-Achsen-Symmetrie

Das Symmetrietyp-Feature verwendet die Informationen der anderen Symmetrie-Features, um Bildern einen sogenannten Symmetrietyp (Kennzahl für die vorkommenden Symmetrien) zuzuweisen. Die Funktion zur Featureextraktion berechnet die Featurewerte aller obigen Features und prüft dann für jeden Wert, ob er einen bestimmten, für die Featureklasse (und teilweise das vorhandene Bildmaterial) typischen, Schwellwert überschreitet. Wenn ja, wird dem Featurevektor des betrachteten Bildes das Flag für die betrachtete Symmetrieart hinzugefügt. Abbildung 16 ist ein Beispiel für ein Bild, das alle (von diesem Feature unterschiedenen) Symmetriearten aufweist.

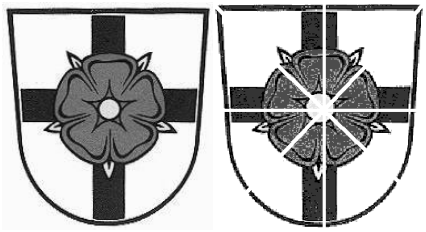

Abbildung 16: Beispiel für das Symmetrietyp-Feature

Zur Symmetrieerkennung werden derzeit folgende Schwellwerte (heuristisch festgelegt, in Prozent zur maximal möglichen Anzahl übereinstimmender Bildpunkte) verwendet:

Feature	Schwellwert (Prozent)	Flag
X-Achsen-Symmetrie	35.3	1
Y-Achsen-Symmetrie	37.7	2
Inverse X-Achsen-Symmetrie	36.7	1
Inverse Y-Achsen-Symmetrie	31.5	2
Diagonale ansteigende Symmetrie	24.7	4
Diagonale absteigende Symmetrie	26.7	8
Kreuz-Symmetrie	26.7	16

Daraus wird ersichtlich, daß eine hohe Symmetrie bereits bei ca. 30 Prozent Übereinstimmungen vorhanden ist. Außerdem haben die inversen Symmetrien dasselbe Flag wie ihre Grundsymmetrien da sie sich auf den selben Symmetrietyp beziehen.

Die Distanzfunktion für den Symmetrietyp unterscheidet drei Fälle:

- Sehr ähnlich, wenn zwei Bilder gleiche Symmetrietypen haben.
- Verwandt, wenn zwei Bilder zumindest ein Symmetrieflag gemeinsam haben.
- Unähnlich: sonst.

Die Symmetriefeatures dienen vor allem der schnellen Vorauswahl passender Bilder; der Symmetrietyp beschreibt zudem die Symmetrien eines Bildes mit einem einzigen Wert sehr genau und spielt folglich bei der Einteilung der Bilder in Cluster eine große Rolle.

3.1.2 Wappenspezifische Features

3.1.2.1 Globale heraldische Features

Speziell für Wappen wurden Features entwickelt, die heraldische Schnitte erkennen, Siegelabdrücke handhaben und die Komplexität eines Bildes bestimmen können, um z. B. eine Aussage darüber machen zu können, ob es sich bei einer bestimmten Abbildung um ein Heroldsbild handelt oder eine andere Art von Wappen.

3.1.2.1.1 Segmentierungs-Feature

Das Segmentierungs-Feature (Klassenname: QbWappSegFeatureClass) dient dazu, eventuelle heraldische Schnitte (das sind Segmentierungen bei zusammengesetzten Wappen; vgl. Abschnitt 2.3) festzustellen. Dazu wird folgendermaßen vorgegangen:

1. Finden von Trennlinien: als Kandidaten für Trennlinien werden gerade verlaufende Übergänge von einer Farbe zu einer anderen (Kanten) gesucht. Dabei wird für jede Linie die Startposition im Bild, ihre Länge und der Winkel zur Horizontalachse aufgezeichnet. Geraden sind so definiert, daß eine ununterbrochene Kante maximal N von der Geraden abweichende Punkte haben darf. Es hat sich als günstig erwiesen, für genormte Wappenbilder N mit 20 Pixel festzulegen. Außerdem wurde versucht, Unterbrechungen von Kanten aufgrund von Fehlern im Bildmaterial bis zu einem gewissen Grad zu tolerieren. Dies erwies sich aber als ungünstig, da dadurch mehr nicht zusammengehörende Kanten falscherweise miteinander verbunden wurden als richtigerweise zusammengehörende.
2. Für jede gefundene Trennlinie wird untersucht, ob sie ein Kandidat für eine Segmentierungslinie ist. Dazu muß sie an einer bestimmten Position beginnen (z. B. oben

in der Bildmitte), eine bestimmte Länge haben (z. B. Hälfte der Höhe des Wappens) und eine bestimmte Lage aufweisen (senkrecht zur Horizontalachse). Erfüllt eine Trennlinie diese Anforderungen, wird sie als Segmentierungslinie vermerkt.

3. Aus der Summe dieser Segementierungslinien wird mit Hilfe von Expertenwissen für heraldische Schnitte auf einen Segmentierungstyp geschlossen. Das Expertenwissen besteht in einer Funktion, die für eine Gruppe von Segmentierungslinien angibt, welcher Schnitt daraus erstellt werden kann. Derzeit werden 22 verschiedene Typen unterschieden, von unsegmentierten über vertikal oder horizontal geteilten Wappen zu komplizierten Segmentierungsarten mit drei oder vier Segmenten. Abbildung 17 zeigt neben einigen anderen typischen Eigenschaften von Wappen eine Segmentierung in T-Form.

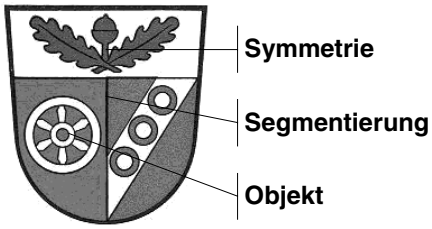

Abbildung 17: Eigenschaften von Wappenbildern

Dieser heraldische Schnitt entspricht dem Segmentierungstyp 7, der besonders häufig in bürgerlichen bayrischen Wappen vorkommt, die zumeist im oberen Viertel das bayrische Hoheitszeichen tragen (vgl. Abbildung 18).

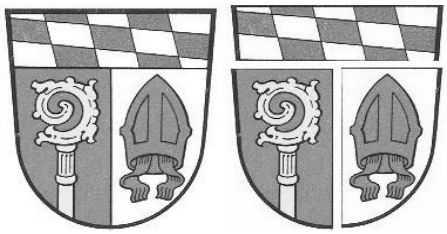

Abbildung 18: Beispielbild für eine Segmentierung

Die Genauigkeit der Schnitterkennung beträgt in der Testdatenbank (siehe Abschnitt 3.1.3) 86 Prozent, wobei die Mehrzahl der Fehler auf mangelhaftes Bildmaterial zurückzuführen sind (Nicht-Erkennbarkeit von Segmentierungslinien, etc.). Die Mehrzahl der Fehlzuordnungen ist auf Nichterkennen von Segmentierungen zurückzuführen. Neben dem oben geschilderten Ansatz zur Erkennung von Segmentierungen wurde versucht, aus den Objektbeschreibungen die Schnitte zu erkennen. Das führte aber zu weniger guten Ergebnissen als bei der auf Trennlinien basierenden Methode und wurde deshalb nicht weiter verfolgt. Das Distanzmaß für das Segmentierungsfeature unterscheidet drei Möglichkeiten:

- Gleiche Segmentierung, wenn die Typvariablen (einstelliger Featurevektor) den gleichen Wert haben.

- Ähnliche Segmentierung: der Benutzer kann festlegen, welche Segmentierungen einander ähnlich sind: z. B. die oben angegebene T-Form kann als ähnlich zur I-Form oder Kreuz-Form bewertet werden.

- Unähnlich: sonst.

Da die heraldischen Schnitte ein wesentliches Design-Element von Wappen sind, spielt dieses Feature in vielen Abfragen eine wesentliche Rolle. Die sehr effiziente Distanzfunktion trägt viel zu einer guten Retrievalqualität (Performance, Ergebnisqualität) bei.

3.1.2.1.2 Siegel-Feature

Das Siegel-Feature (Klassenname: QbSiegelFeatureClass) wurde ursprünglich zur Recherche nach Suchbildern, die keine Farbangaben beinhalten, entwickelt. Alte Wappen liegen häufig nur noch als Siegel oder Siegelabdrucke vor, eine Suche kann also nur aufgrund von Form- oder Textureigenschaften erfolgen. In Abbildung 19 wird ein Wappen und sein (möglicher) Siegelabdruck dargestellt.

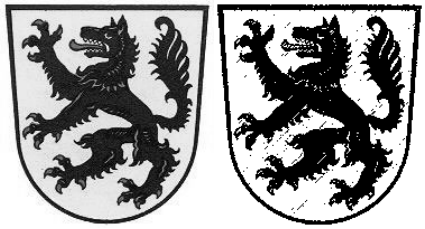

Abbildung 19: Beispielbild für einen Siegelabdruck

Das Siegel-Feature nutzt die heraldische Regel, die die Zuordnung von Tinkturen zu Mustern festlegt. Es speichert keinen Featurevektor ab sondern eine Featurematrix, die nach folgenden Regeln gebildet wird:

1. Das Bild wird standardisiert auf eine Größe von 64x64 Bildpunkten, wobei jeder Bildpunkt mit der vorherrschenden Farbe der auf ihn abgebildeten Region des Ausgangsbildes belegt wird.

2. In der Featurematrix wird jeder Bildpunkt durch eine binäre 4x4-Matrix dargestellt, welche die Schraffur der jeweiligen Farbe wiedergibt. Dabei ist klar festgelegt, daß z. B. bei einer horizontalen Schraffur (Farbe Blau) immer die erste und die dritte Zeile schwarz (Wert "1") und die zweite und vierte weiß ("0") dargestellt wird. Diese Festlegung ist wichtig für die effiziente Suche nach Featurematrizen.

Der Distanzvergleich zweier Featurematrizen erfolgt, indem sie übereinandergelegt werden und die Anzahl der übereinstimmenden 0/1-Werte gezählt wird. Dabei ist neben der globalen Bildsuche auch die Suche nach Teilfeldern möglich, wobei anzugeben ist, nach welcher Region eines Bildes zu suchen ist und wo nach dieser Region zu suchen ist (an der selben

Stelle wie im Suchbild oder beliebig). Es ist klar, daß eine solche Teilbildsuche im Vergleich zu anderen Features sehr langsam abläuft; das Siegel-Feature ist daher zur Feinauswahl passender Bilder gedacht und sollte stets in Kombination mit anderen Features verwendet werden. Bei Tests hat sich herausgestellt, daß das Siegel-Feature neben seinem eigentlichen Zweck auch hervorragend zur Suche nach Farbbildern geeignet ist, da es eine sehr effiziente Kombination von Farb- und Positionswissen darstellt. Nachteilig ist aber festzustellen, daß zur Speicherung der Featurematrizen mehr Speicher als für herkömmliche Features benötigt wird und die Genauigkeit dieser Methode begrenzt ist, wodurch ihre Anwendung für natürliche Bilder (z. B. Landschaftsphotos, etc.) mit vielen Farbübergängen nur mittelmäßige Ergebnisse zeitigt. Abbildung 20 zeigt ein typisches Suchbeispiel.

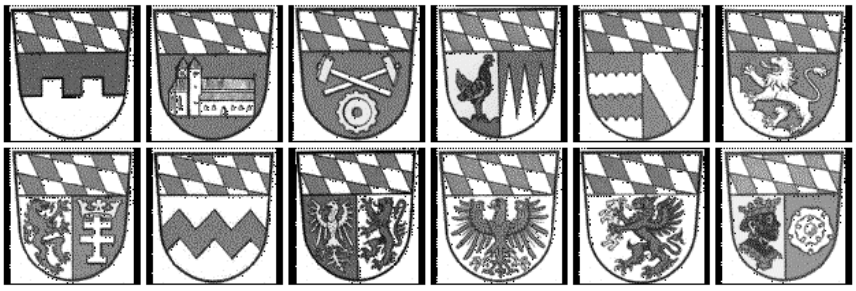

Abbildung 20: Suchbeispiel für das Siegel-Feature

3.1.2.1.3 Komplexitäts-Feature

Das Komplexitäts-Feature (Klassenname: QbBildTypFeatureClass), verwendet die von den Vektorisierungs- und Objekterkennungsroutinen, die für das Objekt-Layout-Feature implementiert wurden, erzeugten Informationen. Um eine Aussage über die Komplexität eines Bildes machen zu können, werden in einem sechsstelligen Featurevektor folgende Kennzahlen abgelegt:

- Anzahl der Objekte in einem Bild
- Summe der Kanten über alle Objekte
- Mittelwert und Varianz der Kantenlängen
- Mittelwert und Varianz der Winkel zwischen den Kanten

Mithilfe dieser Informationen läßt sich der Komplexitätsgrad eines Wappen soweit bestimmen, daß zumindest Heroldsbilder (bestehend aus geometrischen Formen) von Wappen mit komplexen Darstellungen von Lebewesen und anderen Objekten unterschieden werden können. Die Distanzfunktion vergleicht die genormten Einträge der Featurevektoren und bildet die euklidische Distanz, wobei alle Einträge dasselbe Gewicht haben. Das Komplexitäts-Feature dient der schnellen Vorauswahl; es ist nicht besonders genau, erfüllt seinen Zweck, die Bildmenge nach Komplexität zu ordnen, aber recht gut.

3.1.2.2 Regionale Wappen-Features

Eine besondere Gruppe von Features sind die regionsorientierten Farbfeatures, die das Segmentierungsfeature mit den Farbfeatures kombinieren:

Feature	Klassenname	Kombination
Regionales Farbhistogramm	QbWappRFarbFeatureClass	Segmentierung + Farbhistogramm
Regionale Zahl von Farbabstufungen	QbWappRNKFeatureClass	Segmentierung + Zahl der Farbabstufungen
Regionale Zahl von Tinkturen	QbWappRFAnzFeatureClass	Segmentierung + Zahl der Tinkturen

Als Featurevektor wird jeweils der Segmentierungstyp sowie für jede Region ein für das kombinierte Feature charakteristischer Subvektor abgelegt. Da keiner der betrachteten heraldischen Schnitte aus mehr als vier Regionen besteht, werden immer vier Regionen abgespeichert, wobei nicht benötigte Einträge durch Nullen aufgefüllt werden. Abbildung 21 stellt das Ergebnis des regionalisierten Farbhistogramms für das Wappen von Füssen (Bayern) schematisch dar.

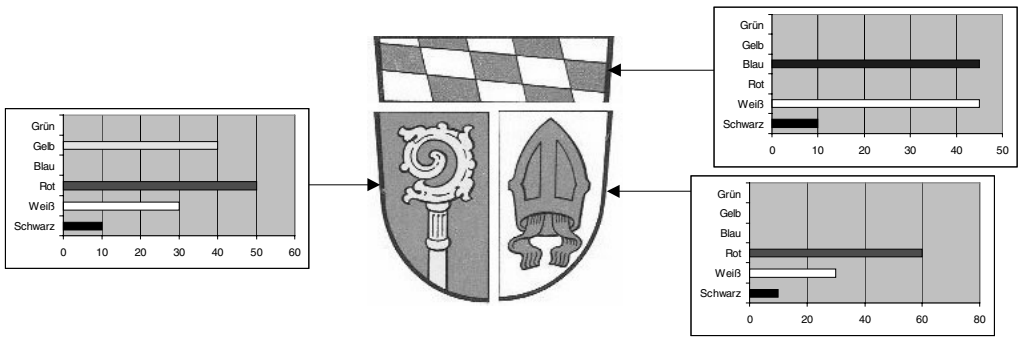

Abbildung 21: Beispielbild für regionale Farbhistogramme

Der Distanzvergleich funktioniert folgendermaßen:

1. Prüfung, ob zwei Bilder denselben Segmentierungstyp haben. Falls nein, werden die Bilder als "unähnlich" bewertet.

2. Wenn der Segmentierungstyp gleich ist, erfolgt ein Distanzvergleich der Subvektoren mit der Methode des Basisfeatures. Als Distanzwert wird der Mittelwert der Distanzen aller Regionen verwendet.

Daraus ist ersichtlich, daß die regionsorientierten Features nicht einfach eine Kombination der Basismethoden sind und durch ein Suchmodell, welches beide Features enthält, ersetzt werden können, sondern eine eigene Form von Ähnlichkeit messen. Im Fall des regionalen Tinkturhistogramms handelt es sich um ein häufig gewünschtes, lokalisiertes Farbhistogramm, wobei die Regionalisierung aufgrund von Wissen über das verwendete Bildmaterial erfolgt. Diese Features sind sehr mächtig und führen daher zu guten Suchergebnissen (vgl. Abbildung 22).

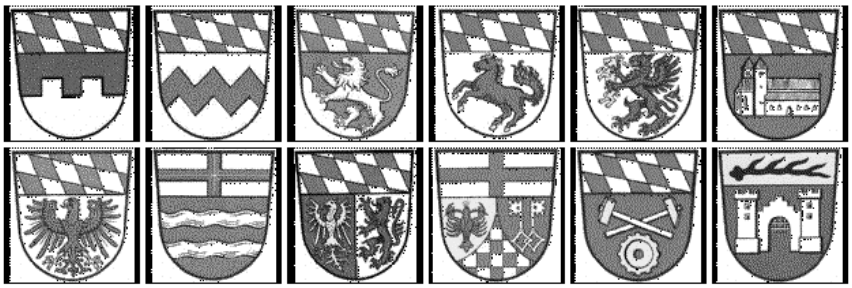

Abbildung 22: Suchbeispiel für das regionale Farbhistogramm

3.1.2.3 Features, die sich als ungeeignet erwiesen haben

Neben den oben angegebenen 19 Features wurden noch einige andere Ideen umgesetzt, die sich aber aus verschiedenen Gründen nicht bewährt haben. Ein Versuch waren lokale RGB-Farbanteilshistogramme, wobei verschiedene Stufen realisiert wurden: 1x1 / 2x2 / 3x3 / 4x4 Regionen, um von einer schnellen Vorauswahl zu immer feineren Unterscheidungen zu kommen. Ein anderer Versuch war der Vergleich regionsorientierter Objektbeschreibungen, wie sie von der oben angegebenen Objekterkennung erzeugt werden. Es wurde also verglichen, ob in zwei Bildern Objekte mit der selben Größe, Position, Kantenzahl, gleichen Extrempunkten, etc. existieren. Dieser Vergleich auf low-level-Ebene erwies als ungünstig (schlechte Ergebnisqualität, zu viele Einflußfaktoren); die Betrachtung auf der höheren Abstraktionsebene des Objekt-Layout-Features erzielte bei weitem bessere Ergebnisse.

Außerdem wurden verschiedene mathematische und statistische Verfahren versucht (Regression, Farbvarianzen, Determinante, etc.), deren Ähnlichkeitsbewertung sich für den menschlichen Beobachter aber stets als kaum nachvollziehbar erwies. Ein anderer Versuch war, das Komplexitäts-Feature (ähnlich wie das Farbhistogramm) zu regionalisieren. Dabei wurde festgestellt, daß zur Beurteilung von Komplexität die Regionen oft zu klein sind und auf der Ebene des Gesamtbildes zutreffendere Informationen gewonnen werden können. Schließlich wurde mit einem Feature versucht, in Bildern nach einzelnen Objekten mit bestimmten Eigenschaften zu suchen. Dabei erwies sich aber die gewählte Form der Objektbeschreibung als zu allgemein, so daß gesuchte Objekte auch in Bildern in zutreffender Form gefunden wurden, wo kein heraldisches Objekt mit dieser Form existierte. Die Stärke des Objekt-Layout-Features liegt in der Beschreibung der Relation von Objekten; der Versuch der Erkennung einzelner Objekte erzielte keine akzeptablen Ergebnisse.

3.1.2.4 Nicht realisierte, mögliche Feature-Erweiterungen

Neben diesen Features hätte man selbstverständlich noch eine Reihe anderer, z. B. aus der Literatur bekannter Features realisieren können. Denkbar wären unter anderem folgende Ansätze:

- Verwendung von Momenten in der Objekterkennung: dadurch lassen sich Objekte mit wenigen Werten (Zirkularität, Schwerpunkt, etc.) beschreiben.

- Texturfeatures: im Bereich Texturen gibt es eine Menge unterschiedlicher Ansätze, die z. B. für die Suche nach Siegeln hätten genutzt werden können. Für den allgemeinen Fall einer Suche nach Wappen erscheint die Verwendung von Textur-Features aber wenig aussichtsreich, da Muster, wenn überhaupt, nur in stark vergröberter Form vorkommen.

- Objekterkennung: Zahl und Aussehen, der in Wappen verwendeten Objekte sind (einigermaßen) beschränkt. Aufbauend auf den Randinformationen der Objekterkennung hätte man versuchen können, mit Hilfe einer Wissensbasis semantische Informationen über die dargestellten Objekte abzuleiten.

3.1.3 Testumgebung

Die oben angegebenen Features, sowie alle in den folgenden Kapiteln behandelten Algorithmen, wurden in folgender Arbeits- und Testumgebung realisiert.

- Als Multimedia-IR-System für CBIR wurde aus den unten angegebenen Gründen das QBIC (Query by Image Content) System (Version 2) von IBM gewählt. Es wurde als Datenbank, als API zur Implementierung zusätzlicher Features sowie als Prototyp für ein Retrieval-System genutzt.

- Als Arbeitsumgebung wurden der GNU C/C++-Compiler verwendet, die GNU Entwicklungsumgebung (GNU Debugger, Make-Utility, etc.) sowie die Scriptsprache Perl (Version 5) zur Implementierung der Benutzerschnittstelle in Form von CGI-Scripts. Die HTML-basierte Benutzerschnittstelle wurde mithilfe eines Apache-Webservers im Web verfügbar gemacht.

- Diese Testumgebung wurde eingerichtet auf einer Intel-basierten Workstation mit Internetanschluß und Linux als Betriebsystem.

Für QBIC sprachen im Vergleich zu anderen Multimedia-IR-Systemen, die inhaltsorientierte Suche unterstützen (Virage, Photobook, VisualSEEk, etc.), unter anderem folgende Vorteile:

- QBIC besitzt ein offenes Framework zum Hinzufügen von neuen Features sowie zum Erstellen von Client/Server-Anwendungen, wie z. B. einer neuen Suchmaschine. QBIC erlaubt die Programmierung beliebiger Features, wobei der Entwickler durch eine

standardisierte, objektorientierte Sicht auf das Bildmaterial sowie die Datenbank selbst unterstützt wird. Negativ in QBIC (Version 2) war die mangelhafte Dokumentation, die eine mehrmonatige Einarbeitungszeit in das System erforderlich machte.

- QBIC unterstützt viele unterschiedlicher Bildformate. Es können unter anderem TIF-, GIF-, JPG-, und BMP-Bilder geladen und durch das API über eine standardisierte Bildklasse manipuliert werden.

- QBIC unterstützt die Verwaltung mehrerer verschiedener Bilddatenbanken in sogenannten Katalogen, wobei für jeden Katalog unterschiedliche Features definiert werden können. Dadurch wird es möglich, z. B. eine Produktions- von einer Entwicklungsdatenbank zu unterscheiden.

Abbildung 23 zeigt schematisch den Aufbau der Testumgebung.

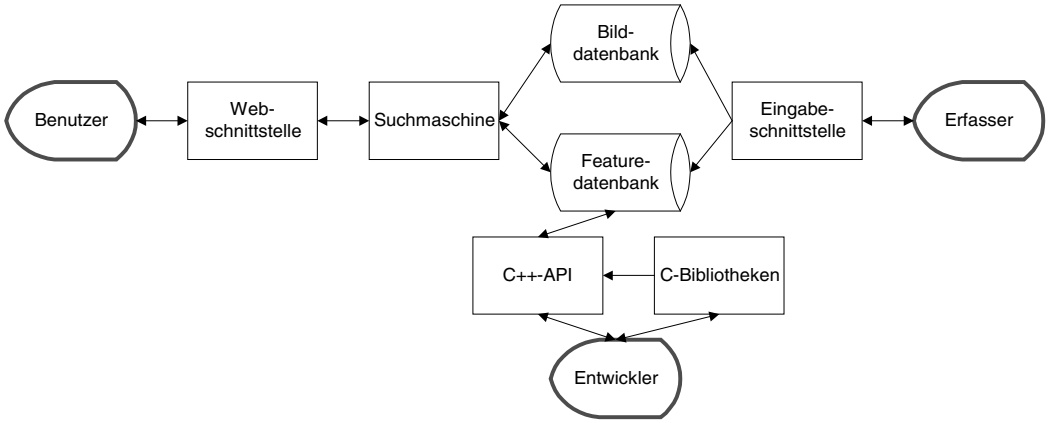

Abbildung 23: Aufbau der QBIC-Testumgebung

Zentrales Element ist eine Datenbank, in der die Featurevektoren abgelegt werden. Die damit verknüpften Bilder werden in der Datenbank nur in Form eines Schlüssels (Dateiname) abgelegt; die eigentlichen Bilddaten befinden sich außerhalb der Datenbank. Das Hinzufügen beziehungsweise Entfernen von Bildern geschieht durch die Eingabeschnittstelle QbMkDbs. Hier können Datenbanken und Kataloge definiert werden sowie Features zu Katalogen hinzugefügt werden. Das Hinzufügen eines Bildes zu einem Katalog, wobei die Featurevektoren aller für diesen Katalog definierten Features berechnet werden, erfolgt ebenfalls durch die Eingabeschnittstelle.

Der Benutzer greift auf QBIC über eine Webschnittstelle zu. Da die Standardschnittstelle nur die Recherche nach jeweils einem der QBIC-Standardfeatures unterstützt, wurde im Rahmen dieses Projektes eine neue Webschnittstelle programmiert, die das Zusammenstellen von Suchmodellen, die aus mehreren Features bestehen, erlaubt. In Abbildung 24 ist ein Screenshot dieser Schnittstelle, wie sie vor einer Suche aussieht, abgebildet. Im Bereich A werden zufällig ausgewählte Beispielbilder dargestellt, wobei zu jedem Bild der Name und ein Link auf eine vergrößerte Darstellung angegeben ist. Im Bereich B kann ein Suchmodell

definiert werden. Die Features werden aus einer Liste ausgewählt und zu jedem Feature können ein Gewicht, ein Schwellwert und optionale Parameter angegeben werden. Durch den Button C schließlich kann die Schnittstelle zurückgesetzt werden. Eine Suche wird nach der Definition eines Suchmodells durch das Auswählen eines der Beispielbilder initiiert. Ein Beispiel für ein Suchergebnis ist in Abbildung 33 im Abschnitt 3.2.5 abgebildet.

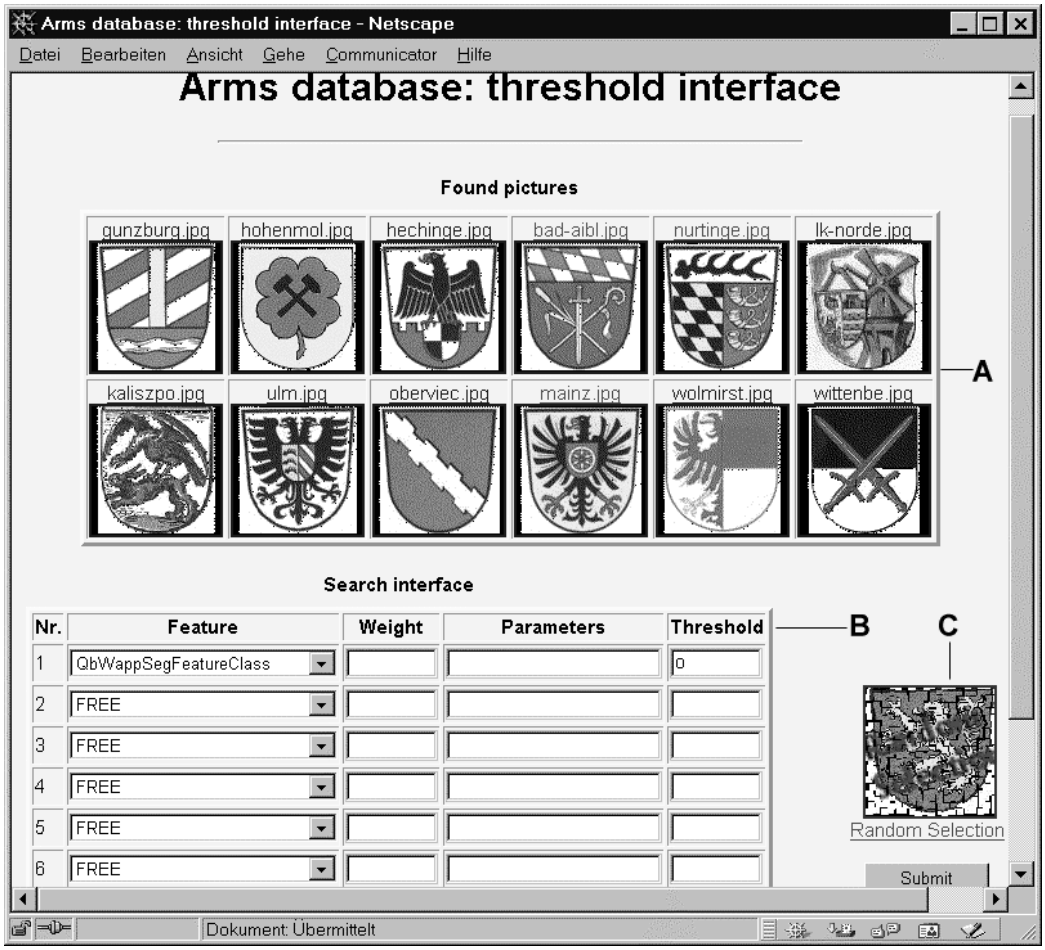

Abbildung 24: Webschnittstelle im Ausgangszustand

Im Hintergrund der Webschnittstelle steht eine Client/Server-orientierte Suchmaschine. Auf der Basis der Standard-Suchmaschine QbQBE wurde eine für Suchmodelle geeignete Suchmaschine entwickelt, die daneben auch noch einige zusätzliche, in den folgenden Kapiteln behandelte, positive Eigenschaften aufweist. Gesucht wird entweder anhand von Beispielbildern (Query by Example) oder aufgrund von als Parametern übergebenen Featurevektoren. Das Ergebnis einer Suche besteht aus einer dreiteiligen Antwort:

1. Suchergebnis: alle der Suche entsprechenden Bilder mit den Distanzwerten für alle Features sowie der gewichteten Summe aller Distanzwerte, aufgrund der die Reihung der Ergebnismenge erfolgt.

2. Suchstatistik: je Feature wird unter anderem die Anzahl der Treffer sowie die Dauer der Distanzberechnung ausgegeben.

3. Suchschnittstelle: es wird das Suchmodell mit allen Parametern ausgegeben und dem Benutzer die Möglichkeit gegeben, Parameter anzupassen, Features hinzuzufügen oder aus dem Suchmodell zu entfernen.

Der Anwendungsentwickler verwendet QBIC über ein C++-API, das unter anderem das Hinzufügen von Features ermöglicht. Jedes Feature ist dabei eine Tochterklasse der Klasse QbFeatureClass, die einige globale Eigenschaften eines Features definiert. Es müssen zumindest folgende Memberfunktionen für ein Feature definiert werden:

- ComputeFeature(): Diese Funktion erzeugt aus einer Bildklasse und einer Parameterklasse den für das Feature spezifischen Featurevektor. Das Bild wird in Form einer Bitmap, deren Einträge auf eine Look-up-table verweisen, an die Funktion übergeben.

- Distance(): nimmt als Parameter zwei Feature-Objekte und berechnet den Distanzwert als einen auf [0,1] normierten Real-Wert.

- ToByteString(): wandelt den Featurevektor in einen String um, so daß er in der Featuredatenbank gespeichert werden kann. Diese Funktion wird von QbMkDbs nach der Berechnung des Featurevektors ausgeführt.

- FromByteString(): wandelt einen String in einen Featurevektor um. Diese Funktion wird von der Suchmaschine vor jedem Distanzvergleich für die zu vergleichenden Objekte ausgeführt.

Neben den Featureklassen gibt es noch eine Reihe anderer Klassen zur Durchführung von Recherchen, dem Zugriff auf Bilder, zum Öffnen und Schließen von Datenbanken sowie zur Verwaltung von Listen. Trotz des guten APIs von QBIC mußten dennoch einige Funktionen hinzugefügt werden. Dazu wurden einige C/C++-Bibliotheken erstellt, die unter anderem folgende Funktionen enthalten:

- Bibliothek "vector.c": in dieser Bibliothek sind die Datenstrukturen definiert, mit denen Objekte beschrieben werden (Objektinhalt, Rand, Extrempunkte, etc.) sowie die Funktionen implementiert, mit denen die Vektorisierung durchgeführt wird (Sobel-Operator, Schwellwert-Verfahren, etc.). Außerdem sind hier Hilfsfunktionen zum Normalisieren der Bildgröße, der Farbenzahl und zur Umwandlung von Farb- in Grauwertbilder enthalten.

- Bibliothek "objecte.c": diese Bibliothek enthält die Definitionen und Datenstrukturen, mit denen Objekte auf höherer Ebene für das Objekt-Layout-Feature beschrieben werden. Außerdem sind hier die Funktionen definiert, mit denen Objekte erkannt und verglichen werden.

- Bibliothek "wappen.c": in dieser Bibliothek sind die Funktionen zur Erkennung heraldischer Schnitte definiert (Erkennen von Segmentierungslinien, Zuordnung von Segmentierungstypen, etc.). Außerdem sind in dieser Bibliothek die Wappenfarben definiert.

In dieser Umgebung wurden 444 deutsche Gemeinde- und Kreiswappen geladen, anhand derer die verwendeten Methoden evaluiert wurden.

3.2 Suchmodelle

Dieser Abschnitt beschäftigt sich mit der Anwendung der Features im CBIR-Suchprozeß. Der übliche Ansatz besteht in der Verwendung eines einzigen oder fixer Kombinationen von Features für eine Suche, wobei sich als offensichtlicher Nachteil eine starre, meist kaum vom Benutzer beeinflußbare Definition der Ähnlichkeit ergibt. In manchen Systemen wird dem Benutzer die Möglichkeit gegeben, wenigstens eine Gewichtung der verwendeten Features vorzunehmen und so auf das Suchergebnis einzuwirken. Weil der Benutzer mit der Vergabe von Gewichten für Features, deren Funktionsweise er eigentlich nicht kennt, im Grunde überfordert ist, wird in anderen Systemen diese Möglichkeit nicht angeboten. Dem Benutzer bleibt dann nur die Möglichkeit, sein Suchobjekt anzugeben und zu hoffen, daß die Ähnlichkeitsdefinition des Systems, mit der er arbeitet, der seinen soweit entspricht, daß er durch eine iterative Verbesserung der Suche (Auswahl anderer Features, Anpassung der Größe der Ergebnismenge) das gewünschte Ergebnis erhält.

Da diese Situation sehr unbefriedigend ist, wird im Rahmen dieser Arbeit ein allgemeines Konzept zur Beschreibung von Ähnlichkeit für CBIR erstellt. Teilweise wird es aus bereits vorhandenen Ansätzen abgeleitet und verallgemeinert, teilweise aber auch die im CBIR allgemeingültige Ähnlichkeitsdefinition erweitert und verfeinert. Die einer Suche zugrunde liegende Ähnlichkeitsdefinition wird dabei durch ein sogenanntes Suchmodell (query model) beschrieben. Im Abschnitt 3.2.1 werden Suchmodelle definiert, in Abschnitt 3.2.2 wird gezeigt, wie man sie auf die Wappen-Features anwenden kann, Abschnitt 3.2.3 führt in die Verwendung von Schwellwerten zur Regulierung der Ergebnismenge ein, Abschnitt 3.2.4 zeigt, wie sich die Ergebnisqualität durch Schwellwerte verändert und Abschnitt 3.2.5 beschreibt die Testumgebung (Webschnittstelle, Suchmaschine), die zur Implementierung und Bewertung der Suchmodelle verwendet wurde.

3.2.1 Aufbau von Suchmodellen

Ein Suchmodell definiert Ähnlichkeit durch die Schichten, aus denen es besteht. Jede Schicht besteht aus einer Featurefunktion, die bestimmt, welche Eigenschaft eines Bildes betrachtet wird, einer Distanzfunktion, die angibt, wie grundsätzlich zwei Featurevektoren verglichen werden, einem Gewicht zur Bestimmung der Bedeutung einer Schicht im Suchmodell sowie einem Schwellwert als Maß für die Größe der Ergebnismenge. Die Anzahl der Schichten ist variabel. Die Größe der Ergebnismenge wird in allen derzeit verwendeten Systemen und auch theoretischen Ansätzen durch eine absolute Zahl angeben. Im Rahmen dieser Dissertation hat sich aber gezeigt, daß die indirekte Bestimmung der Ergebnismenge auf Schicht-Ebene durch eine Maßzahl für die maximale Distanz eines untersuchten Objektes zum Suchobjekt, den sogenannten Schwellwert, bei weitem bessere Ergebnisse erzielt (vgl. Abschnitt 3.2.4).

Welches Suchmodell für eine bestimmte Anfrage benutzt wird, kann entweder, im Falle eines Experten, durch den Benutzer angegeben werden oder mithilfe eines entsprechenden Generierungsalgorithmus automatisch abgeleitet werden. Suchmodelle können als Informations- oder Bildfilter angesehen werden, wobei jede Schicht von der Ausgangsmenge eine Untermenge von für die in der Schicht betrachteten Eigenschaft weniger ähnlichen Bildern abschneidet (vgl. Abbildung 25).

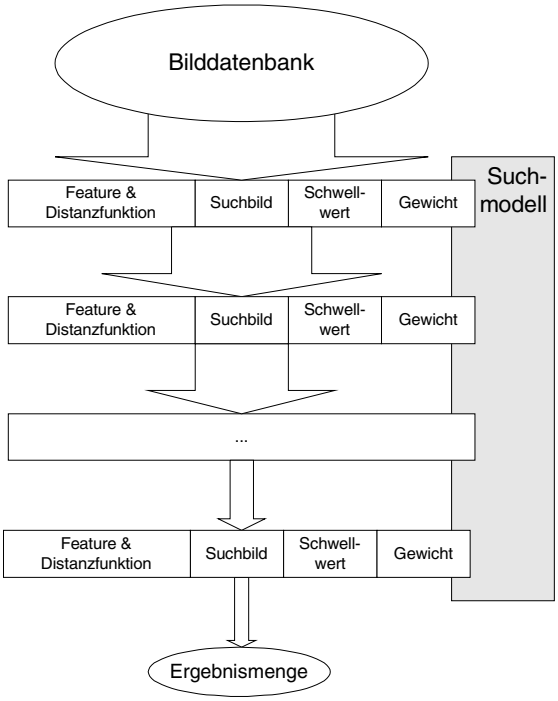

Abbildung 25: Suchmodelle als Informationsfilter

Jene Bilder der Ausgangsmenge, die alle Eigenschaften der durch das Suchmodell definierten Ähnlichkeit erfüllen, bilden die Ergebnismenge. Die einzelnen Schichten sind von einander unabhängig.

Aus einem anderen Blickwinkel kann man Suchmodelle als Beschreibungen für Bildcluster sehen. Wie im Literaturüberblick erwähnt, verfügt eine Menge von Datenvektoren normalerweise über eine Clusterung von Gruppen mehr als durchschnittlich ähnlicher Elemente. Diese Cluster bilden sich aufgrund übereinstimmender Merkmale und können durch Techniken, wie Clusteranalyse oder Self-organizing Maps, sichtbar gemacht werden. Dabei wird üblicherweise zutreffen, daß unterschiedliche Kriterien für die Bildung der einzelnen Cluster verantwortlich sind. So kann eine Bildmenge z. B. aus Clustern mit ähnlicher Farbverwendung und ähnlicher Texturverwendung bestehen. Um erstere zu finden, ist es sinnvoll, Farbfeatures zu verwenden, nach letzteren wird man mit Texturfeatures suchen; das heißt für jeden Cluster gibt es ein passendes Suchmodell und normalerweise wird nicht ein Suchmodell für alle Cluster einer Bildmenge passend sein (vgl. Abbildung 26).

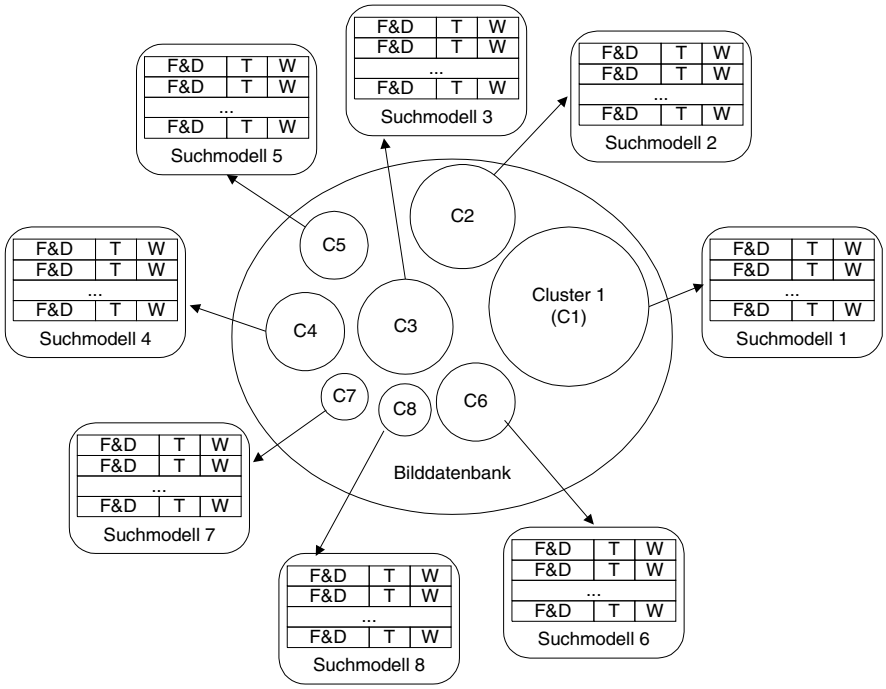

Abbildung 26: Suchmodelle als Entsprechung für die natürliche Clusterung

Im folgenden wird analysiert, welche Ergebnisse sich durch die Verwendung von Suchmodellen bei der Suche nach Wappenbildern erzielen lassen.

3.2.2 Anwendung der Suchmodelle auf die Wappen-Features

Das Konzept der Suchmodelle wurde in der Testumgebung für Wappen-Features realisiert. Anhand von Precision und Recall können die Qualität des Ansatzes sowie der Features experimentell überprüft werden. Zur Prüfung wurde folgendermaßen vorgegangen:

1. Formulierung von verbalen Suchfragen sowie Auswahl passender Beispielbilder.

2. Ableitung möglicherweise passender Suchmodelle, wobei versucht wurde, verschiedene Kombinationen auszuprobieren. Es wurden je Frage ca. drei verschiedene Suchmodelle mit zwei bis vier Features abgeleitet und evaluiert.

3. Durchführung von Testserien für alle Suchmodelle und Auswertung der Rückgabemengen. Für die einzelnen Tests einer Testserie für ein Suchmodell wurden jeweils verschiedene Suchbilder verwendet.

4. Berechnung der Mittelwerte von Recall und Precision für alle Modelle und Testserien sowie Auswahl des jeweils besten Modells. Die Feststellung der für eine Frage relevanten Bilder in der Datenbank wurde als vollständige Prüfung der Bildermenge (444 Wappen) durchgeführt. In der folgenden Tabelle sind die gestellten Suchfragen sowie die jeweils besten Suchmodelle angeführt:

Nr.	Suchfrage	Bestes Suchmodell
1.1	Horizontal symmetrische, rote Bilder	X-Achsen-Symmetrie, Tinktur-Histogramm
1.2	Wappen mit einem vertikalen Schnitt und wenigen Tinkturen	Segmentierung, Zahl der Farbabstufungen
1.3	Bilder mit einem großen, zentralen Objekt in weiß oder gelb auf blauem Hintergrund ohne Segmentierung	Objekt Layout, Tinktur-Histogramm

Abbildung 27 zeigt drei typische Suchbilder für diese Suchfragen, wie sie in den Tests verwendet wurden.

1.1 1.2 1.3

Abbildung 27: Beispielbilder für die Suchfragen 1.1 - 1.3

In der folgenden Tabelle sind für die oben angegebenen Suchmodelle einige Testergebnisse Tests angegeben. Daraus ist zu ersehen, daß Precision und Recall für die einzelnen, teilweise recht unterschiedlichen Abfragen meist ein Niveau von zumindest 80 Prozent erreichen.

Test	Zurückgegebene passende Bilder	Passende Bilder	Zurückgegebene Bilder	Precision	Recall
Suchmodell 1.1					
1	10	14	10	1,00	0,71
2	12	14	13	0,92	0,86
Suchmodell 1.2					
1	14	18	16	0,88	0,78
2	15	18	18	0,83	0,83
Suchmodell 1.3					
1	19	20	20	0,95	0,95
2	18	20	18	1,00	0,90

Die Unterschiede bei der Zahl der passenden Bilder sind darauf zurückzuführen, daß in den einzelnen Tests Ähnlichkeit unterschiedlich streng beurteilt wurde. Die Ergebnisse für die besten Suchmodelle sind außerdem in Abbildung 32 graphisch dargestellt.

3.2.3 Schwellwerte

Wie bereits oben angesprochen, ist ein besonderes Problem des CBIR (im Bereich der Multifeature-Ansätze) die Definition der Größe der Ergebnismenge. Diese wird üblicherweise

durch eine absolute Zahl von zurückzugebenden Bildern (N) angegeben. Für jedes Feature und alle Bilder in der Ausgangsmenge wird die Distanz zum Suchobjekt festgestellt, anschließend wird für jedes Bild eine gewichtete Distanzsumme errechnet und die Bildmenge nach dieser Summe gereiht. Sodann werden die ersten N Bilder als Suchergebnis zurückgegeben. Das bedeutet, daß der Ergebnisraum einer Suche (aufgespannt durch die Schichten des Suchmodells) ein n-dimensionales Ellipsoid ist, das alle Elemente der Ergebnismenge umhüllt. Dieser Sachverhalt ist für ein Suchmodell mit drei Features in Abbildung 28 dargestellt. Auf den Achsen werden die Distanzen der Bilder zum Suchbild für die Features im Suchmodell aufgetragen. Das Suchbild befindet sich im Ursprung.

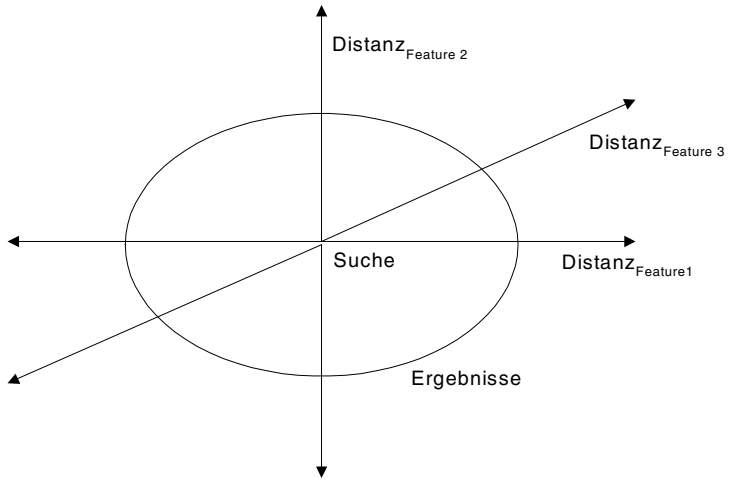

Abbildung 28: Definition der Größe der Ergebnismenge durch eine absolute Zahl

Die offensichtlichen Nachteile dieser Methode sind, daß die Größe des Ellipsoids von der Verteilung der Daten in der Ausgangsmenge abhängt und, in Bezug auf einige der betrachteten Eigenschaften, extreme Objekte (das sind am Rand des Ellipsoids liegende), die eigentlich unähnlich sind, zwangsläufig nicht abgeschnitten werden.

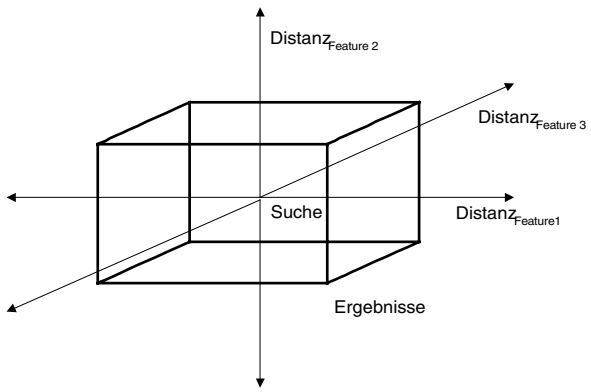

Abbildung 29: Definition der Größe der Ergebnismenge durch Schwellwerte

Deshalb wurde bei der Definition der Suchmodelle die Bestimmung der Größe einer Ergebnismenge auf die Ebene der Schichten verlagert, wo ein Schwellwert die maximale Distanz angibt, die ein Objekt, bezogen auf die betrachtete Bildeigenschaft, zum Suchobjekt

haben darf. Das heißt es wird im Gegensatz zu oben kein Ellipsoid sondern ein Quader aus der Ausgangsmenge herausgeschnitten (vgl. Abbildung 29).

Diese Vorgangsweise, bei der die Größe der Ergebnismenge unabhängig von der Gewichtung der Features ist, führt dazu, daß die Qualität der Ergebnismenge unabhängig von der Verteilung der Ausgangsmenge im Vektorraum wird. Da also bei dieser Methode nicht die Gefahr besteht, daß Objekte, die in bezug auf die meisten Eigenschaften dem Suchbild gut entsprechen, in einer oder wenigen aber stark abweichen, andere, die für alle Eigenschaften akzeptable Werte aufweisen, aus der Ergebnismenge verdrängen, führt die Verwendung von Schwellwerten vor allem zu einer Verbesserung des Recalls.

3.2.4 Verwendung von Suchmodellen mit Schwellwerten zur Suche nach Wappen

In diesem Abschnitt wird untersucht, wie sich Precision und Recall verändern, wenn man Schwellwerte zur Regulierung der Ergebnismenge benutzt. Die Vorgangsweise beim Testen ist dabei wieder dieselbe wie in Abschnitt 3.2.2. Die nachfolgende Tabelle zeigt vier zusätzliche Suchfragen, die neben den oben angegebenen untersucht wurden.

Nr.	Suchfrage	Bestes Suchmodell
2.1	Bilder mit einem großen schwarzen Objekt, in denen drei Tinkturen verwendet werden und die über keine Segmentierung verfügen.	Segmentierung, Siegelabdruck, Anzahl der Wappenfaben
2.2	Bunte, unsymmetrische Wappen mit möglichst vielen Objekten	Anzahl der Wappenfarben, Bild-Komplexität, Symmetrietyp
2.3	Einfache Bilder mit wenigen Farben und ohne heraldische Schnitte	Segmentierung, Anzahl der Wappenfarben, Bild-Komplexität
2.4	Schilde, in denen das bayrische Hoheitszeichen enthalten ist und die nicht die Tinktur Gold enthalten.	Siegelabdruck, Bedingtes Farbhistogramm

Abbildung 30 zeigt für jede dieser Suchfragen ein passendes Beispielbild.

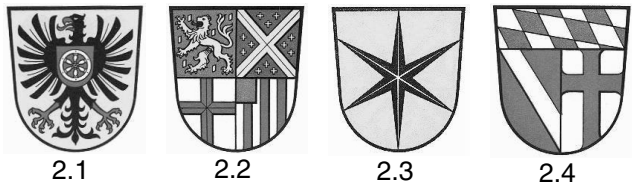

Abbildung 30: Beispielbilder für die Suchfragen 2.1 - 2.4

In der folgenden Abbildung sind für alle Suchfragen die Ergebnisse von jeweils vier Tests angegeben. Dabei zeigt sich, daß die Verwendung von Schwellwerten die Ergebnisqualität erhöht.

Test	Zurückgegebene passende Bilder	Passende Bilder	Zurückgegebene Bilder	Precision	Recall
Suchmodell 1.1					
1	12	14	12	1,00	0,86
2	13	14	13	1,00	0,93
Suchmodell 1.2					
1	16	18	18	0,89	0,89
2	16	18	18	0,89	0,89
Suchmodell 1.3					
1	17	20	19	0,89	0,85
2	18	20	20	0,90	0,90
Suchmodell 2.1					
1	8	9	9	0,89	0,89
2	7	9	8	0,88	0,78
Suchmodell 2.2					
1	23	25	25	0,92	0,92
2	22	25	25	0,88	0,88
Suchmodell 2.3					
1	18	20	20	0,90	0,90
2	17	20	20	0,85	0,85
Suchmodell 2.4					
1	9	10	10	0,90	0,90
2	9	10	9	1,00	0,90

Mit den besten Modellen konnte für alle Suchfragen ein Ergebnis von Precision, Recall größer 80 Prozent erzielt werden, teilweise auch im Bereich von mehr als 95 Prozent. Abbildung 31 gibt genau Auskunft über die erreichten Werte. Im Schnitt lagen die Ergebnisse bei ca. 90 Prozent, was selbst für künstliche Bilder und Features, die teilweise speziell für diesen Anwendungsbereich entwickelt wurden, ein gutes Ergebnis ist. Der besseren Ergebnisqualität durch die Verwendung von Schwellwerten steht aber der Nachteil gegenüber, daß diese für den Benutzer nur schwer festzulegen sind.

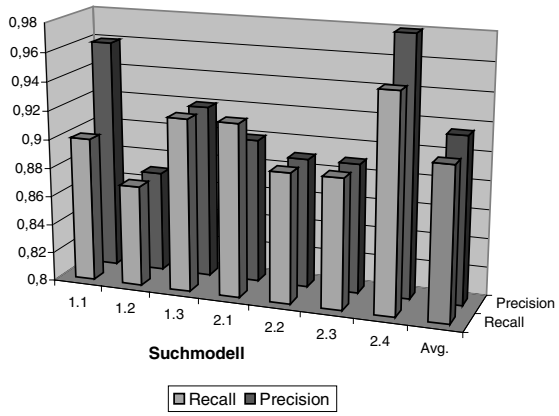

Abbildung 31: Suchergebnisse für die Wappen-Features

Das beste Ergebnis wurde mit einer Precision von 100 Prozent bei einem Recall von 94 Prozent für Suchfrage 2.4 erzielt, wo mit dem Siegel-Feature und dem bedingten Farbhistogramm zwei der zuverlässigsten Features verwendet wurden.

Bei diesen Versuchen wurden Precision und Recall als gleichwertig betrachtet; würde man den Fokus auf den Recall legen, ließen sich durch entsprechende Änderungen der Schwellwerte zweifellos noch bessere Recall-Werte erzielen. Abbildung 32 zeigt für die Suchfragen 1.1 bis 1.3 einen Vergleich der Schwellwert-Methode zur Methode, bei der die Größe einer Ergebnismenge durch die absolute Anzahl bestimmt wird. Für letztere liegen die Ergebnisse für den Recall bereits bei über 80 Prozent. Die Anwendung der Schwellwerte erhöht den Recall aber nochmals um durchschnittlich sieben Prozent, wobei die Precision ungefähr gleich bleibt.

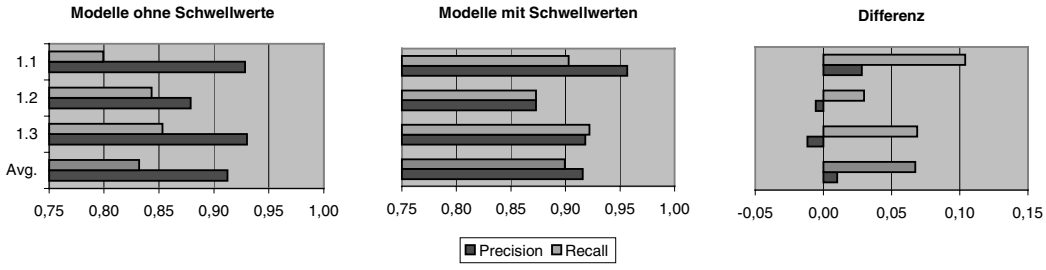

Abbildung 32: Verbesserung der Suchergebnisse durch Schwellwerte

Diese Ergebnisse lassen den Schluß zu, daß die Definition einer maximalen Abweichung für jede Bildeigenschaft ein weiteres Kriterium der Ähnlichkeitsdefinition ist. Alle Eigenschaften durch Merging zu verschmelzen und nur das Gesamtergebnis zu prüfen, führt zu einer signifikanten Verschlechterung der Ergebnisse. Man könnte argumentieren, daß die Festlegung von Schwellwerten für die maximale Abweichung eines Bildes (bezogen auf eine bestimmte Eigenschaft) eine Information ist, die die Genauigkeit der Suche erhöht und folglich vor allem positive Auswirkungen auf die Precision des Suchergebnisses haben sollte. Das läßt sich experimentell aber nicht belegen. Zu beobachten ist vielmehr der Effekt, daß

durch die Verwendung von Schwellwerten Bilder, die nur in wenigen Eigenschaften stark abweichen und sonst in der Ergebnismenge weit vorne gereiht werden würden, aussortiert werden und so andere, die in allen Eigenschaften die gestellten Anforderungen erfüllen, nicht mehr nach hinten und damit aus der Ergebnismenge verdrängt werden.

3.2.5 Testumgebung für Suchmodelle

3.2.5.1 Webschnittstelle

Abbildung 33 zeigt einen Screenshot der Testumgebung, die für die oben angegebenen Auswertungen verwendet wurde. Im oberen Bereich des Bildes (A) werden statistische Daten zur letzten Suche angegeben: welche Features verwendet wurden, die Anzahl der gefundenen Bilder in Relation zur Zahl der Bilder, die untersucht wurden und die Dauer des Berechnungsvorganges. Darunter werden die gefundenen Bilder angezeigt (B), wobei zu jedem Bild der Name, ein Link auf das Vollbild sowie die Distanzsumme (aufgrund derer die Reihung der Ergebnismenge erfolgt; C) angegeben wird. Außerdem werden die Einzeldistanzen (D), aus denen sich die Distanzsumme zusammensetzt, in der selben Reihenfolge wie die Features im Statistikbereich (A) angezeigt.

Darunter wiederum befindet sich die Suchschnittstelle (E). Es werden alle zur letzten Suche verwendeten Features inklusive aller Parameter (Schwellwert, Gewicht, etc.) angegeben. Diese Einstellungen können geändert werden, außerdem können Features hinzugefügt oder entfernt werden. Eine neue Suche wird durch das Anklicken eines der Bilder gestartet. Durch einen Button (F) kann die Suchmaske auf eine zufällige Auswahl von Bildern mit einem Standard-Suchmodell zurückgesetzt werden. Für Testzwecke wird unter der Suchschnittstelle in einer vierten Sektion die Ausgabe der Suchmaschine während des Suchvorganges ausgegeben (Featurevektoren, Zeitangaben, etc.).

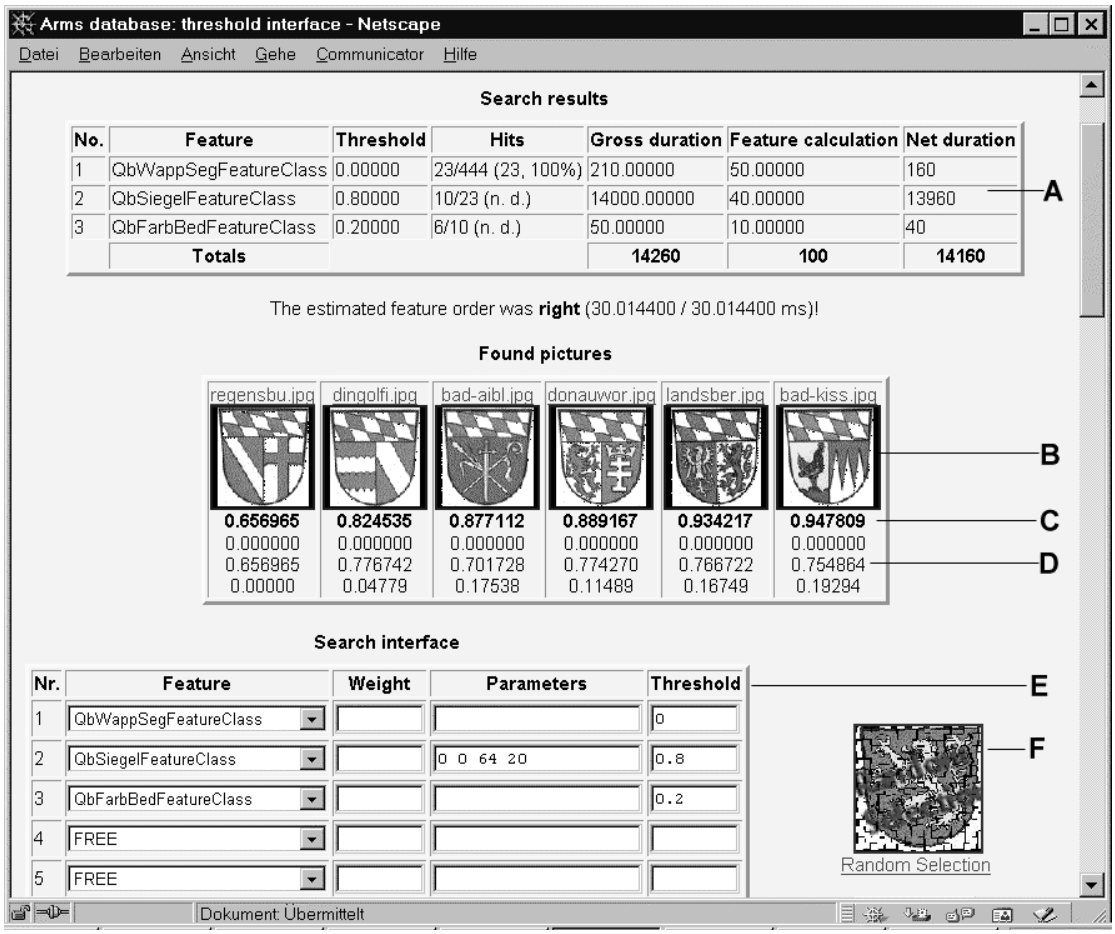

Abbildung 33: Screenshot der Webschnittstelle nach einer Suche

Dieses GUI bildet die Grundlage für alle Webschnittstellen, die in den folgenden Abschnitten vorgestellt werden. Es ist für den Experten gedacht und bietet daher alle Möglichkeiten, Parameter einzustellen, Statistiken zu erzeugen, etc. Für den Laien erscheint es aufgrund der Fülle der Eingabemöglichkeiten wenig geeignet, hier scheint die Verwendung einer einfachen Schnittstelle, die Standard-Suchmodelle (eventuell in Kombination mit iterativer Verfeinerung der Suche) vorgibt, sinnvoller (vgl. auch Kapitel 5).

3.2.5.2 Verwendete Suchmaschine

Im Rahmen dieser Arbeit wurde eine Suchmaschine (Name: NetSrv) entwickelt, die das Konzept der Suchmodelle für die Suche nach Bildbeispielen (Query by Example) umsetzt. Diese Suchmaschine erzeugt für jedes Feature eines Suchmodelles einen eigenen Suchstring und führt jede Abfrage nur für die Ergebnismenge der vorhergehenden Suche aus. Dadurch werden die meisten Suchvorgänge nur für einen geringen Bruchteil der Bilder in der Datenbank ausgeführt. Diese Verminderung der Anzahl der Distanzvergleiche führt zu einer Erhöhung der Performance, die durch andere Methoden (Nutzung der Dreiecksungleichung des geometrischen Ähnlichkeitsmodells, etc.) nicht erreicht werden könnte. Abbildung 34 zeigt exemplarisch den Ablauf einer Suche.

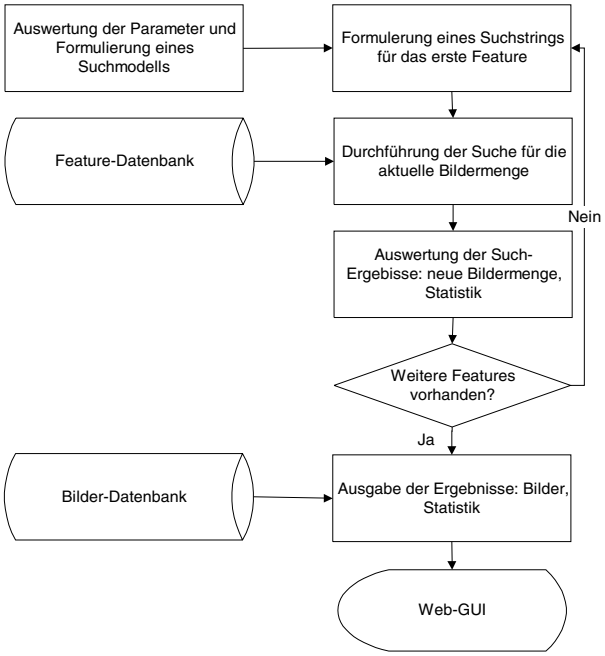

Abbildung 34: Ablauf einer Suchabfrage in der Suchmaschine

Zur Abarbeitung von Suchen werden Instanzen der im C++ - API von QBIC zur Verfügung gestellten Klassen benutzt:

- QbDatumClass: zum Speichern eines Suchstrings.

- QbConnectClass: zum Verbindungsaufbau zu einem Katalog einer QBIC-Bilddatenbank. Möglich ist das Aufbauen einer direkten Verbindung zu einem Katalog (in NetSrv verwendet) oder der Aufbau einer Server/Client-Verbindung über den QBIC-Datenbankserver (QbQrySrv).

- QbQueryClass: zur Durchführung einer Suche in einer offenen Verbindung. Dazu ist eine Suche erst zu formulieren (Methode: QbBuildQueryString()) und dann auszuführen (Methode: Evaluate())

- QbKeyDatabaseClass: zur Speicherung eines Suchergebnisses. Abgelegt wird jeweils ein Schlüssel (Bildname) und der Distanzwert für das betrachtete Feature. Die Einzelergebnisse sind durch eine einfach verkettete Liste verbunden.

- QbDbIteratorClass: zur Abarbeitung eines Suchergebnisses.

Grundlage für den NetSrv war die QBIC-Standard-Suchmaschine QbQBE, die um die Abarbeitung von Suchmodellen, Verwendung von Schwellwerten, Erzeugung von Statistik, etc. erweitert wurde.

3.2.6 Zusammenfassung

Es wurde ein Konzept entwickelt, das die Ähnlichkeitsdefinition im CBIR vereinheitlicht und erweitert. Es definiert Ähnlichkeit bezogen auf eine bestimmte Bildähnlichkeit mit einem Distanzmaß und einem Schwellwert für die maximale Unähnlichkeit zweier Objekte. Für dieses Konzept wurde eine passende Suchmaschine sowie eine webbasierte Suchschnittstelle implementiert, die zur Evaluierung der im Abschnitt 3.1 beschriebenen Features und Distanzfunktionen herangezogen wurde. Dabei zeigte sich, daß einerseits die Verwendung von Schwellwerten die Qualität von Suchergebnissen verbessert und andererseits die für Wappen implementierten Features für die verschiedensten Such-Szenarien gut geeignet sind. Dem Vorteil des besseren Recalls steht bei der Verwendung von Schwellwerten aber der Nachteil gegenüber, daß diese vor allem für Laien schwer zu bestimmen sind.

3.3 Analyse der Datenbasis

Dieser Abschnitt ist dem Datenmaterial gewidmet. Dabei werden weniger die Wappenbilder selbst betrachtet, als viel mehr die durch die Features erzeugten Datenvektoren. Zur Analyse der Objekte (Wappenbilder) und Variablen (Features) werden eine Clusteranalyse, Self-organizing Maps und eine Faktorenanalyse herangezogen. Dadurch können Zusammenhänge zwischen den untersuchten Einheiten gefunden und entsprechende Rückschlüsse gezogen werden, auf die dann in den folgenden Kapiteln immer wieder zurückgegriffen werden kann. Mit dem Begriff "Clusteranalyse" seien im folgenden nur die Methoden zur Gruppierung von Daten gekennzeichnet, die nicht auf der Theorie neuronaler Netze basieren. Self-organizing Maps dienen natürlich dem selben Zweck, werden in diesem Abschnitt aber gesondert behandelt.

In Abschnitt 3.3.1 werden zuerst die verwendeten Methoden vorgestellt, wobei der Fokus auf der Cluster- und der Faktorenanalyse liegt, da Self-organizing Maps bereits im Literaturüberblick behandelt wurden, Abschnitt 3.3.2 beschreibt kurz die durch die Features erzeugten Daten, während Abschnitt 3.3.3 die durchgeführten Tests sowie die erzielten Ergebnisse beschreibt.

3.3.1 Verwendete Methoden

3.3.1.1 Clusteranalyse

Die Clusteranalyse ([3], [22]) dient dazu, Mengen von Merkmalsvektoren in möglichst homogene, das heißt gleichartige Gruppen einzuteilen. Die Merkmalsvektoren können dabei die Eigenschaften von Objekten oder Zuständen, etc. repräsentieren. Im folgenden sei die klassische Form der Clusteranalyse anhand ihrer wesentlichen Schritte skizzenhaft beschrieben (vgl. [22]):

- Stichprobenauswahl & Wahl der Variablen: diese beiden Punkte spielen für den Anwendungszweck in dieser Arbeit keine besondere Rolle. Es werden alle Bilder der Testdatenbank untersucht und dabei möglichst alle Features genützt.

- Homogenisierung der Variablen: alle Variablen müssen auf den selben Wertebereich skaliert werden. Zu beachten ist, daß die Variablen zwar einen beliebigen Skalentyp (Nominal-, Ordinal-, Intervall oder Ratioskala) als Grundlage haben dürfen, bei der Skalierung der Variablen aber darauf Rücksicht genommen werden muß.

- Wahl eines Ähnlichkeitsmaßes: das Ähnlichkeitsmaß (vgl. auch Literaturüberblick) dient zur Quantifizierung der Ähnlichkeit von Merkmalen. Um Datenvektoren in homogene Gruppen einteilen zu können, muß normalerweise auch eine Aussage über ihre Ähnlichkeit getroffen werden.

- Wahl eines Cluster-Kriteriums: durch das Cluster-Kriterium wird bestimmt, wie die Cluster gebildet werden. Im folgenden werden einige Algorithmen zur Bildung einer Clusterstruktur vorgestellt. Die Verwendung eines Algorithmus zur Clusterung ist aufgrund des enormen Rechenaufwandes für den sonst notwendigen Vergleich jedes Vektors der Ausgangsmenge mit jedem anderen unmittelbar verständlich.

- Bestimmung der Zahl der Cluster: bei den Verfahren zur Clusteranalyse ist grundsätzlich zwischen hierarchischen und partitionierenden (nicht-hierarchischen) zu unterscheiden. Bei letzteren muß die Zahl der gewünschten Cluster angegeben werden, bei ersteren nicht.

- Interpretation der Ergebnisse: an den Ergebnissen einer Clusteranalyse ist neben den eigentlichen Gruppen vor allem auch das Ausmaß der Unterschiede zwischen den Gruppen interessant. Es muß aber beachtet werden, daß das gewählte Cluster-Kriterium wesentlichen Einfluß auf die Bildung der Cluster hat und daher, wenn möglich, mehrere Clusterungen mit unterschiedlichen Kriterien vorgenommen werden sollten.

Hierarchische Verfahren lassen sich in zwei Gruppen einteilen ([22]):

- Agglomerative Verfahren versuchen, von der feinsten Partitionierung ausgehend (wobei jedes Objekt einem Cluster entspricht), durch Zusammenfassung größere Cluster zu bilden.

- Divisive Verfahren gehen umgekehrt vor: ausgehend von der gröbsten Partitionierung (alle Objekte gehören zu einem Cluster) werden solange Gruppen gebildet, bis wiederum die feinste Partitionierung erreicht ist.

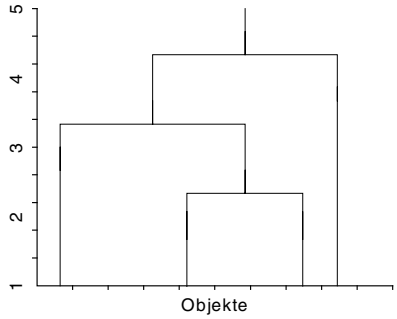

Abbildung 35: Beispiel-Dendrogramm

Die durch eine hierarchische Clusteranalyse gefundene Struktur läßt sich durch ein sogenanntes Dendrogramm darstellen (vgl. Abbildung 35). Dabei werden auf der X-Achse die untersuchten Objekte und auf der Y-Achse die Distanz (Ähnlichkeit) aufgetragen. Bei maximaler Distanz findet man die gröbste Partitionierung, bei minimaler Distanz die feinste. Die Distanz zwischen zwei Gruppen (das sind Partitionierungen auf mittlerer Ebene) kann

durch die Subtraktion des kleineren Distanzwertes vom größeren berechnet werden. Agglomerative Verfahren arbeiten folgendermaßen:

1. Ausgegangen wird von der feinsten Partition

2. Fusion der beiden Cluster mit dem minimalen Abstand

3. Neuberechnung des Abstandes für den neuen Cluster zu den übrigen

4. Falls mehr als ein Cluster existiert: wiederholen ab Schritt 2

Agglomerative Verfahren unterscheiden sich lediglich nach der Methode, mit der in Schritt 3 die Distanz neu berechnet wird. Bekannte Verfahren sind: Nearest Neighbour, Furthest Neighbour, Average-Linkage Verfahren, Centroid-Verfahren, etc. Beim (in den folgenden Auswertungen verwendeten) Nearest Neighbour Verfahren wird die Distanz des neuen Clusters gemäß Gleichung 25 berechnet:

$$d_{neu} = \min(d_{C1}, d_{C2}) \tag{25}$$

Hier sind d_{neu} die Distanz des neuen Clusters und d_{C1} beziehungsweise d_{C2} die Distanzen der Ausgangscluster. Es wird also als Distanzwert immer die minimale Teildistanz genommen. Das Nearest Neighbour (oder: Single-Linkage) Verfahren neigt zur Kettenbildung, wobei auf jeder Distanzstufe d_h alle Objekte zu einem Cluster fusioniert werden, deren Distanz zu mindestens einem Objekt nicht größer als d_h ist. Für die im folgenden beschriebenen Auswertungen durch hierarchische Clusteranalyse wurde das Nearest Neighbour Verfahrens angewendet, wobei als Distanzmaß die euklidische Distanz verwendet wurde.

Von den divisiven Verfahren der hierarchischen Clusteranalyse sei hier aufgrund ihrer geringen Bedeutung nur das Verfahren nach MacQueen erwähnt, bei dem zu zerlegende Cluster aufgrund der maximalen Abstandsquadratsumme (Varianz-Kriterium) ausgewählt werden (vgl. [3]).

Algorithmen zur Durchführung einer partitionierenden Clusteranalyse müssen eine bestimmte Menge von m Objekten zu n Clustern zuordnen. Dazu wird entweder eine Zielfunktion verwendet oder heuristisch vorgegangen. Von den heuristischen Verfahren sei hier der Leader-Algorithmus erklärt ([22]):

1. Das erste Objekt wird zum sogenannten Klassenführer des ersten Clusters.

2. Alle anderen Objekte werden dem Cluster zugeordnet, zu dessen Klassenführer ihr Abstand kleiner als ρist.

3. Wenn kein solcher Cluster existiert, wird das betrachtete Objekt Klassenführer eines neuen Clusters.

4. Der Algorithmus endet, wenn alle Objekte durchgearbeitet wurden. Nicht zugeordnete Objekte bleiben unbehandelt.

Der Parameter ρ ist frei wählbar; das Ergebnis hängt stark von der Reihenfolge der zu clusternden Objekte ab. Typische Zielfunktionen nicht-heuristischer Verfahren zur Partitionierung sind das Varianzkriterium und das Determinantenkriterium.

3.3.1.2 Self-organizing Maps

Selbstorganisierende Karten (SOMs, [44]) wurden bereits im Literaturüberblick beschrieben. Es handelt sich um zweischichtige, vollständig verknüpfte neuronale Netze, die durch einen Algorithmus aus drei Schritten hochdimensionale Eingabedaten in einer zweidimensionalen Map anordnen. Die Adaption der Gewichte erfolgt dabei unüberwacht. Im folgenden sollen einige Begriffe erklärt werden, die zum Verständnis der SOM-Implementierung SOM-PAK notwendig sind.

Die Anpassung der Mittelwertvektoren, nachdem für einen Inputvektor ein gewinnender Knoten gefunden worden ist, erfolgt aufgrund folgender Gleichung:

$$m_i^{neu} = m_i^{alt} + h_{ci}\left|x - m_i^{alt}\right| \tag{26}$$

Hierbei sind x der Inputvektor und m_i die Mittelwertvektoren. Die Anpassung erfolgt für alle Vektoren. h_{ci} wird als neighbourhood kernel bezeichnet. Diese Funktion dient dazu, die Veränderung, die sich durch die Zuordnung eines Vektors zu einem Knoten ergibt, auch für die zweidimensionale Umgebung des gewinnenden Knotens sichtbar zu machen. Dadurch wird die für SOMs charakteristische Erhaltung der Ordnung im Raum erzielt. In SOM-PAK werden folgende neighbourhood kernels unterschieden ([24]):

- Die bubble Funktion ist eine monoton fallende Funktion der Zeit. Das heißt je länger die Lernphase dauert, um so weniger Änderungen werden gemacht. Während das Lernen zu Beginn also zu einer sehr elastischen Map führt, wird sie im Laufe der Zeit immer statischer. Allerdings lernen überhaupt nur jene Punkte, die sich innerhalb eines bestimmten Radius zum gewinnenden Knoten befinden, alle anderen (außerhalb der bubble) werden nicht angepaßt.

- Gauß-Funktion: hier wird die Lernrate aufgrund einer negativ-exponentiellen Funktion, die als Parameter unter anderem den Abstand eines Knotens vom gewinnenden Knoten verwendet, bestimmt. Die Autoren von SOM-PAK empfehlen diese Funktion für die Verwendung in großen Maps.

Nachdem eine Map berechnet wurde, kann durch den sogenannten mittleren Quantifizierungsfehler ihre Qualität bestimmt werden. Dazu werden nochmals alle Eingabevektoren mit ihren gewinnenden Knoten verglichen und der Mittelwert über die Fehlerdistanz gebildet. In Gleichung 27 sind die x_i die Eingabevektoren und m_c ist der jeweilige gewinnende Knoten.

$$qe = \frac{\left\| x_i - m_c \right\|}{i} \tag{27}$$

3.3.1.3 Faktorenanalyse

Im Gegensatz zu den bisher dargestellten Verfahren dient die Faktorenanalyse nicht zur Gruppierung von Datenvektoren. Vielmehr versucht man mit diesem multivariaten statistischen Verfahren, Abhängigkeiten zwischen Variablen durch wenige, gemeinsame Faktoren zu erklären ([50]). Dabei wird davon ausgegangen, daß der Zusammenhang zwischen zwei Variablen stets durch eine oder mehrere andere, unabhängige Variablen (die sogenannten Faktoren) erklärt wird. Die Ziele der Faktorenanalyse lassen sich folgendermaßen zusammenfassen ([50]):

- Ermittlung der Anzahl voneinander unabhängiger Dimensionen (Faktoren), auf die sich eine vieldimensionale Datenmenge reduzieren läßt.

- Datenreduktion durch die Verwendung der Faktoren anstelle der Variablenwerte.

- Feststellung, welcher Faktor durch welche Variablen charakterisiert wird. Variablen, die durch dieselben Faktoren bestimmt werden, lassen sich zu Gruppen zusammenfassen.

- Trennung von durch die Faktoren bestimmten gegenüber von spezifischen Varianzanteilen der Variablen. Der spezifische Varianzanteil einer Variablen kann als charakteristisch für die Variable oder als Fehlerkomponente erklärt werden.

Dem Verfahren der Faktorenanalyse liegen folgende Annahmen zugrunde ([50]):

- Die Variablen sollen zumindest auf einer Intervallskala gemessen werden.

- Die Variablenwerte sollen normalverteilt sein.

Da die Variablen als Linearkombinationen der Faktoren gebildet werden und umgekehrt, werden durch dieses Verfahren nur lineare Zusammenhänge zwischen Variablen erklärt. In den beiden folgenden Abschnitten wird die Vorgangsweise bei der Faktorenanalyse sowie bei der Interpretation der Ergebnisse skizziert.

3.3.1.3.1 Vorgangsweise

Ziel der Faktorenanalyse ist es, die gegebenen Variablen durch Linearkombinationen von Faktoren auszudrücken (Annahme der Faktorenanalyse; [50]):

$$Z = L \,.\, F \tag{28}$$

Dabei ist Z die Variablenmatrix, F die Matrix der Faktorenwerte und L die Matrix der Koeffizienten (Faktorladungen). Ausgangspunkt ist die Beobachtungsmatrix X, aus deren Elementen durch folgende Umformung die standardisierte Variablenmatrix Z gewonnen werden kann:

$$z_{i,j} = \frac{x_{i,j} - m_j}{s_j} \tag{29}$$

Hierbei sind $x_{i,j}$ und $z_{i,j}$ die Elemente der Matrizen X und Z, m_j ist der Mittelwert über alle Ausprägungen der Variable j und s_j ist die Standardabweichung der j-ten Variable. In der Matrix Z ist der Mittelwert jeder Variablen immer 0 und die Varianz stets 1. Aus der Variablenmatrix läßt sich durch folgende Umformung die für die Faktorenanalyse benötigte Korrelationsmatrix der Variablen R gewinnen (wobei n die Anzahl der Variablen ist):

$$R = \frac{1}{n} . Z^T . Z \tag{30}$$

Die Matrix R ist quadratisch und symmetrisch. Setzt man Gleichung 28 in diese Gleichung ein, so ergibt sich nach einigen Umformungen und unter der Annahme, daß die Faktoren unabhängig sind, das Fundamentaltheorem der Faktorenanalyse ([14], [50]):

$$R = L . L^T \tag{31}$$

Ziel der Faktorenanalyse ist es, zu einer gegebenen Korrelationsmatrix R eine Ladungsmatrix L zu finden, die multipliziert mit ihrer transponierten wieder R ergibt. Dazu wird folgendermaßen vorgegangen ([50]):

1. Schätzung der Kommunalitäten: die Kommunalität einer Variablen ist der Beitrag, den alle Faktoren zur Erklärung ihrer Varianz leisten. Die gesamte Varianz einer Variablen besteht aus der Kommunalität, der Spezifität und der Restvarianz (Meßfehler, etc.). Ziel des ersten Schrittes ist es, die Diagonalelemente von R (Selbstkorrelationen der Variablen) durch geschätzte Kommunalitäten zu ersetzen.

   Dazu werden vor allem folgende Schätzverfahren angewendet ([50]):

   - Methode des multiplen Bestimmtheitsmaßes. Hierbei wird als Kommunalität die gemeinsame Varianz zwischen einer Variable und allen anderen verwendet. Dieses

Verfahren ist unabhängig von der Zahl zu bestimmender Faktoren, bedingt aber einen hohen Berechnungsaufwand.

- Methode des höchsten Korrelationskoeffizienten je Zeile. Hierbei wird als Kommunalität die höchste Kovarianz einer Variablen mit einer anderen herangezogen. Dem geringen Berechnungsaufwand dieser Methode steht als Nachteil ihre mangelhafte theoretische Begründung gegenüber.

- Iteratives Verfahren. Hierbei wird mit groben Schätzungen der Kommunalitäten begonnen (z. B. konstant 1), die dann iterativ durch Faktorenanalysen verfeinert werden, bis die Differenz der Kommunalitäten vor und nach einem Durchlauf minimal ist. Dieses Verfahren ist theoretisch nicht exakt begründet, abhängig von der Zahl der Faktoren und sehr rechenintensiv. Es wird aber in der Praxis weitaus am häufigsten verwendet und wurde auch für die folgenden Auswertungen angewandt.

2. Faktorenextraktion: in diesem Schritt wird zuerst die Zahl der zu extrahierenden Faktoren bestimmt und anschließend die Matrix der Faktorladungen gebildet. Bei der Bestimmung der Zahl der Faktoren geht es im Grunde darum, den Rang der Korrelationsmatrix R zu bestimmen. Die Zahl der Faktoren entspricht der Anzahl der linear unabhängigen Variablen. Zur Bestimmung der Anzahl der Faktoren werden vor allem folgende Kriterien verwendet:

   - Eigenwert- oder Kaiser-Kriterium: es werden alle Faktoren mit einem Eigenwert größer oder gleich 1 extrahiert. Dabei mißt der Eigenwert den Beitrag eines Faktors zur Erklärung der gesamten Varianz. Da in der Variablenmatrix Z die Varianz jeder Variable 1 ist, werden so nur Faktoren extrahiert, deren Erklärungsanteil größer oder gleich dem der Ausgangsvariablen ist. Dieses Kriterium wird in den folgenden Auswertungen verwendet.

   - Rippe-Test: hierbei wird nach jeder Bestimmung eines Faktors anhand des Fundamentaltheorems der Faktorenanalyse der Grad der Reproduktion der Matrix R überprüft.

   - Gesamtkommunalität: hierbei werden alle Faktoren extrahiert, die einen ausreichend großen Anteil der Gesamtkommunalität erklären. Wann der Anteil ausreichend ist, muß vom Faktorenanalytiker bestimmt werden.

   Nachdem die Zahl der Faktoren bestimmt wurde, müssen die Faktorladungen bestimmt werden. Dazu wird meist eines der beiden folgenden Verfahren angewandt:

   - Zentroidmethode: hierbei wird versucht, die Koordinatenachsen (Ladungsvektoren) so durch die vieldimensionale Punktwolke der Faktorenwerte zu legen, daß die erste Achse durch den Schwerpunkt des Gesamtraums geht. Die zweite Achse geht durch

den Schwerpunkt des um den ersten Faktor reduzierten Unterraums, etc. Dieses Verfahren wird aufgrund der einfachen Durchführbarkeit häufig angewendet.

- Hauptachsenmethode: hier wird von der elliptischen Hülle der Punktwolke der Faktorenwerte ausgegangen und versucht, die erste Achse so durch die Hülle zu legen, daß die Fläche des Durchschnitts maximal ist. Die erste Achse (der erste Faktor) soll also ein Maximum der Gesamtvarianz erklären. Bei den weiteren Achsen wird analog vorgegangen. Dieses Verfahren wird auch in den folgenden Auswertungen verwendet.

3. Faktorenrotation: dieser Vorgang dient dazu, die ermittelten Faktoren interpretationsfähig zu machen, da die durch die Extraktion gefundene Ladungsmatrix oft wenig aussagekräftig ist. Grundsätzlich gibt es für die Ladungsmatrix L des Fundamentaltheorems unendlich viele Lösungen. Die Rotation erfolgt, indem die Ladungsmatrix mit einer Transformationsmatrix T multipliziert wird. Die Matrix T ist quadratisch, die Elemente der Hauptdiagonale sind 1 und alle anderen außer bei den zu rotierenden Faktoren sind 0. Außerdem muß das Produkt von T mit ihrer transponierten Matrix die Einheitsmatrix ergeben, damit die Orthogonalitätsforderung erfüllt ist.

   Die häufigste Rotationsmethode ist die Varimax-Methode ([14]). Sie dient dazu, die Spalten der Ladungsmatrix zu vereinfachen. Das wird durch Maximierung der Streuungen der Ladungen über die Variablen erreicht, sodaß die Faktorladungen entweder nahe 0 oder 1 sind. Die Varimax-Methode wird für die folgenden Auswertungen benutzt.

3.3.1.3.2 Interpretation der Ergebnisse

Das wesentliche Element der Interpretation der Ergebnisse einer Faktorenanalyse ist die Untersuchung der Faktormatrix und die Deutung der Faktoren. Dazu müssen je Faktor die Variablen mit einem hohen (Absolutwert größer 0.5) beziehungsweise sehr niedrigen (Absolutwert kleiner 0.1) Ladungsanteil gefunden werden. Aus den Variablen mit markanten Ladungszahlen läßt sich eine intuitive Deutung der Faktoren ableiten. Als Anhaltspunkte für die Interpretation der Ergebnisse dienen:

- Die Anzahl der extrahierten Faktoren
- Der Anteil der erklärten Varianz je Faktor.
- Der Anteil der erklärten Varianz durch alle gefundenen Faktoren.
- Die Höhe sowie das Vorzeichen der Ladungszahlen der Variablen je Faktor.
- Die Kommunalitäten der Variablen.

Ziel des Interpretationsvorganges ist es, den Einfluß der gewählten Methode zur Faktorenanalyse auf die Ergebnisse zu minimieren.

3.3.2 Daten

Für die im folgenden beschriebenen Auswertungen wurden 16 der 19 vorgestellten Features herangezogen, woraus sich für jedes Objekt (Bild) ein Featurevektor mit 58 Stellen ergibt. Auf das bedingte Farbhistogramm, das Siegel- und das Objekt-Layout-Feature mußte aus den im Abschnitt 4.1.2 angegebenen Gründen verzichtet werden. Es wurden 444 Bilder untersucht, wobei die Featurevektoren folgenden Aufbau haben:

Stelle(n)	Feature
1-6	Bild-Kompexität
7	Symmetrietyp
8	Diagonale Symmetrie mit negativem Anstieg der Symmetrieachse
9	Diagonale Symmetrie mit positivem Anstieg der Symmetrieachse
10	Inverse X-Achsen-Symmetrie
11	Inverse Y-Achsen-Symmetrie
12	X- und Y-Achsen-Symmetrie
13	X-Achsen-Symmetrie
14	Y-Achsen-Symmetrie
15	Anzahl der Wappenfarben
16-21	Wappen-Farbhistogramm
22	Anzahl der Farbabstufungen
23-27	Regionalisierte Anzahl der Wappenfarben
28-52	Regionalisiertes Farbhistogramm
53-57	Regionalisierte Anzahl der Farbabstufungen
58	Segmentierung

Für die Auswertungen wurden die (recht unterschiedlichen) Elemente des Featurevektors auf den Wertebereich [0,10] normiert, wobei rationale Zahlen mit maximal drei Dezimalstellen verwendet wurden. Die folgende Tabelle zeigt einen Ausschnitt aus den Daten (vor der Skalierung):

Objekte	Linien	m(Länge)	v(Länge)	m(Winkel)	v(Winkel)	SymTyp	Sym \	Sym /	SymInvX
41	421	12,11	3889,02	106	32428,54	0	2608	2609	8205
53	451	10,19	3610,98	103,77	35095,59	0	2966	2976	6898

Objekte	Linien	m(Länge)	v(Länge)	m(Winkel)	v(Winkel)	SymTyp	Sym \	Sym /	SymInvX
36	326	10,76	2652,5	112,49	27156,58	0	2353	2353	7207
47	417	9,88	3191,03	104,43	32217,83	0	2600	2574	8026
22	140	12,5	1332,37	121,71	12962,81	0	1862	1936	7823
39	261	13,39	2867,84	109,41	21923,88	12	4017	4008	8090
12	293	12,68	3131,88	117,65	26718,1	2	3067	3063	7293
27	334	11,09	2720,91	113,07	27146,88	0	2383	2380	8025
106	356	8,14	2129,04	106,81	27028,5	0	2062	2105	7128
57	252	14,19	3014,45	110,26	21881,04	0	2063	2081	8088
60	310	10,57	2496,57	97,66	20707,67	0	1811	1698	8858
76	380	11,89	3287,54	104,58	28940,43	0	3105	3078	7933
58	365	10,17	2986,09	103,82	29193,94	0	2988	3027	8190
31	476	12,3	4273,76	111,91	37830	0	2961	2940	9061
84	225	9,46	1763,44	100,49	17761,64	0	2055	1977	7929
13	139	17,32	2016,07	115,95	13614,52	0	2526	2535	7325
50	207	13,04	2053,43	91,87	13374,94	0	2595	2546	7634
20	263	11,71	2231,29	107,39	20318,7	0	2115	2067	6723
9	201	10,37	1833,38	121,74	21067,55	0	1680	1692	8174
105	223	8,5	1508,69	87,01	14572,26	0	1409	1410	7149
179	276	6,69	1500,85	72,2	16368,14	0	2513	2557	8888
95	230	11,45	2135,58	91,66	16984,73	0	1814	1815	6554
39	178	15,46	2189,65	91,41	11563,38	12	4061	4051	6647
35	244	13,1	2291,04	99,82	15978,81	0	1913	1971	7697
71	388	10,66	3381,53	102,75	30954,85	2	3765	3777	8029

In den Spalten sind jeweils die Variablen, in den Zeilen die Objekte dargestellt; gezeigt sind die ersten zehn Stellen des Featurevektors für die ersten 25 Objekte. Daraus wird ersichtlich, daß die verwendeten Daten sich unter anderem in den folgenden Eigenschaften unterscheiden:

- Unterschiedliche Skalen: während z. B. der Symmetrietyp (Spalte 7 der obigen Tabelle) einen Wert auf einer nominalen Skala mißt, werden die Elemente des Bild-Komplexitäts-Features (Spalte 1-6 der Tabelle) auf einer Intervallskala gemessen.

- Unterschiedliche Variablentypen: die meisten Elemente werden als ganze Zahlen gemessen, manche aber auch als reelle Zahlen. Dabei kommt den Kommastellen, abhängig von der Varianz einer Variablen, unterschiedliche Bedeutung zu.

- Unterschiedliche Wertebereiche: vor der Skalierung waren die Daten auf Intervallen zwischen [0, 1], [1, 22], etc. bis zu [0, 65535] definiert.

3.3.3 Auswertungen

Im folgenden werden die Auswertungen dargestellt, die mit den oben angegebenen Methoden am gegebenen Datenmaterial unternommen wurden sowie die erzielten Ergebnisse. Es wurden sowohl die Objekte (Bilder) als auch die Variablen (Elemente der Featurevektoren) untersucht. Die Objekte entsprechen in der obigen Datentabelle den Zeilen, die Variablen den Spalten.

3.3.3.1 Bilder (Objekte)

Die Untersuchung der Clusterung des Bildmaterials erfolgte einerseits mithilfe einer hierarchischen Clusteranalyse und andererseits partitionierend mit dem SOM-Algorithmus. Im wesentlichen ging es darum festzustellen, ob für die Suchmodell-Hypothese überhaupt passende Daten vorliegen. Bei der Begründung der Verwendung von Suchmodellen wurde ja damit argumentiert, daß eine Datenmenge auf mittlerer Betrachtungsebene aus Gruppen von homogenen Teilmengen besteht, die mehrheitlich unterschiedliche Eigenschaften aufweisen. Deshalb sollte die Suche nach einzelnen dieser Teilmengen am besten spezialisiert, das heißt mit Suchmodellen, erfolgen (Clusterhypothese). Die hierarchische Clusteranalyse wurde mithilfe des Statistikprogramms SPSS und folgenden Parametern durchgeführt:

Parameter	Wert
Clustermethode	Verlinkung innerhalb der Gruppen (Single-Linkage Verfahren)
Distanzmaß	Quadrat der euklidischen Distanz
Werteanpassung	Skalierung aller Elemente auf [0, 1]

Das Ergebnis-Dendrogramm kann aufgrund seiner Größe hier leider nicht dargestellt werden. Es bestätigt aber recht gut die Clusterhypothese, da auf mittlerer Distanzebene viele, in etwa gleich große Cluster bestehen, bei denen es aufgrund ihrer spezifischen Eigenschaften sinnvoll ist, individuell nach ihnen zu suchen. Als partitionierendes Verfahren wurde, wie bereits erwähnt, eine SOM verwendet. Für die Clusterung der Vektoren wurde SOM-PAK benutzt, wobei die erstellte Map folgende Eigenschaften aufweist:

Parameter	Wert
Map-Layout	hexagonal
Dimensionen	8 x 6 bins (bei 444 Bildern)
Neighbourhood kernel	bubble

Da die Map sehr klein ist, wurde dem Handbuch von SOM-PAK gefolgt und als Neighbourhood-Kernel die "bubble"-Funktion (anstelle der Gauß-Funktion) verwendet. Während der Trainingssession wurden (auf einem Digital Alpha-Server mit vier CPUs) 50 verschiedene Tests mit jeweils 550000 Lernschritten durchgeführt. Dabei ergab sich für die beste Map ein minimaler Quantifizierungsfehler von 7.467.

Es zeigt sich, daß einige Features, die mit wenigen Vektorelementen grundlegende Aussagen über die Eigenschaften eines Bildes treffen, einen sehr großen Einfluß auf die Clusterung einer Bildmenge haben (z. B. Segmentierung, Symmetrietyp, etc.). An dieser Stelle sei auch noch erwähnt, daß verschiedene Versuche unternommen wurden, die Bildmenge direkt, das heißt aufgrund der Farbwerte der einzelnen Pixel, zu clustern. Bei zwei Versuchen mit standardisierten Grauwert- beziehungsweise Farbbildern konnte durch eine SOM aber keine sinnvolle, durch den Menschen nachvollziehbare Partitionierung gefunden werden.

3.3.3.2 Features (Variablen)

Die Elemente der Featurevektoren wurden anhand einer hierarchischen Clusteranalyse sowie durch eine Faktorenanalyse untersucht.

3.3.3.2.1 Untersuchung der Features durch eine hierarchische Clusteranalyse

Für die Clusteranalyse wurden dieselben Einstellungen wie oben gewählt. In Abbildung 36 ist ein Ausschnitt aus dem Ergebnisdendrogramm abgebildet. Zu beachten ist, daß das Dendrogramm gedreht ist und die Distanz auf der X-Achse abgebildet ist, während die Variablen auf der Y-Achse aufgetragen wurden. Die Variablennamen verweisen jeweils auf eine Stelle im Gesamtfeaturevektor.

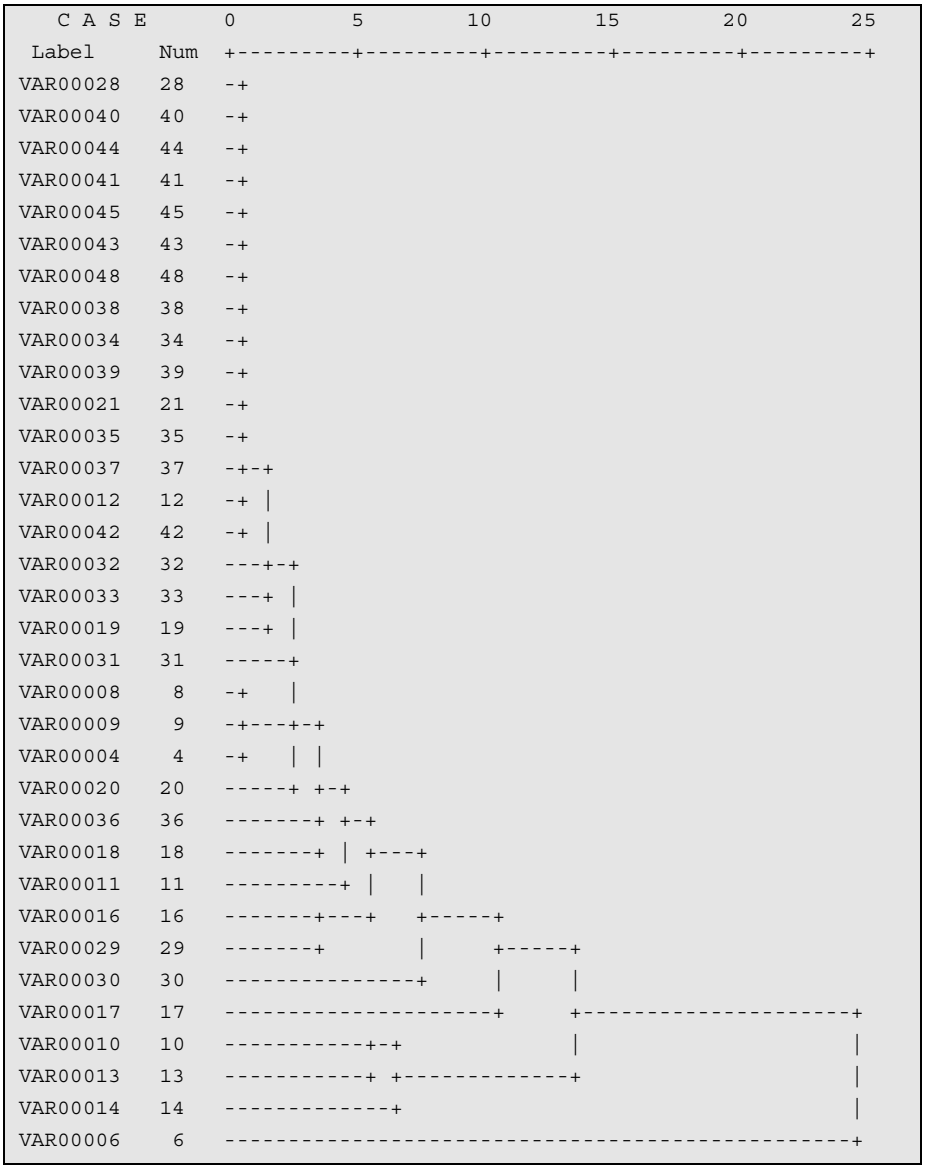

Abbildung 36: Dendrogramm der Featureelemente

Zwischen den Features konnten unter anderem folgende Zusammenhänge festgestellt werden:

- Die Features für die diagonalen Symmetrien messen sehr ähnliche Eigenschaften. Ist ein Bild in einer Diagonale symmetrisch, so meist auch in der anderen (vgl. Abbildung 37). Dieser Zusammenhang wurde im folgenden für verschiedene Algorithmen genutzt.

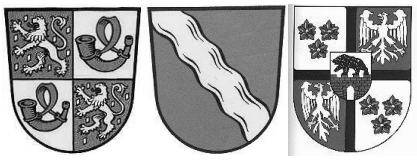

Abbildung 37: Diagonal symmetrische Bilder

- Es ergab sich ein hoher Zusammenhang zwischen dem Farbhistogramm und der ersten Region des regionalisierten Farbhistogramms, was wenig verwundert, wenn man bedenkt, daß fast die Hälfte aller Bilder unsegmentiert sind.

- Die Feature-Paare X-Achsen-Symmetrie und inverse X-Achsen-Symmetrie beziehungsweise Y-Achsen-Symmetrie und inverse Y-Achsen-Symmetrie erwiesen sich als sehr gegensätzlich, was beim Entwurf dieser Features auch gewünscht war.

- Nicht ergeben hat sich ein (zu erwartender) hoher Zusammenhang zwischen dem Feature für die Anzahl der Farbabstufungen und dem Feature für die Anzahl der Wappenfarben, die ja beide Farben beziehungsweise Farbabstufungen zählen. Da ein Feature mit vielen Wappenfarben aber aufgrund der heraldischen Regeln für die Farbverwendung nicht notwendigerweise viele Farbabstufungen enthält und umgekehrt, besteht ein solcher Zusammenhang nicht.

- Schließlich zeigte sich ein großer Gegensatz zwischen dem sechsten Element des Vektors des Komplexitäts-Features (Varianz der Winkel zwischen den Kanten eines Bildes) und allen anderen Eigenschaften. Mit der Varianz der Winkel zwischen den Kanten in einem Bild wird also etwas völlig anderes gemessen als durch alle anderen Features.

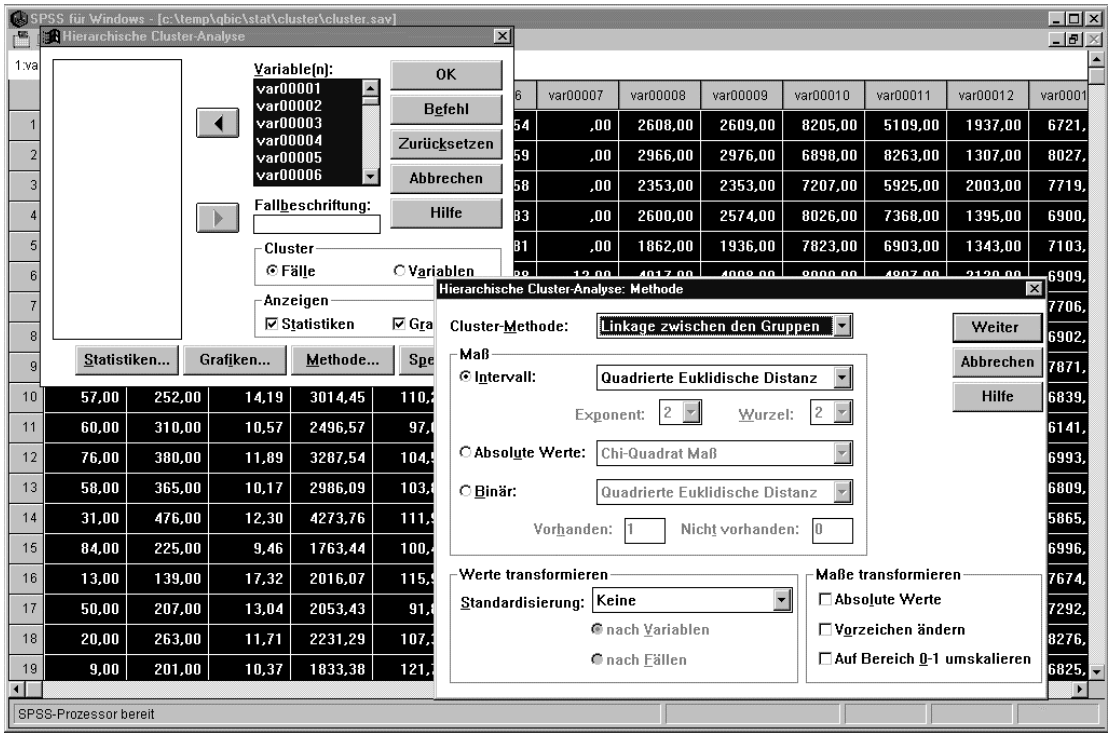

Abbildung 38: Screenshot des Statistikprogramms SPSS

Neben der hierarchischen Analyse der Featurevektoren wurde auch eine Partitionierung mit SOMs vorgenommen, wobei die Ergebnisse der hierarchischen Analyse bestätigt wurden.

Abbildung 38 zeigt einen Screenshot des für die hierarchischen Clusteranalysen verwendeten Statistikpakets SPSS.

Im Hauptfenster sieht man die noch nicht normalisierten Eingabedaten, die linke obere Dialogbox ist die Hauptauswahl für die Durchführung von Clusteranalysen, wobei entweder Analysen für Objekte (Fälle) oder Variablen gewählt werden können. Im rechten unteren Fenster kann ein Clusterkriterium und ein Distanzmaß ausgewählt werden.

3.3.3.2.2 Faktorenanalyse der Features

Für die Faktorenanalyse (ebenfalls mit SPSS durchgeführt) wurden folgende Verfahren gewählt:

- Iterative Bestimmung der Kommunalitäten
- Bestimmung der Anzahl der Faktoren durch das Eigenwert-Kriterium
- Bildung der Ladungsmatrix mit der Hauptachsenmethode
- Rotation der Ladungsmatrix durch die Varimax-Methode.

Es wurden 17 Faktoren gefunden, die einen Eigenwert größer als 1 haben. Diese Faktoren erklären ca. 83 Prozent der Varianz der Variablen. Die folgende Tabelle zeigt die Eigenschaften der extrahierten Faktoren.

Faktor	Eigenwert	Erklärungsanteil (Prozent)	Gesamter Erklärungsanteil (Prozent)
1	11,07263	19,1	19,1
2	5,65696	9,8	28,8
3	4,17493	7,2	36,0
4	3,99724	6,9	42,9
5	3,11268	5,4	48,3
6	2,66170	4,6	52,9
7	2,55859	4,4	57,3
8	2,29830	4,0	61,3
9	2,17543	3,8	65,0

Faktor	Eigenwert	Erklärungsanteil (Prozent)	Gesamter Erklärungsanteil (Prozent)
10	1,79176	3,1	68,1
11	1,61974	2,8	70,9
12	1,31517	2,3	73,2
13	1,20111	2,1	75,2
14	1,18860	2,0	77,3
15	1,12893	1,9	79,2
16	1,06446	1,8	81,1
17	1,05060	1,8	82,9

Nach der Faktorenrotation ergab sich die folgende Ladungsmatrix (aufgrund der Größe nur für die acht wichtigsten Faktoren dargestellt). Werte mit besonders hoher Ladung sind fett dargestellt.

Variable	Faktor 1	Faktor 2	Faktor 3	Faktor 4	Faktor 5	Faktor 6	Faktor 7	Faktor 8
V01	0,01040	-0,05794	0,00253	0,02823	-0,24182	0,29493	0,01852	0,05108
V02	0,04982	-0,01559	0,08558	0,00862	0,13982	**0,90888**	-0,00386	-0,02717
V03	0,07232	0,00699	-0,04409	-0,09565	-0,04403	-0,13711	0,06587	0,01314
V04	0,06193	-0,01384	0,05587	-0,06007	0,12228	**0,90616**	-0,00253	-0,09223
V05	0,07836	0,04686	-0,00058	0,03677	0,19396	0,28769	-0,11502	0,13309
V06	0,04253	-0,00052	0,07284	-0,04270	0,24712	**0,92020**	-0,01253	0,03441
V07	-0,08655	0,01322	0,06885	**-0,52105**	0,29791	-0,03039	**0,49676**	0,15428
V08	-0,03848	-0,00743	0,03421	-0,06054	**0,93527**	0,15218	-0,10175	0,07992
V09	-0,04102	-0,00816	0,02737	-0,05836	**0,93684**	0,14702	-0,09592	0,07313
V10	0,16177	0,03021	0,02107	0,08397	0,15831	0,01460	**-0,91235**	0,01649
V11	-0,03748	0,02893	0,12541	**0,93714**	-0,02715	-0,04580	-0,01213	-0,05823
V12	-0,07765	-0,02944	-0,02848	**-0,77719**	-0,09073	0,02576	**0,56062**	0,00388
V13	-0,16250	-0,03111	-0,02246	-0,08423	-0,15968	-0,01275	**0,91173**	-0,01528
V14	0,03702	-0,02786	-0,12507	**-0,93740**	0,02650	0,04588	0,01149	0,05894
V15	0,19678	0,08464	0,03260	0,13385	0,06303	0,19063	-0,07231	0,09276
V16	-0,11456	0,08173	0,09346	0,09835	0,15927	0,19418	-0,18285	**-0,76435**
V17	-0,05744	0,00458	0,00617	-0,06548	**-0,78554**	-0,31939	0,11899	0,16891
V18	0,04513	0,07282	-0,03350	-0,03919	0,18558	0,10216	-0,06945	**0,82841**
V19	0,04397	-0,03381	-0,07677	-0,05520	0,31282	0,33779	0,32701	-0,31643
V20	0,05772	-0,04305	0,00836	0,02991	0,11896	0,07298	-0,08736	-0,04326
V21	-0,02225	-0,03626	-0,01070	-0,07869	0,04002	-0,00078	-0,04046	-0,04722
V22	0,11676	0,06136	0,07604	0,18637	0,10864	0,02705	-0,25169	0,04378
V23	**-0,81993**	-0,40671	-0,14273	0,11269	-0,01344	-0,00603	0,17055	-0,02637
V24	-0,13847	0,01204	0,00819	-0,04072	0,11737	0,21213	-0,07008	0,05979
V25	**0,79901**	0,20761	0,11943	0,25490	-0,00707	0,08007	0,06251	0,07763
V26	0,31426	**0,81798**	0,33732	0,06115	0,02643	-0,01190	0,00797	0,01635
V27	0,11504	0,16733	**0,91920**	0,08479	0,01006	0,05506	-0,02802	0,01936
V28	**-0,81993**	-0,40671	-0,14273	0,11269	-0,01344	-0,00603	0,17055	-0,02637
V29	-0,45271	-0,17419	-0,15524	-0,01561	0,15209	0,16011	-0,13998	**-0,63373**
V30	**-0,61519**	-0,26804	-0,17808	-0,17926	-0,46724	-0,22653	0,08339	0,10233
V31	-0,25375	-0,15376	-0,03826	-0,20448	0,17619	0,05141	-0,22968	**0,72403**

Variable	Faktor 1	Faktor 2	Faktor 3	Faktor 4	Faktor 5	Faktor 6	Faktor 7	Faktor 8
V32	-0,07535	-0,14783	-0,11936	-0,10156	0,29433	0,31988	0,30838	-0,36260
V33	-0,16788	-0,04968	-0,08677	-0,00122	0,14140	0,05029	-0,08081	-0,05046
V34	-0,12015	-0,04522	-0,03190	-0,11566	0,05716	-0,00410	-0,01563	-0,04288
V35	**0,49151**	0,05922	0,07688	0,17384	0,01512	0,03388	-0,05140	-0,14069
V36	**0,80285**	-0,15555	-0,06249	0,18950	-0,14923	-0,00547	0,02413	0,04850
V37	0,46606	0,05603	-0,13677	0,31421	0,02986	0,11132	0,27178	0,26706
V38	0,16171	0,04835	-0,07317	0,07598	0,08318	0,09445	0,07956	-0,00124
V39	0,38713	-0,20792	0,13105	0,05237	-0,05041	0,05320	0,00673	-0,01103
V40	0,19984	-0,13573	0,04130	-0,08159	-0,05322	-0,01157	-0,15230	-0,04541
V41	0,17953	**0,71225**	0,31409	0,05515	-0,01024	0,02391	-0,03148	-0,10130
V42	0,25671	**0,85890**	0,16666	0,03818	-0,05247	-0,02430	-0,03132	-0,01053
V43	0,13883	**0,69883**	0,09054	-0,05669	0,02386	-0,01037	0,01164	0,09714
V44	0,20339	0,34336	0,06845	-0,01483	0,09260	0,01966	0,13382	0,02020
V45	0,14926	0,38766	-0,07967	-0,01481	0,00317	-0,00605	-0,07634	0,03262
V46	0,06788	0,17267	-0,06509	0,16369	0,04156	0,00684	0,07190	0,01514
V47	0,04895	0,20065	**0,77585**	-0,01640	-0,00548	0,03683	-0,01618	-0,08298
V48	0,07815	0,19089	**0,91531**	0,04539	-0,01764	0,02800	0,01165	-0,01392
V49	0,04782	0,13851	**0,54255**	0,00697	0,04732	0,01331	-0,02080	0,10898
V50	0,05312	0,09372	0,41511	0,07426	-0,02114	0,08995	-0,06876	0,03996
V51	0,12759	0,02268	**0,56014**	0,13447	0,11813	0,06358	0,01332	0,00095
V52	0,05065	-0,01952	0,22699	0,04256	-0,05997	0,04479	0,00151	0,00724
V53	**-0,81993**	-0,40671	-0,14273	0,11269	-0,01344	-0,00603	0,17055	-0,02637
V54	**0,60752**	0,17679	-0,08242	-0,28622	-0,03042	0,04794	-0,19380	0,01823
V55	**0,55820**	0,38088	0,40464	0,23769	0,02788	0,09552	0,06073	0,04476
V56	0,29630	**0,87658**	0,14237	0,04708	0,00968	-0,03005	-0,01599	-0,00249
V57	0,11080	0,17787	**0,92518**	0,08165	0,03137	0,03737	-0,02669	0,01523
V58	**-0,76799**	-0,38706	-0,15393	0,15220	-0,03199	-0,01665	0,16857	0,01221

Die Analyse der rotierten Ladungsmatrix ergab die in der folgenden Tabelle angeführten Interpretationen der Faktoren (angeführt sind die acht wichtigsten Faktoren).

Faktor	Interpretation	Variablen mit hoher Ladung
1	Farbanzahl unsegmentierter Bilder	Segmentierung, Zahl der Farben, Zahl der Farbabstufungen, Farbe Weiß des regionalisierten Farbhistogramms
2	Elemente, die in ungefähr der Hälfte aller Fälle 0 sind	Dritte Region des regionalen Farbhistogramms sowie der regionalisierten Farbabstufungen
3	Elemente, die in mehr als 80 Prozent aller Fälle 0 sind	Vierte Region des regionalen Farbhistogramms sowie der regionalisierten Farbabstufungen
4	Y-Achsen-Symmetrie	Symmetrietyp, Y-Achsen-Symmetrie, inverse Y-Achsen-Symmetrie
5	Diagonale Symmetrie	Diagonale Symmetrien, Farbe Weiß des Farbhistogramms
6	Varianzen des Komplexitäts-Features	Varianz der Kantenlänge, Varianz der Winkel zwischen Kanten, Anzahl der Kanten in einem Bild

Faktor	Interpretation	Variablen mit hoher Ladung
7	X-Achsen-Symmetrie	Symmetrietyp, X-Achsen-Symmetrie, inverse Y-Achsen-Symmetrie
8	Unsegmentierte Bilder mit gleicher Relation des Rot- und Schwarz-Anteils	Farben Rot und Schwarz des Farbhistogramms sowie der ersten Region des regionalisierten Farbhistogramms

Die Analyse der Kommunalitäten ergab, daß die Faktoren für fast die Hälfte aller Variablen mehr als 90 Prozent der Varianz erklären:

Erklärungsanteil (Prozent)	Anzahl der Variablen	Anteil (Prozent)
95 – 100	13	22.4
90 - 95	15	48.3
80 - 90	10	65.5
70 - 80	8	79.3
0 - 70	12	100

Besonders schlecht erklärt wird nur die Varianz der Farben Rot, Blau und Grün der zweiten Region des regionalisierten Farbhistogramms. Das ist das jeweils zweite Feld eines segmentierten Wappens.

3.3.4 Zusammenfassung

In diesem Abschnitt wird auf die Analyse des Datenmaterials der Testumgebung eingegangen. Eine solche ist wichtig, da ohne genaue Kenntnis der Eingabedaten die Qualität eines CBIR-Systems nicht beurteilt werden kann. Es werden die verwendeten Methoden, hierarchische Clusteranalyse, Self-organizing Maps und Faktorenanalyse sowie einige wichtige, praxisrelevante Begriffe erklärt. Außerdem wird auf die zur Verfügung stehenden Eingabedaten eingegangen. Im letzten Abschnitt werden, getrennt für Fälle und Variablen, die Ergebnisse der Auswertungen (Dendrogramme, Maps, etc.) gezeigt und allgemeine Schlußfolgerungen gezogen. Die wichtigste dieser Schlußfolgerungen ist die Erfüllung der Clusterhypothese durch die angegebenen Daten, so daß die Verwendung von Suchmodellen zur Suche gerechtfertigt ist.

In den folgenden Kapiteln wird immer wieder von der sogenannten „natürlichen Clusterung“ (oder: Gruppierung) gesprochen werden. Die Vielzahl der in diesem Abschnitt vorgestellten Methoden, deren Anwendung zu meist unterschiedlichen Ergebnissen führt, zeigt, daß es nicht nur eine passende Clusterung für eine Datenmenge gibt. Da das Verfahren von Kohonen aber dazu führt, daß die natürlichen Ordnung in der Ergebnismap teilweise erhalten bleibt, sei unter einer natürlichen Clusterung in der Folge eine Clusterung aufgrund von Self-organizing Maps verstanden.

4 Automatisierung der Gewichtung und Reihung in Suchmodellen

In diesem Kapitel werden Lösungen für zwei grundlegende Probleme des CBIR beschrieben, die Gewichtung von Features und die Verbesserung der Performance bei der Suche. Beide Algorithmen beruhen auf dem Suchmodell-Konzept und verwenden gemeinsame Basismethoden.

4.1 Automatische Gewichtung für Suchmodelle

CBIR-Ansätze, die auf der Verwendung von mehreren Features beruhen (Multifeature-Ansätze), verwenden zumeist Gewichte, um die relative Bedeutung der Features festzulegen. Gewichtet wird dabei der Distanzwert eines untersuchten Bildes zu einem Suchbild für die Distanzfunktion des betrachteten Features. Die Bildung einer gewichteten Summe von Distanzwerten wurde bereits im Literaturüberblick ("Positionswert") erklärt. Dabei wurde auch erwähnt, daß sich aus dieser Methode, dem sogenannten linearen Merging von Features, im wesentlichen zwei Probleme ergeben:

1. Manche Featureklassen stehen nicht in einem linearen Verhältnis zueinander und es erscheint daher wenig passend, die Verknüpfung linear durchzuführen. Vielmehr wird zumeist die Verknüpfung durch neuronale Netze empfohlen (vgl. [54]). Im Literaturüberblick werden dazu einige Beispiele angeführt.

2. Die Gewichte der Features sind, so sie nicht ohnehin fixer Bestandteil des Suchsystems sind (wie z. B. in QBIC), durch den Benutzer anzugeben. Dieser kennt normalerweise aber weder die interne Struktur der Features, noch ist er dazu in der Lage, eine Präferenzordnung dieser Features anzugeben. Er ist daher, und das wird in der Literatur häufig bemängelt (vgl. [38]), mit der Vergabe von Gewichten eigentlich überfordert.

Für die Lösung des zweiten Problems wurde überlegt, wie eine automatische Ableitung der Gewichte funktionieren könnte und welche Daten dafür benötigt würden. Auf der Basis dieser Überlegungen wurde ein Algorithmus implementiert, der für ein bestimmtes Feature und Bild einer Bildmenge das Gewicht ableitet. Außerdem wurde eine Testumgebung geschaffen, in der die Qualität dieses Algorithmus evaluiert wurde.

Abschnitt 4.1.1 beschäftigt sich detailliert mit dem Problem des Mergings und leitet direkt zum gewählten Lösungsansatz über, Abschnitt 4.1.2 skizziert die Implementierung des Algorithmus sowie aller (externen) Komponenten, aus denen er besteht und Abschnitt beschreibt die Testumgebung, welche Methoden zur Evaluierung verwendet wurden, welche Ergebnisse erzielt wurden und wie diese zu bewerten sind.

4.1.1 Merging durch Self-organizing Maps

Zum linearen Merging wird üblicherweise Gleichung 32 benutzt. Die Reihung von Bildern in der Ergebnismenge erfolgt dabei anhand der Positionswerte der Objekte, der gewichteten Distanzsummen über alle Features. Das Bild mit dem kleinsten Positionswert wird ganz vorne gereiht.

$$Positionswert_{Objekt} = \sum_{i=1}^{F} w_i d_i \tag{32}$$

Dabei sind w_i das Gewicht und d_i der Distanzwert für das i-te von F Features. Grundlegende Voraussetzung für die Anwendbarkeit dieser Formel ist die Standardisierung aller verwendeten Distanzfunktionen auf denselben Wertebereich. Für die Features, die im Rahmen dieser Dissertation entwickelt wurden, ist das das Intervall [0, 1]. Bei Anwendung von Gleichung 32 und der Vereinbarung, daß das Bild mit dem kleinsten Positionswert in der Ergebnismenge an erster Stelle steht, bedeutet ein hohes Gewicht für ein Feature, daß ihm bei der Suche große Bedeutung zukommt. Bilder, die für ein Feature mit großer Bedeutung eine große Distanz zum Suchbild aufweisen, werden durch das größere Produkt aus Gewicht und Distanzwert tendenziell weiter hinten gereiht. Das Ziel eines Algorithmus für die automatische Vergabe von Gewichten muß es also sein, für wichtigere Features (an welchen Kriterien das auch immer beurteilt wird) größere Gewichte zu vergeben als für weniger wichtige.

Von besonderer Bedeutung für die Vergabe von Gewichten ist die Wahl der Methode, durch die beim Suchen die Größe der Ergebnismenge bestimmt wird. Bei der Steuerung der Ergebnismenge durch eine absolute Zahl können die Gewichte ihre Funktion, die tatsächlich ähnlichsten Bilder innerhalb der Grenzen der Ergebnismenge zu reihen, nicht zufriedenstellend erfüllen. Das wurde im Abschnitt 3.2 gezeigt. Verwendet man Schwellwerte, so reduziert sich die Funktion der Gewichte darauf, innerhalb der Ergebnismenge eine Reihung der Bilder vorzunehmen. Das ist eine Aufgabe, zu deren Erfüllung bereits weniger maßgeschneiderte, z. B. automatisch generierte, Gewichte ausreichen. Die Größe der Ergebnismenge ist bei dieser Methode von den Gewichten unabhängig.

Der unten dargestellte Gewichtungsalgorithmus beruht auf der Idee, daß sich eine Bildermenge normalerweise in inhaltlich zusammengehörende Bereiche clustern läßt. Eine solche Clusterung kann z. B. mit Hilfe einer Clusteranalyse oder von Self-organizing Maps (SOM) erfolgen, wobei als Eingabedaten üblicherweise nicht die Bilder selbst, sondern passende Repräsentationen verwendet werden. Als Repräsentation eines Bildes bieten sich natürlich die, von den verschiedenen Features erzeugten, zu einem Gesamtfeaturevektor verbundenen Featurevektoren an. Gruppiert man nun diese Gesamtfeaturevektoren in einer zweidimensionalen Map, so läßt sich aus der Distanz eines für einen Cluster repräsentativen Featurevektors zu allen direkten Nachbarn (gemessen durch die zum Feature gehörige

Distanzfunktion) eine Aussage über die Bedeutung dieses Features für die Bildung des betrachteten Clusters machen (vgl. Abbildung 39). Geht man weiters davon aus, daß es sich bei der durch die Clusterungsmethode erzeugte Gruppierung der Bilddaten um eine natürliche, das heißt für den menschlichen Betrachter akzeptable, Einteilung handelt, so wird der Beitrag eines Features zur Bildung eines Clusters ein relevanter Wert für die Bedeutung des Features, wenn nach Bildern aus dem betrachteten Cluster gesucht wird. Die Idee des Gewichtungsalgorithmus ist es daher, aus dieser Information das Gewicht eines Features abzuleiten.

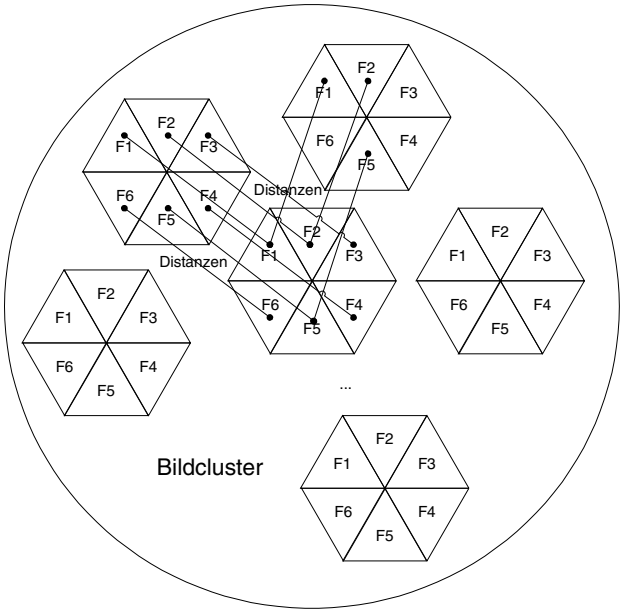

Abbildung 39: Featurebasierte Bildmengenclusterung

Im Detail besteht der Gewichtungsalgorithmus aus drei Schritten:

1. Clusterung der Bilddatenbank: dazu müssen die Gesamtfeaturevektoren für alle Bilder gebildet werden. Diese Vektoren werden dann durch einen SOM-Algorithmus in einer zweidimensionalen Map angeordnet. Dabei wird jeder Cluster durch einen Mittelwertvektor repräsentiert und ein Bild gehört zu jenem Cluster, zu dessen Mittelwertvektor es die geringste euklidische Distanz aufweist.

2. Berechnung der Gewichte für ein Bild: es ist zuerst festzustellen, zu welchem Cluster das Bild gehört, sodann ist für jedes Feature getrennt die Distanz des Featurevektors zu seinen Nachbarn festzustellen und aufzusummieren. Für diesen Schritt bieten sich zwei Möglichkeiten an: Messung der Distanz vom Clustermittelpunkt zu den Clustermittelpunkten (Variante 1) der Nachbarn oder Messung der Distanz vom Suchbild zu den Clustermittelpunkten der Nachbarcluster (Variante 2). In Abbildung 40 werden diese beiden Varianten dargestellt. Während der Implementierung des Gewichtungsalgorithmus wurden beide Varianten untersucht und festgestellt, daß sich für Variante 1 größere Varianzen der Distanzsummen ergeben. Da größere Varianzen und damit

unterschiedlichere Gewichte generell positiv beurteilt wurden, wurde für die Feststellung der Distanzsummen im Gewichtungsalgorithmus Variante 1 implementiert.

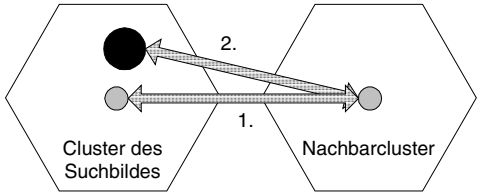

Abbildung 40: Varianten der Berechnung von Gewichten

Bevor die Distanzsumme als Gewicht verwendet werden kann, muß sie noch auf eine einheitliche Zahl von Nachbarclustern standardisiert werden (da Randcluster weniger Nachbarn haben als andere). Dazu wird sie durch die Anzahl der vorhandenen Nachbarn dividiert.

3. Die Anwendung der Cluster erfolgt relativ, das heißt es ist nicht der absolute Wert eines Gewichtes entscheidend, sondern sein relativer Wert im Vergleich zu den Gewichten anderer Features. Daher ist auch keine Standardisierung der Gewichte auf einen einheitlichen Wertebereich nötig. Die Summe der verwendeten Gewichte ist nicht konstant.

4.1.2 Implementierung

Der Gewichtungsalgorithmus wurde in der im Abschnitt 3.1.3 vorgestellten QBIC-Testumgebung als C-Bibliothek implementiert, wobei nicht alle Features verwendet werden konnten. Das bedingte Farbhistogramm, das Siegel-Feature und das Objekt-Layout-Feature erwiesen sich aus verschiedenen Gründen als ungeeignet. Das bedingte Farbhistogramm hat den selben Featurevektor wie das Tinktur-Histogramm und unterscheidet sich nur in der Distanzfunktion, das Siegel-Feature erzeugt keinen Featurevektor sondern eine Featurematrix, wobei die Besonderheit des Features wiederum im Distanzvergleich liegt und das Objekt-Layout-Feature schließlich erzeugt einen Featurevektor von dynamischer Länge, wobei die Bedeutung der einzelnen Elemente unterschiedlich ist. Die restlichen 16 Features konnten aber für den Gewichtungsalgorithmus herangezogen werden. Sie bilden einen insgesamt 58-stelligen Gesamtfeaturevektor, dessen Elemente für alle Bilder auf den Wertebereich [0, 10] normalisiert wurden.

Für die Clusterung der Vektoren wurde SOM-PAK benutzt, wobei die erstellte Map folgende Eigenschaften aufweist:

Parameter	Wert
Map-Layout	hexagonal
Dimensionen	8 x 6 bins (bei 444 Bildern)
Neighbourhood kernel	bubble

Es wurden 50 verschiedene Tests mit jeweils 550000 Lernschritten durchgeführt. Dabei ergab sich für die beste Map ein minimaler Quantifizierungsfehler von 7.467. Die so erzeugte Map wurde auch im Abschnitt 3.3 verwendet.

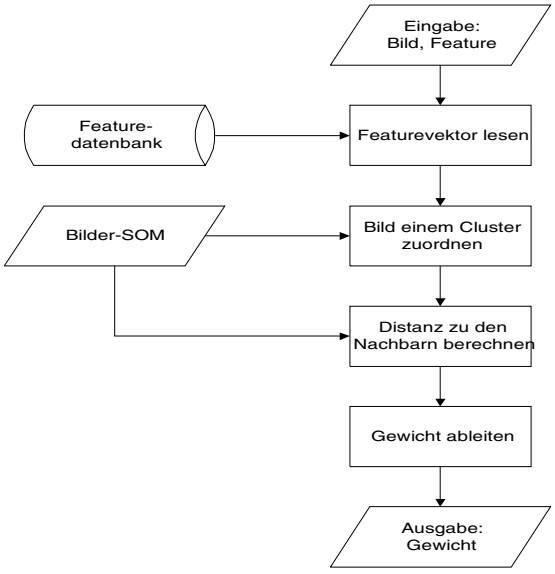

Abbildung 41: Datenfluß im Gewichtungsalgorithmus

Zentrales Element des Gewichtungsalgorithmus ist die Funktion getWeight(), die schematisch in Abbildung 41 dargestellt ist. Zuerst wird für das gegebene Bild aus dem angegebenen Bild-Katalog der Datenvektor für das angegebene Feature gelesen. Mit dem Gesamtfeaturevektor wird in der Cluster-Map der Bin (Cluster), zu dem das Bild gehört, gesucht und der den Bin beschreibende Mittelwertvektor zwischengespeichert. Danach wird festgestellt, wie viele Nachbarn es zu diesem Bin gibt und die Distanzsumme zu allen Nachbarn berechnet. Dieser Wert schließlich wird standardisiert und als Gewicht benutzt.

Ein generelles Problem der Anwendung einer statischen SOM in einer lebenden Bilddatenbank und damit des Gewichtungsalgorithmus ist es, die Clusterung aktuell zu halten. Viele Eingaben in die Datenbank führen möglicherweise zu einer Verschiebung der Datenverteilung.

Für die praktische Anwendung des Gewichtungsalgorithmus sind unter anderem folgende Lösungsansätze denkbar:

- Periodische Neuberechnung der Map aus allen Bilddaten, wobei die Periode entweder auf die Zeit (wöchentlich, etc.) oder auf die durchgeführten Datenbankoperationen (z. B. Neuberechnung nach N Eingaben) bezogen sein kann. Der Hauptnachteil dieser Methode ist ihr hoher Ressourcenaufwand.

- Auswahl einer repräsentativen Stichprobe, anhand derer die Map berechnet wird. Durch die Wahl einer kleinen Stichprobe ließe sich die Map häufiger aktualisieren, die Clusterung würde die Verteilung der Bilddaten aber auch weniger genau widerspiegeln.

4.1.3 Auswertungen

Aus den 444 Gesamtfeaturevektoren der Wappendatenbank wurde ein Map mit 8x6 Bins berechnet, so daß jeder Cluster ca. 5 bis 20 Bilder enthält. Abbildung 42 zeigt einen Blick auf die Ergebnis-Map, wobei jeder Cluster durch das Bild dargestellt wird, das dem Mittelwertvektor am nächsten liegt (ikonischer Index).

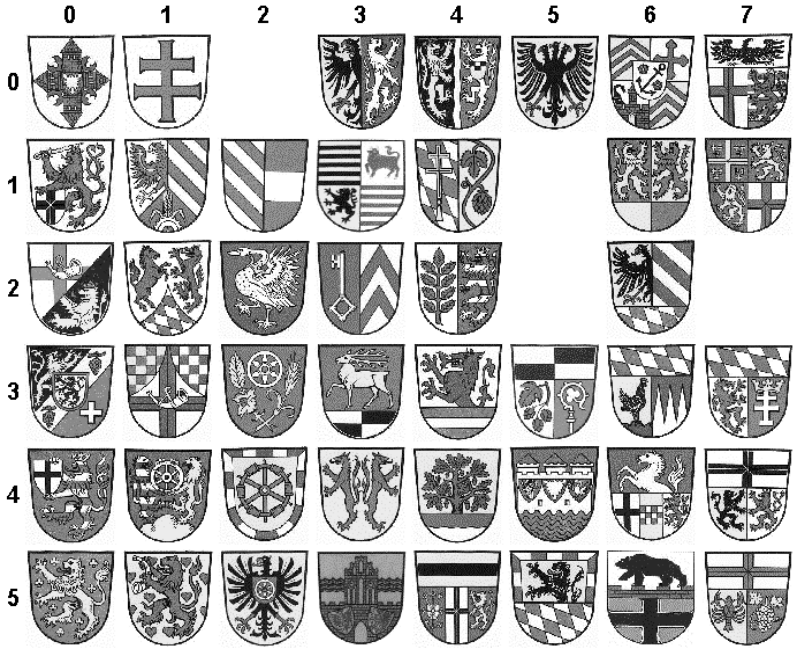

Abbildung 42: Ikonischer Index der Datenbasis

Die freien Bereiche (z. B. Cluster 2/0, 5/1) zeigen Cluster, zu denen keine passenden Bilder gefunden wurde. Beim Durchsehen des Bildmaterials ergibt sich der Eindruck, daß bei der Clusterung vor allem folgende Features eine wesentliche Rolle gespielt haben: Segmentierung eines Wappens, Histogramm der Tinkturen, Zahl der Farben und Farbabstufungen. Abbildung 43 zeigt einen typischen Bildcluster (Cluster 0/5). Diese Bilder weisen folgende gemeinsamen Eigenschaften auf: gleiche Segmentierung (keine), ähnliche

Farbverwendung (blau, gelb, weiß), ähnliche Farbenanzahl (3 - 4), kaum Symmetrien, mittlere Komplexität.

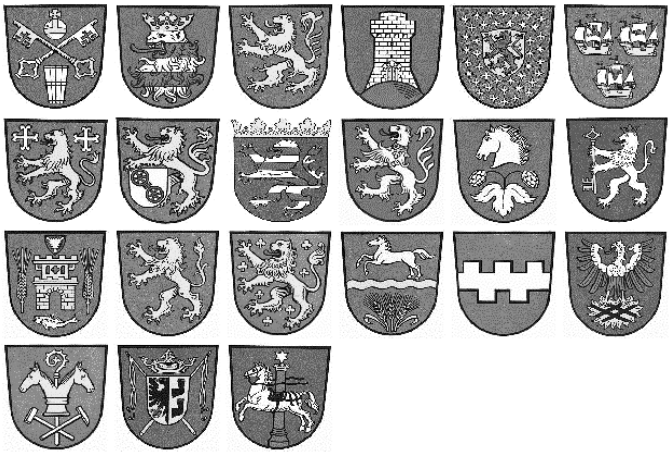

Abbildung 43: Beispielcluster

Nach der Auswertung der Bildcluster wurde in einem nächsten Schritt die Verteilung der Gewichte über alle Bilder und Features untersucht. Dabei ergaben sich nach dem Gewichtungsalgorithmus folgende durchschnittlichen Gewichte der Features (vgl. Abbildung 44): den größten Einfluß haben die regionalisierten Features für die Anzahl der Wappenfarben und Farbabstufungen sowie das Segmentierungsfeature. Das mag unter anderem daran liegen, daß sie mit nur sehr wenigen Vektorelementen grundlegende Eigenschaften der Bilder beschreiben. Die Symmetriefeatures liegen sämtlich im Bereich mittlerer Bedeutung, eher gering ist jene des Komplexitäts-Features.

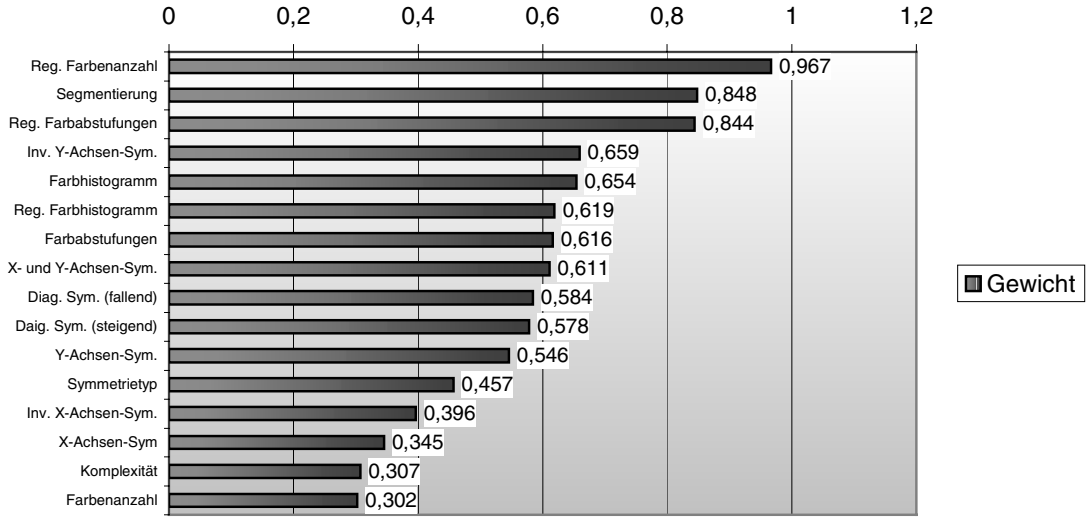

Abbildung 44: Durchschnittliche Gewichte der Wappen-Features

Außerdem wurde untersucht, welche Features bezogen auf jedes einzelne Bild am häufigsten das größte Gewicht (und damit die größte Bedeutung) haben (vgl. Abbildung 45). Dabei ergab sich, daß nur sieben der 16 betrachteten Features überhaupt jemals die wichtigste

Eigenschaft eines bestimmten Bildes messen: am häufigsten das Farbhistogramm, das Segmentierungsfeature und die regionalisierte Anzahl der Wappenfarben (> 20 Prozent), von durchschnittlicher Bedeutung sind die Features zur Messung der Anzahl der Farbabstufungen (global und regional) und manchmal (< 5 Prozent) ist die wichtigste Eigenschaft eines Bildes die diagonale Symmetrie. Das liegt vermutlich unter anderem daran, daß keine diagonalen Segmentierungstypen unterschieden werden und Bilder mit spezifisch diagonalem Layout daher vor allem anhand von Symmetrien erkannt werden.

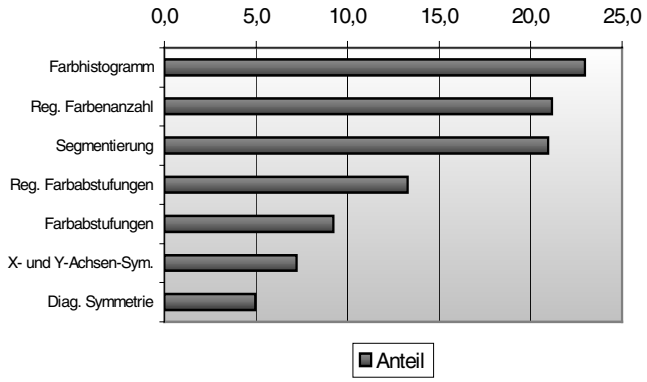

Abbildung 45: Häufigste wichtigste Features

Die eigentliche Bewertung des Gewichtungsalgorithmus erfolgte durch Suchbeispiele. Anhand der Suchergebnisse wurde die Reihung, die sich durch die Gewichtung aufgrund von Clusterinformationen ergab ("SOM Gewichte"), mit der Reihung verglichen, die sich ergab, wenn alle Gewichte gleich gesetzt wurden ("Konstante Gewichte"). Grundlage für die Suche waren in beiden Fällen Suchmodelle mit Schwellwerten, das heißt es wurde jeweils die gleiche Ergebnismenge retourniert, unterschiedlich war nur die Reihung dieser Mengen.

Zur Beurteilung der Reihungsergebnisse wurde folgender Algorithmus angewendet:

1. Wahl der fünf zur Suche ähnlichsten Bilder der Ergebnismengen und Bestimmung der Reihung, wie sie aus Sicht des Beurteilenden sein sollte. Dies ist der einzige Schritt, in dem die Subjektivität des Beurteilers in den Bewertungsprozeß einfließt.

2. Bestimmung der Fehlersumme für die Reihung der Suchergebnisse nach der SOM-Methode beziehungsweise der Methode mit den konstanten Gewichten, wobei nur die in Punkt 1 ausgesuchten Bilder betrachtet werden. Die Fehlersumme für ein Suchergebnis ist anhand von Gleichung 33 definiert:

$$Fehlersumme = \sum_{i=1}^{5} \left| Istposition_i - Sollposition_i \right| \tag{33}$$

   Hierbei sind $Istposition_i$ und $Sollposition_i$ die Position von Bild i in der Ergebnismenge (beginnend mit 1).

3. Die Qualität einer Reihungsmethode wurde definiert als das Verhältnis der Fehlersumme zur maximal möglichen Fehlersumme (vgl. Gleichung 34). Es wurden nur die ersten zwölf Bilder einer Ergebnismenge betrachtet, woraus sich eine maximal mögliche Fehlersumme von 45 ergab (Fehler: 11 beim ersten, 10 beim zweiten, etc.).

$$Qualität = 1 - \frac{Fehlersumme}{höchstmögliche\ Fehlersumme} \tag{34}$$

Dieser Qualitätswert ist definiert für das Intervall [0, 1], wobei das optimale Ergebnis 1 ist. Abbildung 46 zeigt die Auswertungsergebnisse für die oben angegebenen Basismethoden.

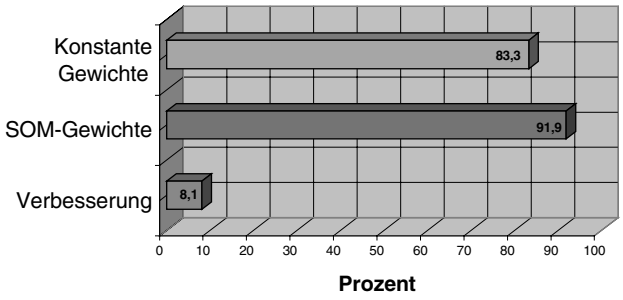

Abbildung 46: Qualität der Gewichtungsergebnisse

Dabei zeigt sich, daß die Verwendung von Suchmodellen mit Schwellwerten zur Bestimmung der Größe der Ergebnismenge bereits in 83.3 Prozent aller Fälle zu einer aus Sicht des Beobachters richtigen Reihung der Ergebnismenge führt ("Konstante Gewichte"). Bezieht man bei der Reihung darüber hinaus noch Wissen über die Verteilung der Daten mit ein, verbessert sich die Reihungs-Qualität um weitere 8 Prozent auf fast 92 Prozent. Das heißt in 92 Prozent aller Fälle ist eine Ergebnismenge, die durch die Verwendung von Suchmodellen mit automatisch hergeleiteten Gewichten generiert wurde, optimal gereiht. Das ist, wenn man außerdem noch berücksichtigt, daß die Bedeutung der Gewichte durch die Verwendung von Schwellwerten ohnehin stark reduziert wurde, ein ausreichend gutes Ergebnis, um die Gewichtung vollautomatisch durchzuführen.

Abschließend sei hier noch die Webschnittstelle beschrieben, die zur Bewertung des Gewichtungsalgorithmus erstellt wurde (vgl. Abbildung 47). Sie wurde abgeleitet aus der allgemeinen Webschnittstelle für Suchmodelle, wobei folgende Veränderungen vorgenommen worden sind:

- Der Abschnitt über die Statistik der letzten Suche wurde, da er für diese Aufgabe nicht relevant ist, entfernt.

- Anstelle einer Ergebnismenge werden zwei angezeigt: die erste mit der Reihung aufgrund konstanter Gewichte, die zweite mit SOM-Gewichten. Es werden nur die ersten zwölf

Bilder angezeigt und auch nur die Bilder selbst. Positionswert und Distanzwerte werden zur Erhöhung der Übersichtlichkeit nicht angezeigt.

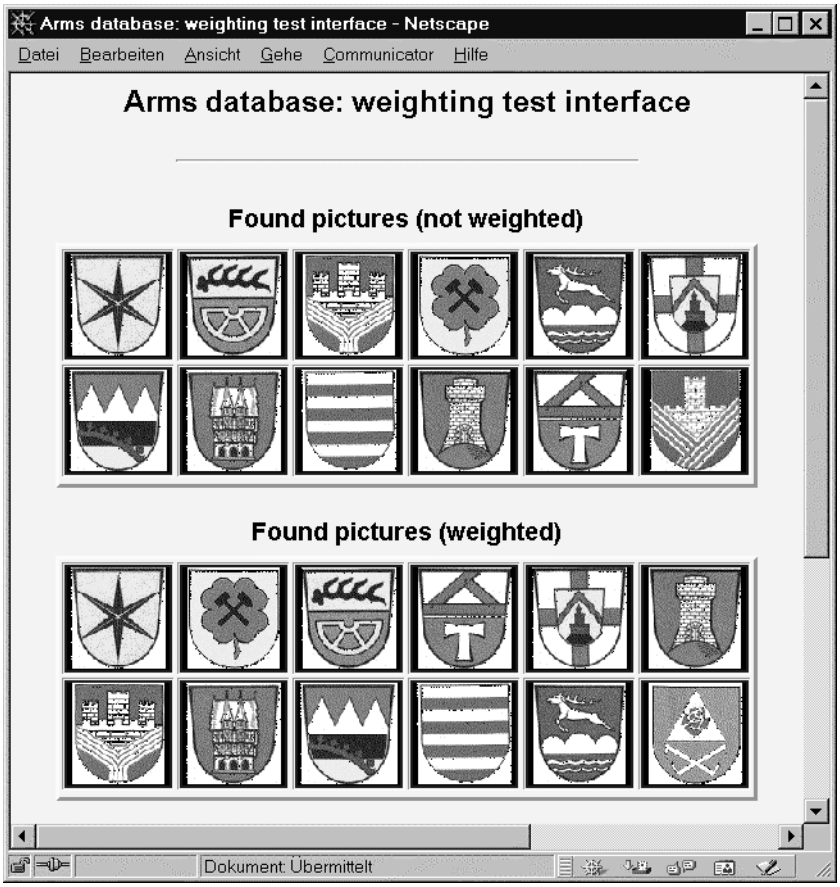

Abbildung 47: Screenshot von der Webschnittstelle für die Gewichtung

Im in Abbildung 47 angegebenen Beispiel wurde nach Bildern gesucht, die eine Symmetrie um die Y-Achse aufweisen und von mittlerer Komplexität sind. Bei der Betrachtung des Suchergebnisses ist zu beachten, daß die verwendeten Suchmodelle nicht gebildet wurden, um Ähnlichkeit möglichst allgemein abzubilden, sondern um den Algorithmus zur Gewichtung zu verifizieren. Demnach sind die Ergebnisse nur im Hinblick auf das verwendete Suchmodell ähnlich.

4.1.4 Zusammenfassung

Dieser Abschnitt beschäftigt sich mit der automatischen Gewichtung der Schichten von Suchmodellen. Der vorgestellte Ansatz leitet die Gewichte aus der natürlichen Clusterung der gegebenen Bildmenge ab, wobei die Clusterung durch Self-organizing Maps erfolgt. Die Ergebnisse legen nahe, daß der gewählte Ansatz gut für die praktische Anwendung in CBIR-Systemen geeignet ist.

4.2 Performance-optimierte Reihung von Suchmodellen

Eines der grundlegenden Probleme des CBIR ist die Retrieval-Geschwindigkeit. Da Ähnlichkeit durch bisweilen sehr aufwendige Distanzfunktionen gemessen wird, können sich beim Durchsuchen großer Bilddatenbanken lange Antwortzeiten ergeben. Aus diesem Grund wird in der CBIR-Forschung großes Augenmerk auf die Beschleunigung des Suchvorganges gelegt (vgl. die entsprechenden Abschnitte im Literaturüberblick).

Die Verwendung von Suchmodellen mit Schwellwerten ermöglicht nun eine ganz erhebliche Beschleunigung von Suchen. Dazu werden die verwendeten Features nach ihrem Aufwand für die Distanzberechnung in zwei Gruppen eingeteilt:

1. Features zur schnellen Vorauswahl von Bildern: diese zeichnen sich durch einfache Distanzfunktionen aus, wobei zumeist kurze Featurevektoren untersucht werden (z. B. die Features zur Bestimmung der Zahl der Farbabstufungen beziehungsweise Wappenfarben, etc.). Aufgabe dieser Features ist es, den Großteil der Bilder, die auf eine spezifische Anfrage nicht passen, schnell aus der Untersuchung auszuschließen.

2. Features zur Feinauswahl: dienen dem Finden der dem Suchbild ähnlichsten Bilder. Hier können, da nur noch wenige Bilder untersucht werden müssen, komplexe Distanzfunktionen für lange Featurevektoren benützt werden (Beispiele: Siegel-Feature, Objekt-Layout-Feature, etc.)

Durch die Kombination von Features zur Vor- und Feinauswahl erfolgt, wie in Abbildung 48 dargestellt, über die Features eines Suchmodells eine deutliche Abnahme der zu untersuchenden Bilder. Das Hauptproblem dieses Ansatzes ist es herauszufinden, welche Features zur Vor- und welche zur Feinauswahl geeignet sind. Um das automatisch feststellen zu können und Suchmodelle vor der Ausführung performance-optimiert reihen zu können, muß eine Aussage darüber gemacht werden können, wie groß die Ergebnismengen der einzelnen Features sind. Bei der Implementierung dieses Reihungsalgorithmus müssen folgende Teilprobleme gelöst werden:

1. Es muß vorhergesagt werden können, wie viele Bilder von einem bestimmten Feature, wenn es sich an einer bestimmten Stelle in einem Suchmodell befindet, als ähnlich weiter untersucht beziehungsweise als unähnlich ausgeschieden werden (Prognose-Problem).

2. Es muß ein Optimierungsalgorithmus entwickelt werden, der aus diesen Prognosedaten und den meßbaren Performancedaten der einzelnen Features die optimale Reihung ableitet (Optimierungs-Problem).

Dieser Abschnitt beschäftigt sich mit der Lösung dieser beiden Probleme, wobei das erste in zwei Teilbereiche aufgeteilt wird: Vorhersage der Größe der Rückgabemenge, wenn das betrachtete Feature an erster Stelle im Suchmodell steht und Bestimmung der Ähnlichkeiten

von Features, um eine Aussage darüber machen zu können, wie groß die Schnittmenge von jeweils zwei Features ist.

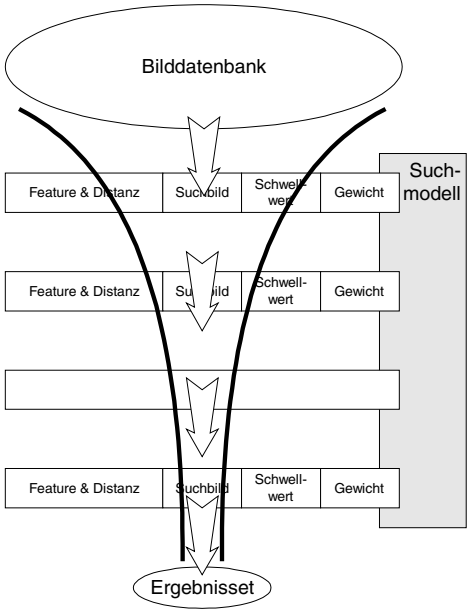

Abbildung 48: Schnelle Vorauswahl von Bildern

Im Abschnitt 4.2.1 wird gezeigt, aus welchen Komponenten der Reihungsalgorithmus besteht, welche Datenstrukturen verwendet werden und wie der Datenfluß im Modell ist, Abschnitt 4.2.2 beschäftigt sich mit der Implementierung des Reihungsalgorithmus, wobei auch skizziert wird, wie er in die Suchmaschine der Testumgebung eingefügt wurde und Abschnitt 4.2.3 schließlich beschreibt, wie der Reihungsalgorithmus getestet wurde und welche Ergebnisse dabei erzielt wurden.

4.2.1 Reihungsalgorithmus

Der Reihungsalgorithmus dient dazu, ein ungeordnetes Suchmodell so zu ordnen, daß die darauffolgende Suche performance-optimiert, das heißt möglichst schnell abläuft. Dazu wird im gewählten Ansatz ein Optimierungsalgorithmus verwendet, der als Eingaben die Performance der Distanzfunktionen der Features sowie Angaben über die zu erwartenden Rückgabemengen der einzelnen Features benötigt. Abbildung 49 zeigt den Datenfluß im dem Reihungsalgorithmus zugrunde liegenden Modell. Dabei bezeichnen die Rechtecke die Komponenten des Algorithmus: Vorhersage-, Relationen- und Optimierungsalgorithmus, die Parallelogramme Ein- und Ausgabedaten und die Zylinder die einzelnen Datenbasen.

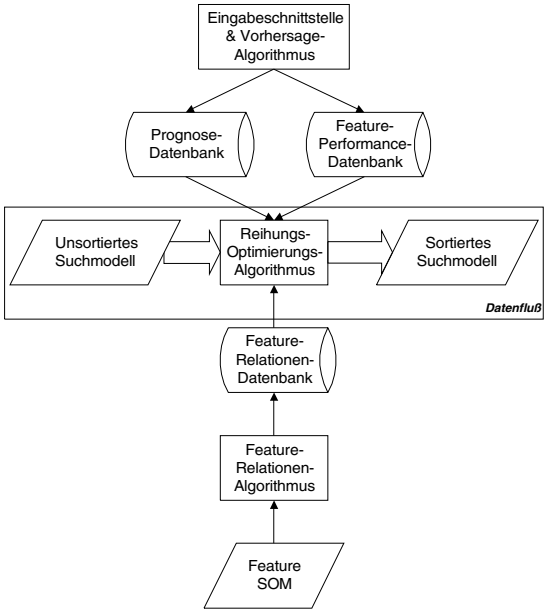

Abbildung 49: Datenfluß im Reihungsalgorithmus

Im Prognosemodell wird, wie oben beschrieben, zuerst versucht festzustellen, wie groß die Rückgabemenge eines Features ist, wenn es als erstes im Suchmodell gereiht wird. Dafür wird eine Datenstruktur verwendet, die für jedes Feature, jede Bildgruppe und jeden Schwellwert angibt, wie viele Bilder der Datenbank für ein Suchbild der betrachteten Gruppe bei einer hypothetischen Suche innerhalb des angegebenen Schwellwertes liegen. Dazu wird eine diskrete Dichtefunktion verwaltet. In Abbildung 50 ist schematisch (weil kontinuierlich eingezeichnet) die Datenstruktur für ein Feature dargestellt.

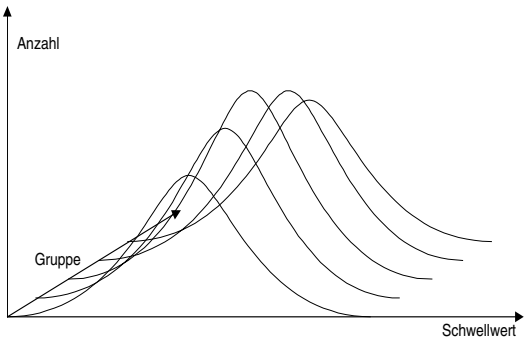

Abbildung 50: Prognose-Datenstruktur für Features

Bei jedem Hinzufügen oder Entfernen eines Bildes zur beziehungsweise aus der Datenbank wird für jedes Feature eine Abfrage über alle Bilder (mit dem maximalen Schwellwert) ausgeführt, um diese Datenstruktur aktuell zu halten. Die Gruppe eines Bildes wird folgendermaßen bestimmt:

- Verwendung des Featurevektors als Gruppenbezeichnung. Diese Methode wird bei allen Features mit einem eindimensionalen Featurevektor verwendet (Segmentierungs-Feature, etc.).

- Durchführung einer Distanzmessung zu einem Referenzobjekt, wobei der erhaltene Distanzwert als Gruppenbezeichnung verwendet wird. Diese Methode wird bei allen übrigen Features verwendet (Farbhistogramm, etc.). Abbildung 51 zeigt ein Referenzobjekt (schwarzer Kreis) und drei Bildgruppen mit gleicher Distanz (D1, D2 und D3).

Da das Hinzufügen beziehungsweise Entfernen eines Bildes kein zeitkritischer Vorgang ist, kann das Durchführen einer Suche über alle Features und Bilder in Kauf genommen werden.

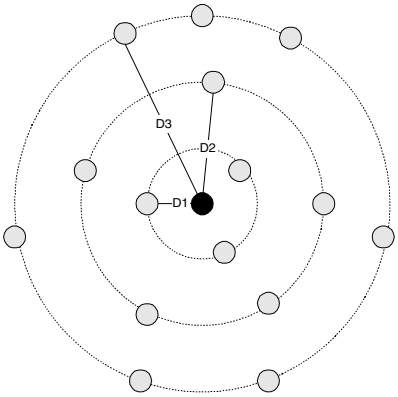

Abbildung 51: Bildung von Bildgruppen mit gleicher Distanz zu einem Referenzobjekt

Im zweiten Schritt wird versucht, Aussagen über die Ähnlichkeit von Features bezüglich ihrer Ergebnismenge zu machen, also die Schnittmenge von jeweils zwei Features zu bestimmen. Dadurch wird es möglich zu prognostizieren, wie groß die Ergebnismenge eines Features sein wird, wenn es sich an einer beliebigen Stelle im Suchmodell befindet. Zur Bestimmung der Ähnlichkeiten wurden die Elemente der Featurevektoren geclustert und ein Algorithmus zur Bestimmung der Relationen von Features anhand dieser Clusterung entworfen. Dieser Algorithmus arbeitet folgendermaßen:

1. Vergleich der Länge der Featurevektoren zweier Features.

2. Für alle Elemente des Featurevektors des Features mit dem längeren Vektor wird festgestellt, zu welchem Cluster der Featureelemente-Map (Cluster X) es gehört.

3. Sodann wird jenes Element des anderen Featurevektors gesucht, das sich im Cluster mit der minimalen Distanz zum Cluster X befindet. Als Distanzmaß wird dabei die euklidische Distanz verwendet.

Die Summe dieser Distanzen wird (reziprok) als Ausdruck für die Ähnlichkeit zweier Featurevektoren herangezogen. Das heißt je näher beisammen die Vektorelemente zweier Features in der Map liegen, um so größer ist vermutlich die Schnittmenge bei Abfragen nach diesen beiden Features. Die Auswertungen in Abschnitt 4.2.3 bestätigen diese Hypothese. In Abbildung 52 ist ein Beispiel für diesen Algorithmus angeführt. Dabei sind durch Kreise zwei

Featurevektoren $X = (X_1, X_2, X_3)^t$ und $Y = (Y_1, Y_2, Y_3, Y_4)^t$ und durch Pfeile die minimalen Distanzen angegeben.

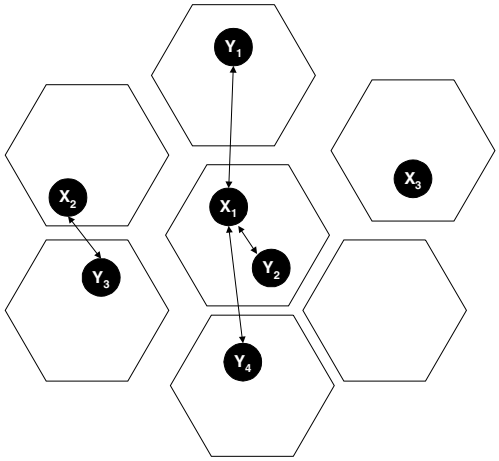

Abbildung 52: Relationen von Featurevektorelementen

Der Optimierungsalgorithmus versucht auf der Basis dieser Informationen sowie der Performancewerte der Distanzfunktionen der Features, das in Gleichung 35 dargestellte Optimierungsproblem zu lösen:

$$\min: \mathit{Ausführungsdauer} = \sum_{i=0}^{\mathit{Features}} t_i A_i \tag{35}$$

Dabei ist t_i die Dauer eines Distanzvergleiches für Feature i und A_i die Anzahl der Bilder, die durch das i-te Feature untersucht werden müssen (A_0 ist die Ausgangsmenge, das heißt alle Bilder in der Datenbank). Minimiert wird also die Dauer der Suche. In Gleichung 36 ist dargestellt, wie die Ausgangsmenge des i-ten Features aus der Größe der Grundmenge und den (erwarteten) Rückgabemengen der einzelnen Schichten eines Suchmodells abgeleitet werden kann:

$$A_i = A_{i-1} - R_{i-1} = A_{i-2} - R_{i-1} - R_{i-2} = \ldots = A_0 - \sum_{j=0}^{i-1} R_j \tag{36}$$

Hier ist R_i die Rückgabemenge des i-ten Features. Die Lösung dieser Optimierungsaufgabe erfordert die Entwicklung eines Algorithmus zur nicht-linearen Optimierung. Aus der Erkenntnis heraus, daß die Größe der Menge von Features in einem Suchmodell recht begrenzt ist, wurde nicht versucht, einen performance-optimierten Lösungsalgorithmus zu entwickeln. Der gewählte Optimierungsalgorithmus berechnet statt dessen die Ausführungszeit für jede mögliche Reihung eines Suchmodells und wählt jenes mit der minimalen Dauer aus. Die Ordnung dieses Algorithmus ist O(n) = n! für n Features im Suchmodell.

4.2.2 Implementierung

Von den 19 für die Wappendatenbank entwickelten Features wurden 15 im Reihungsalgorithmus implementiert. Auf das Objekt-Layout-Feature, das regionalisierte Farbhistogramm, das Siegel-Feature und das bedingte Farbhistogramm wurde verzichtet, da für diese Features nur schwer ein Bild einer Gruppe zugeordnet werden kann. Dieses Zuordnen eines Bildes zu einer Gruppe funktioniert folgendermaßen:

- Bei Features, deren Distanzfunktionen Ähnlichkeit auf einen rationalen Wertebereich abbilden, wird als Gruppe der Distanzwert zu einem feature-spezifischen Referenzvektor (beziehungsweise Orientierungspunkt) verwendet. Die folgende Tabelle zeigt einige typische Referenzvektoren für Wappen-Features:

Feature	Referenzvektor
Anzahl der Farbabstufungen	$(0)^t$
Farbhistogramm	$(16000, 8000, 4000, 2000, 1000, 0)^t$
Bild-Komplexität	$(0, 0, 0, 0, 0, 0)^t$

  Im Fall des Farbhistogramms erwies sich die Verwendung des Referenzvektors (abgeleitet aus den Farbhäufigkeiten in Wappen, vgl. Abschnitt 3.1.1.1) für die Zuordnung zu Gruppen dem Nullvektor überlegen.

- Für die anderen Features, deren Distanzfunktionen Ähnlichkeit auf eine diskrete Menge von Werten abbilden (z. B. Segmentierungs-Feature), wurde jeweils eine Funktion getClass() implementiert, die aus dem Featurevektor einen Identifier für die Gruppe ableitet.

Um im Reihungsalgorithmus beide Arten von Features (und Distanzfunktionen) verwenden zu können, wurde der Berechnung der Gruppe eine Funktion vorgeschaltet, die nach Art des Features entweder einen Distanzvergleich zum Referenzvektor oder getClass() ausführt. Abbildung 53 zeigt den Ablauf der Feststellung der Gruppe eines Bildes. Dazu wird zuerst das Distanzmaß eines Features geprüft und dann in Abhängigkeit vom Ergebnis entweder ein Distanzvergleich durchgeführt oder die Gruppe aus dem Featurevektor berechnet. Schließlich wird der Gruppenwert auf das Intervall [0, 100] standardisiert. Die Berechnung der Gruppe findet im Rahmen der Anpassung der Datenstruktur des Prognose-Moduls statt; also immer dann, wenn ein Bild zur Datenbank hinzugefügt oder aus ihr entfernt wird. Dazu wurde in die QBIC-Eingabeschnittstelle QbMkDbs ein entsprechender Aufruf eingebaut, der für die Durchführung dieser Datenpflegearbeiten sorgt.

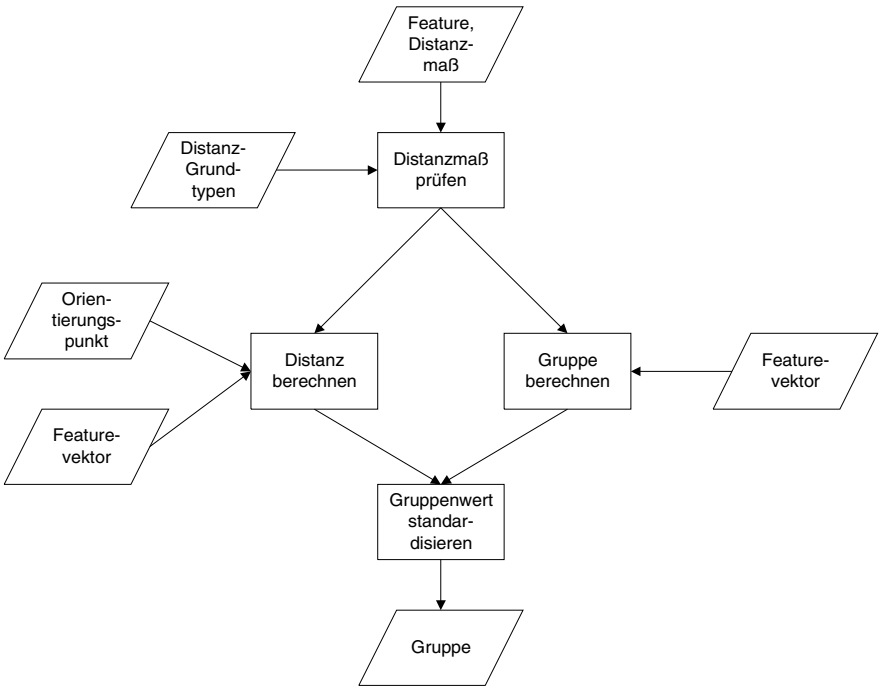

Abbildung 53: Datenfluß in der Funktion zur Feststellung der Gruppe eines Features

Die Berechnung der optimalen Reihung wurde in der C-Bibliothek für den Reihungsalgorithmus durch die Funktion getOptReihung() implementiert. Diese Funktion berechnet nacheinander für alle möglichen Permutationen eines Suchmodells die Ausführungszeit. Dazu wird folgendermaßen vorgegangen (Funktion calcReihung()):

1. Nach einer neuen Permutation wird für jedes Feature des Suchmodells der aus der Position im Modell folgende erwartete Wert für die Größe der Rückgabemenge berechnet.

2. Diese Werte werden zusammen mit den Performancewerten für die Features abgespeichert.

3. Die Ausführungszeit wird gemäß Gleichung 35 berechnet.

Die Performancewerte für die einzelnen Features wurden durch die Bildung des Mittelwerts für die Ausführungszeit der Distanzfunktionen für eine Vielzahl von Abfragen gewonnen. Diese Werte sind folglich rechner- und auslastungsabhängig. Da aber für die Bestimmung der optimalen Reihung durch die oben angegebene Zielfunktion nicht die absoluten Werte der p_i sondern nur die Relationen zwischen ihnen ausschlaggebend sind, ist diese Art der Performancebeurteilung ausreichend genau. Die folgende Tabelle zeigt die Performancewerte einiger Distanzfunktionen.

Feature	Distanzfunktion	Ordnung (bezüglich der Breite des Featurevektors)	Performancewert / Bild (ms)
Farbhistogramm	euklidische Distanz	O(n) = n	0.0901
Anzahl der Farbabstufungen	City Block	O(n) = 1	0.0450
Segmentierung	IF-Bedingung	O(n) = konstant	0.0676

Der Reihungsalgorithmus wurde in Form der Funktion getOptReihung() in der Suchmaschine für Suchmodelle (NetSrv) vor der Abarbeitung des Modells eingebaut. Zudem wird vom NetSrv neben der erwarteten die tatsächliche Ausführungszeit aufgezeichnet und in der Webschnittstelle im Statistikblock mit ausgegeben (vgl. Abbildung 54). Außerdem wird anhand der tatsächlich gefundenen Bilder eine Bewertung der Qualität der Vorhersage vorgenommen und geprüft, ob die angegebene Reihung tatsächlich die beste war. Im folgenden Abschnitt wird auf diese Tests und die erzielten Ergebnisse näher eingegangen.

4.2.3 Auswertungen

Die Qualität des Prognosealgorithmus liegt für alle Features im Schnitt über 80 Prozent, für die meisten Features werden Werte von mehr als 95 Prozent erreicht, was zum Teil aber auch daran liegt, daß diese Features sehr kurze Featurevektoren haben und daher die Bilder leicht in Gruppen eingeteilt werden können.

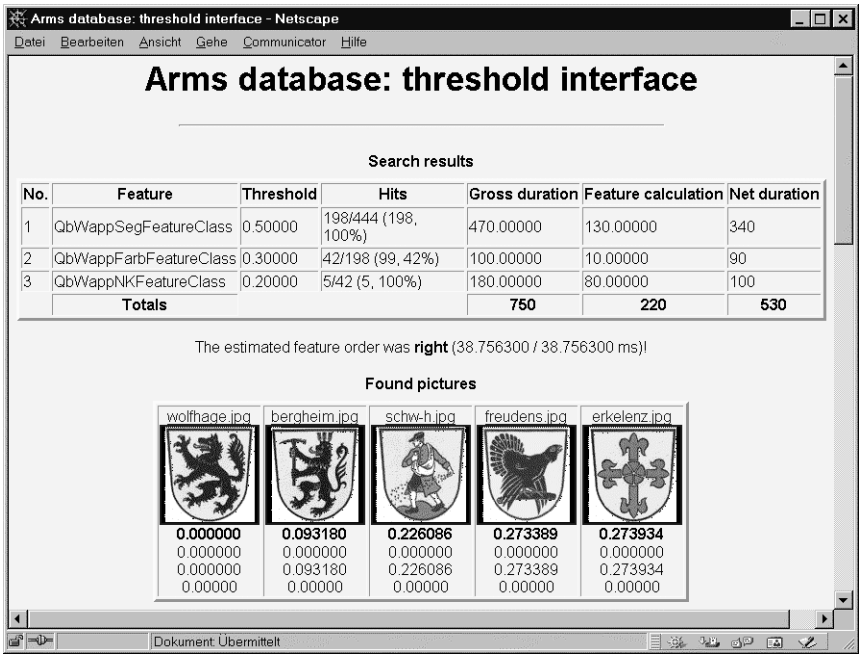

Abbildung 54: Screenshot mit Statistik für den Reihungsalgorithmus

In Abbildung 54 ist ein Suchmodell mit drei Features dargestellt, wobei in der Sektion "Search results" in der Spalte "Hits" Angaben zur Qualität der Prognose in der Form: "<tatsächliche Treffer>/<Ausgangsmenge> (<erwartete Treffer> %)" angegeben sind. Im angegebenen Beispiel war die Prognose für das Farbhistogramm (Klassenname: QbWappFarbFeatureClass) besonders schlecht, was jeweils zum Teil auf das spezifische Farbhistogramm des angegebenen Suchbildes sowie auf das nicht allzu genaue Relationen-Modell zurückzuführen ist. Die Reihung des Modells war dennoch richtig. Auswertungen ergaben, daß die meisten Fehler in der Reihung entweder auf die Vertauschung von zwei Features oder die falsche Einordnung eines einzelnen Features zurückzuführen sind. Reihungsfehler, bei denen mehr als zwei Features falsch eingeordnet werden, sind auch bei großen Suchmodellen selten. Die folgende Tabelle zeigt zwei typische Reihungsfehler.

Falsch	**Richtig**	**Fehlerbeschreibung**
- Regionalisierte Anzahl der Wappenfarben **- Anzahl der Wappenfarben** **- Symmetrietyp** - Inverse X-Achsen-Symmetrie	- Regionalisierte Anzahl der Wappenfarben **- Symmetrietyp** **- Anzahl der Wappenfarben** - Inverse X-Achsen-Symmetrie	Vertauschung zweier Features (Symmetrietyp und Anzahl der Wappenfarben)
- Diagonale Symmetrie - Symmetrietyp - Farbhistogramm **- Bild-Komplexität**	**- Bild-Komplexität** - Diagonale Symmetrie - Symmetrietyp - Farbhistogramm	Falsche Einordnung des Bild-Komplexitäts-Features

Die Auswertung des gesamten Reihungsalgorithmus (Prognose, Relationen, Optimierung) wurde durch ein Perl-Programm automatisiert, das realistische Suchmodelle mit zwei bis zehn Features erzeugt, diese durch die Suchmaschine NetSrv auswertet und die erhaltenen Statistiken zu einer Performanceanalyse aufbereitet. Dabei wurden jeweils vier mögliche Reihungsarten evaluiert:

- Zufällige Reihung durch einen Zufallsgenerator (Wahrscheinlichkeits-Methode).

- Reihung aufgrund der Heuristik der Performancewerte der Distanzfunktionen (Performance-Heuristik); das heißt Features mit schnellen Distanzfunktionen werden im Suchmodell vorne gereiht.

- Reihung aufgrund des Prognosemodells durch den Optimierungsalgorithmus (Prognosealgorithmus); das heißt es wird davon ausgegangen, daß die einzelnen Features voneinander unabhängig sind.

- Reihung aufgrund von Prognosemodell und Relationen-Modell durch den Optimierungsalgorithmus (Reihungsalgorithmus).

Abbildung 55 zeigt die Entwicklung der Qualität für diese vier Methoden bei steigender Anzahl der Schichten in den Suchmodellen. Dabei ist zu beachten, daß die Angaben kumuliert sind, das heißt die Qualität des Reihungsalgorithmus für Suchmodelle mit bis zu zehn Schichten ist der Mittelwert über die Qualität für Suchmodelle mit zwei, drei, etc. bis zehn Schichten. Unter Qualität wird dabei das Verhältnis von richtig gereihten Modellen zur Gesamtheit der untersuchten Modelle verstanden. Grundlage der folgenden Statistiken sind Auswertungen von tausend Suchmodellen je Reihungsmethode.

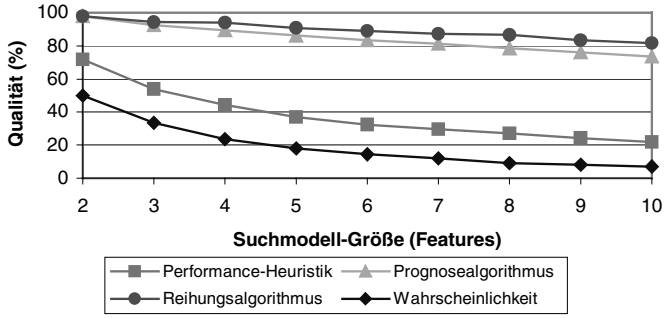

Abbildung 55: Entwicklung der Qualität des Reihungsalgorithmus

Die Qualität der Wahrscheinlichkeits-Methode nimmt den erwarteten Verlauf von ca. 50 Prozent bei Suchmodellen mit zwei Schichten bis zu nur ca. 5 Prozent bei Modellen von zwei bis zehn Schichten. Jene der Performance-Heuristik hat den selben Kurvenverlauf mit einem konstanten Abstand von ca. 20 Prozent, so daß ein Suchmodell mit zwei bis zehn Schichten voraussichtlich in 20 Prozent aller Fälle korrekt gereiht wird. Die Kurven des Prognose- und Reihungsalgorithmus zeigen einen bei weitem besseren (eher linearen) Verlauf, das heißt sie sind von der Modellgröße weniger abhängig. Die Qualität des Prognosealgorithmus sinkt von knapp 100 Prozent für Modelle mit zwei Schichten nur auf ca. 75 Prozent für Modelle mit zwei bis zehn Schichten. Jene des Reihungsalgorithmus liegt aufgrund der zusätzlichen Informationen über die Ähnlichkeit von Features noch ca. 8 Prozent über der des Prognosealgorithmus.

Neben der Qualität wurde auch gemessen, wie groß im Falle eines falsch gereihten Modells der Fehler für drei der vier verschiedenen Reihungsmethoden ist (vgl. Abbildung 56). Da diese Abweichungen für die Wahrscheinlichkeits-Methode beinahe beliebig sind, wurde diese Methode hier nicht betrachtet. Es zeigt sich, daß für die Performance-Heuristik der Fehler über die betrachteten Fälle annähernd linear ansteigt. Für den Prognose- und den Reihungsalgorithmus verläuft die Kurve quasi-exponentiell, so daß für einige wenige Fälle die Ausführung eines Suchmodells sogar länger dauert als bei der Performance-Heuristik, in der Mehrzahl der Fälle (mehr als 80 Prozent) der Fehler in der Ausführungszeit der Suche aber unter 10 Prozent liegt.

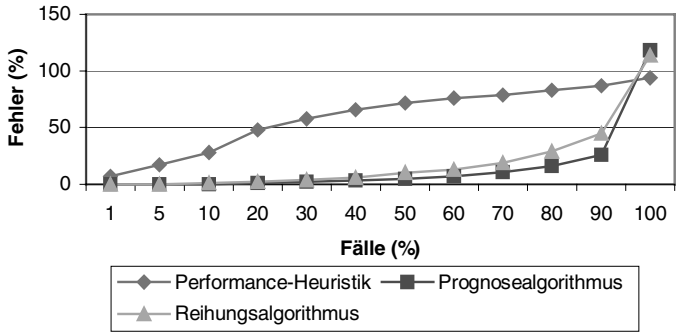

Abbildung 56: Ausmaß der Abweichungen bei Fehlern des Reihungsalgorithmus

Abbildung 57 zeigt nochmals vergleichend die Qualität der vier betrachteten Methoden. Daraus wird ersichtlich, daß durch die Verwendung des Reihungsalgorithmus ein Suchmodell (mit zwei bis zehn Features) in fast 82 Prozent aller Fälle richtig gereiht wird. Wird es dennoch falsch gereiht, so ist in ca. 80 Prozent der Fälle die Abweichung in der Ausführungszeit geringer als 10 Prozent.

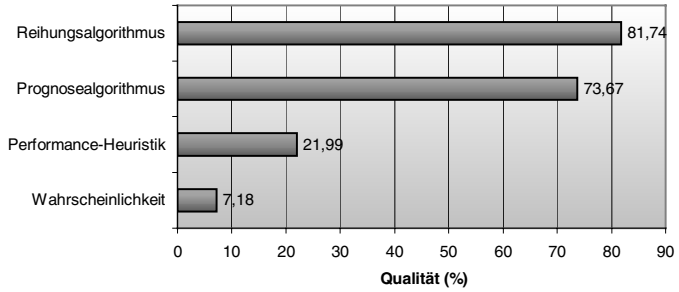

Abbildung 57: Qualitäts-Vergleichswerte für den Reihungsalgorithmus

Was bedeuten aber nun diese ausgezeichneten Werte für die Antwortzeiten des Systems, das heißt wie groß ist der durchschnittliche Gewinn des Benutzers bei der Ausführung einer Recherche durch die Anwendung des Reihungsalgorithmus? Auf den Statistiken für die Bearbeitungszeit der generierten Suchmodelle basierende Berechnungen haben ergeben, daß sich die Ausführungszeit einer durchschnittlichen Suche von 190.7 ms (bei Reihung aufgrund der Performance-Heuristik) auf 64.6 ms (bei Reihung durch den Reihungs-algorithmus) verringert. Das ist eine Verbesserung von 126.1 ms oder 66 Prozent der ursprünglichen Ausführungszeit. Im besten Fall, wenn die Reihung immer richtig wäre, würde die Ausführung einer durchschnittlichen Suche 58.7 ms dauern; das sind nur 6.1 ms (oder 3.2 Prozent von der ursprünglichen Ausführungszeit) weniger als beim Reihungsalgorithmus.

Der gewählte Ansatz für die performance-optimierte Reihung von Suchmodellen ist also sehr nahe am Optimum. Wenn man weiters berücksichtigt, daß diese Zeitangaben bereits die Kosten durch die zusätzlichen Berechnungen des Reihungsalgorithmus beinhalten und der Vergleich nicht mit einer Suchmethode ohne Schwellwerte oder einer zufälligen Reihung von

Suchmodellen gezogen wurde, wird deutlich, wie groß die Verbesserungen durch den Reihungsalgorithmus sind.

4.2.4 Zusammenfassung

In diesem Abschnitt wird ein Ansatz zur Nutzung des Konzepts der Suchmodelle für eine Steigerung der Performance von CBIR-Recherchen vorgestellt, der darauf beruht, Features mit schnellen Distanzfunktionen zur Vorauswahl und andere zur Feinauswahl so zu reihen, daß die Ausführungszeit einer Suche minimiert wird. Neben dem Konzept wird dargestellt, wie der Algorithmus implementiert und in die bestehende Testumgebung eingefügt wurde. Die Auswertungen zeigen, daß der Algorithmus die Ausführungszeit einer Suche auf maximal ein Drittel reduziert.

5 Automatische Generierung von Suchmodellen

In diesem Kapitel werden Algorithmen vorgestellt, mit deren Hilfe CBIR-Suchen weitgehend automatisiert durchgeführt werden können. Dabei wird grundsätzlich unterschieden, ob für den Suchvorgang ein oder mehrere Suchbilder zur Verfügung stehen, da je nach der Anzahl der Beispielbilder, die einen wesentlichen Teil der für die Suche zur Verfügung stehenden Informationen darstellen, unterschiedliche Strategien zur automatisierten Suche denkbar sind.

Die Vereinheitlichung der Ähnlichkeitsdefinition für CBIR-Suchen durch Suchmodelle bringt (in Kombination mit Schwellwerten für die Ergebnismengendefinition) einige wesentliche Vorteile: bessere Suchergebnisse (vgl. Abschnitt 3.2), Verringerung der Ausführungsdauer (z. B. durch den Reihungsalgorithmus, vgl. Abschnitt 4.2), Vereinfachung von Problemen des CBIR (Bestimmung der Gewichte, usw.), etc. Allerdings ergibt sich das Problem, daß die Modelle durch den Benutzer definiert werden müssen. Dieser kennt im allgemeinen jedoch weder die dem CBIR zugrunde liegenden Konzepte (Vektorraummodell, Distanzmaße zur Ähnlichkeitsdefinition, etc.) noch die Funktionsweise der einzelnen Features, so daß er, wenn man ihn nicht entsprechend schult und entsprechende Hilfen anbietet, mit der Aufgabe, ein Suchmodell zu definieren und eine Suche nach Bildinhalten zu initiieren, hoffnungslos überfordert ist. Das ist zwar kein Problem, das nur durch die Suchmodelle entsteht, die Suchschnittstellen anderer CBIR-Systeme sind meist ebenso kryptisch. Die Verwendung von Schwellwerten zur Begrenzung der Ergebnismenge verwirrt den Benutzers aber zusätzlich.

Wünschenswert wäre es, dem Benutzer einen möglichst einfachen Einstieg in ein Suchsystem zu ermöglichen; im Idealfall sollte er nur sein Suchbild angeben oder aussuchen müssen und schon eine Suche starten können. Das Ergebnis dieser ersten Abfrage sollte er dann durch sein Relevanzurteil und eine passende Methode zur Verfeinerung der Suche solange verbessern können, bis er den gewünschten Ausschnitt aus der Bilddatenbank gefunden hat (Click & Refine - Ansatz, vgl. auch Abbildung 58). Dieser benutzerfreundliche Ansatz für CBIR könnte die Akzeptanz solcher Systeme wesentlich heben. Im Rahmen dieser Arbeit wurde überlegt, wie man den ersten Schritt dazu, die Ableitung des ersten Suchmodells, realisieren könnte. Dazu wurde von zwei möglichen Szenarien ausgegangen:

- Der Benutzer gibt nur ein Suchbild an und sucht jene Bilder, die möglichst gut entsprechen. Es wird also mit einer "objektiven" (beziehungsweise technischen) Ähnlichkeitsdefinition gesucht.

- Der Benutzer gibt eine Gruppe von Bildern zur Suche an und möchte alle Bilder, welche die angegebene Reihe "verlängern". Da hier Ähnlichkeit durch den Benutzer fast willkürlich definiert werden kann, läßt sich von einer "subjektiven" Ähnlichkeitsdefinition sprechen. Diese Aufgabenstellung kommt teilweise dem Gedanken der Verwendung des

Relevanzurteils des Benutzers zur Suche sehr nahe, wobei der Benutzerwunsch nur durch positive Beispiele ausgedrückt wird.

Im folgenden Abschnitt wird ein Lösungsansatz für das erste Szenario vorgestellt, die Definition semantischer Gruppen wird im Abschnitt 5.2 behandelt. Dabei wird zur Gewichtung der Schichten der abgeleiteten Suchmodelle jeweils der vorgestellte Gewichtungsalgorithmus (vgl. Abschnitt 4.1) angewendet. In Abschnitt 5.3 wird evaluiert, wie sich die Qualität der Generierungsalgorithmen entwickelt, wenn zusätzliche Bilder zur Datenbank hinzugefügt werden.

5.1 Automatische Generierung von Suchmodellen aus einem Suchbild

Als Eingabedaten für ein Modell zur automatischen Ableitung von Suchmodellen stehen im hier betrachteten Fall nur die Bildeigenschaften (Features) sowie die Einordnung des Suchbildes innerhalb der Bilddatenbank (Clusterung der Datenbank, umgebende Bilder, etc.) zur Verfügung. Es liegt nahe, diese Informationen zur Ableitung von Suchmodellen zu verwenden.

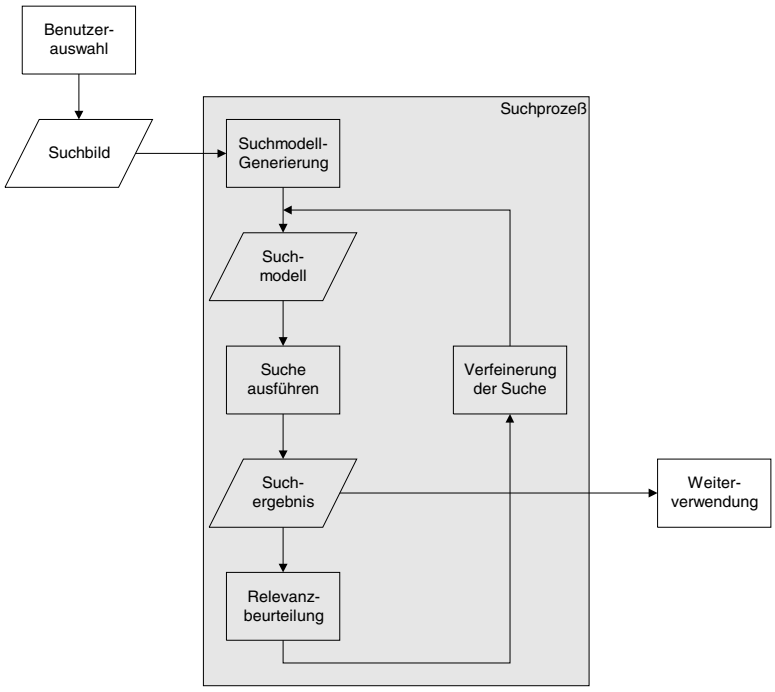

Abbildung 58: Datenfluß im Click & Refine - Suchprozeß

Abschnitt 5.1.1 beschäftigt sich mit dem grundsätzlichen Lösungsansatz, wobei für Features und Schwellwerte getrennt überlegt wird, wie man sie ableiten kann. Abschnitt 5.1.2 widmet sich der Implementierung der Generierungsalgorithmen in der QBIC-Testumgebung und dem Einbau in die Standard-Webschnittstelle. Dabei wird auch darauf eingegangen, für welche Features die vorgestellten Modelle implementiert wurden. Abschnitt 5.1.3 schließlich hat die Evaluierung der Algorithmen anhand der von ihnen abgeleiteten Suchmodelle zum Inhalt,

wobei zuerst die Methoden einzeln untersucht werden und dann ein Vergleich der verschiedenen Kombinationen mit den besten Ergebnissen, erzielt von einem Fachmann, unternommen wird.

5.1.1 Ableitung von Suchmodellen

Die automatische Ableitung von Suchmodellen umfaßt drei Bereiche:

- Auswahl der passenden Features. Im allgemeinen ist es nicht sinnvoll, alle verfügbaren Features für eine Suche zu verwenden. Die Nachteile sind vor allem schlechtere Performance der Suche, da sehr viele Einzelsuchen durchgeführt werden müssen, und (experimentell bestätigt, vgl. Abschnitt 5.1.3) schlechtere Suchergebnisse, da die Ähnlichkeitsdefinition teilweise irrelevante Elemente enthält.

- Bestimmung der Schwellwerte für die ausgewählten Features, so daß die Features, ihrer Bedeutung entsprechend, die passende Ergebnismenge aus der Ausgangsmenge sieben.

- Vergabe von Gewichten für die Features, um ihre relative Bedeutung zu bestimmen. Dieser Bereich wird im folgenden nicht mehr behandelt, da dafür der in Abschnitt 4.1 vorgestellte Gewichtungsalgorithmus verwendet wird.

Zur Bestimmung der Features und Schwellwerte stehen im wesentlichen folgende Arten von Daten zur Verfügung:

- Eigenschaften des Suchbildes. Die Featurevektoren der einzelnen Features beschreiben in einem guten System umfassend die verschiedenen Eigenschaften des Suchbildes. Eventuell wäre auch die Verwendung sogenannter Testfeatures denkbar, die nur dazu dienen würden, Aussagen über die Sinnhaftigkeit der Verwendung bestimmter Features zur Modell-Generierung zu machen. Aufgrund des Testfeaturevektors würde dann entschieden, welche Features im Suchmodell verwendet werden.

- Position des Suchbildes in der Datenbank. Aus der Clusterzugehörigkeit des Suchbildes und den Relationen dieses Clusters zu seinen Nachbarn sowie der Verteilung der Datenbank insgesamt lassen sich Aussagen über ihre Homogenität (relevant für die Wahl der Features), die voraussichtliche Größe der Ergebnismenge (relevant für die Bestimmung der Schwellwerte), etc. ableiten. Zu diesem Bereich gehört auch das in der Prognosedatenbank des Reihungsalgorithmus abgespeicherte Wissen.

- In den Generierungsalgorithmus eingebrachtes Expertenwissen. Dieses umfaßt unter anderem Wissen, das zur Beurteilung der Signifikanz der einzelnen Featurevektoren dient, Wissen, um aus der Clusterstruktur der Datenbank Erfolgswahrscheinlichkeiten für die Verwendung bestimmter Features abzuleiten, etc.

Folglich erscheinen folgende grundlegende Lösungsansätze zur Auswahl der Features passend: Auswahl aller Features, deren Featurevektoren für das Suchbild markante Werte beinhalten und beziehungsweise oder aller Features, die wesentlichen Anteil an der Bildung der natürlichen Clusterstruktur haben. Die Schwellwerte für die einzelnen Schichten werden entweder mithilfe von Expertenwissen oder aufgrund von Informationen über den Suchcluster abgeleitet. Außerdem können lineare Kombinationen der beiden Schwellwertansätze verwendet werden.

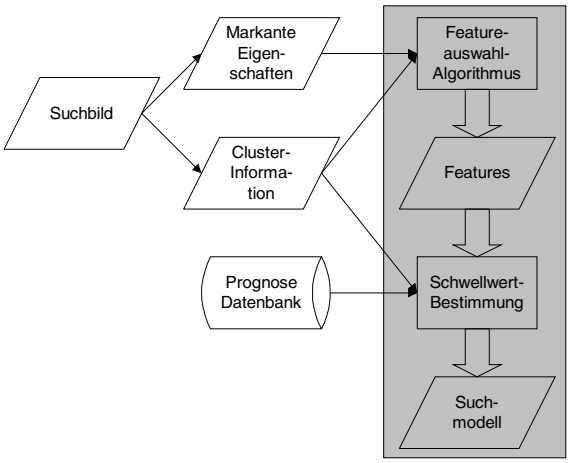

Abbildung 59: Datenfluß in den Feature- und Schwellwertalgorithmen

Abbildung 59 zeigt den Ablauf und Datenfluß im Generierungsmodell. Die Algorithmen sind als Rechtecke dargestellt, die Ein- und Ausgabedaten als Parallelogramme und die Prognose-Datenbank als Zylinder. Zuerst werden aufgrund der markanten Eigenschaften und der Clusterinformationen die Features ausgewählt und dann die Schwellwerte aufgrund der Clusterinformationen und der Prognose-Daten bestimmt.

5.1.1.1 Auswahl von Features

5.1.1.1.1 Eigenschaften-Methode

Bei der Eigenschaften-Methode erfolgt die Bestimmung der Features für ein Suchmodell aufgrund der markanten Eigenschaften des Suchbildes. Für jedes Feature wird eine Bedingung formuliert, die mit den Werten des Vektors evaluiert wird. Diese Bedingung legt fest, was für das betrachtete Feature ein außergewöhnlicher und damit untersuchenswerter Wert ist. Bedingung 37 bezieht sich auf das Feature zur Bestimmung der Farbabstufungen in einem Bild, das einen einstelligen Featurevektor ($F = (f_0)$) für die Anzahl der Farbabstufungen erzeugt.

$$f_0 < 4 \vee f_0 > 10 \xrightarrow{?} \textit{verwende Feature} \tag{37}$$

Das angegebene Beispiel bedeutet, daß das Farbabstufungen-Feature dann in einem automatisch generierten Suchmodell verwendet wird, wenn die Anzahl der Farbabstufungen

kleiner als 4 oder größer als 10 ist. Daraus wird ersichtlich, daß einerseits die Bestimmung der passenden Bedingung von der Implementierung des Features abhängig ist und andererseits die richtige Auswahl von Features sehr stark von einer geschickten Wahl der Auswahl-Bedingung abhängt.

5.1.1.1.2 Cluster-Methode

Zur Auswahl der Features aufgrund der Clusterstruktur der Datenbank werden die durch den Gewichtungsalgorithmus erzeugten Gewichte herangezogen. Das Gewicht eines Features wird aus seinem Beitrag zur Bildung der Clusterstruktur abgeleitet (vgl. Abschnitt 4.1). Es wird der Mittelwert über alle Gewichte gebildet und das Gewicht jedes einzelnen Features aufgrund der in Gleichung 38 dargestellten Bedingung verglichen.

$$w_f > g(\mu_w) \tag{38}$$

Hierbei ist w_f das Gewicht von Feature f, μ_w der Mittelwert über alle Gewichte und g() eine passende lineare Funktion. Für die praktische Anwendung in der Wappen-Testumgebung hat sich aufgrund zahlreicher Versuche folgende Funktion g() als nützlich erwiesen:

$$g(\mu_w) = 0.9\mu_w \tag{39}$$

Es werden alle Features benutzt, deren Gewicht zumindest 90 Prozent des Durchschnittsgewichtes beträgt. Versionen von g() mit einem konstanten Anteil führten zu schlechteren Versuchsergebnissen.

5.1.1.1.3 Andere Methoden zur Featureauswahl

Zusätzlich zur Verwendung der Eigenschaften- oder der Gewichte-Methode können alle Features verwendet werden, die wenigstens von einem Algorithmus ausgewählt wurden (kombinierte Featureauswahl). Schließlich können noch Verknüpfungen zwischen Features definiert werden, so daß ein Feature auch dann zu einem Suchmodell hinzugefügt wird, wenn es sich zwar nicht selbst qualifiziert hat, aber mit einem anderen Feature, das im Suchmodell verwendet wird, verknüpft ist. In der Testumgebung besteht eine solche Verknüpfung z. B. zwischen den Features zur Messung diagonaler Symmetrien, da Messungen ergeben haben, daß diese beiden Features sehr ähnliche Eigenschaften messen (vgl. Abschnitt 3.3.3).

5.1.1.2 Bestimmung der Schwellwerte

5.1.1.2.1 Prognose-Methode

Dieses Modell zur Bestimmung passender Schwellwerte für die ausgewählten Features betrachtet alle Features als gleichwertig und versucht daher, mit jedem Feature einen gleich großen Teil der Ausgangsmenge wegzuschneiden, um so zur passenden Ergebnismenge zu

gelangen. In Abbildung 60 wird dieser Gedanke dargestellt: die Rechtecke bezeichnen fünf Features, die sich überschneiden und von der (als rote Ellipse eingezeichneten) Ausgangsmenge je Feature etwa gleich viele nicht passende Bilder wegschneiden und so eine passende Ergebnismenge erzeugen.

Zur Bestimmung passender Werte für die Schwellwerte werden die Prognosedaten des Reihungsalgorithmus zusammen mit den Feature-Relationen verwendet. Aus der Verteilung der Bilder je Feature, Gruppe und Schwellwert wird zurückgerechnet, bei welchem Schwellwert eine passend große Ergebnismenge zurückbleibt. Die Bestimmung gleich großer Anteile für die Features ist nicht exakt, da die Auswirkungen der Ähnlichkeiten zwischen den Features im Relationenmodell nicht genau bestimmt werden, aber ausreichend gut, um handliche Ergebnismengen zu erzeugen. Der Hauptnachteil dieser Methode ist, daß letztlich die ungefähre Größe der Ergebnismenge in Relation zur Größe der Bilddatenbank angegeben werden muß, das heißt es muß festgelegt werden, auf welches Maß die Ausgangsmenge zu einer Ergebnismenge reduziert werden soll. Daraus folgt, daß bei dieser Methode nicht die volle Verbesserung der Ergebnisse durch die Verwendung von Schwellwerten zum Tragen kommt. Dennoch sind die Ergebnisse besser, als wenn überhaupt auf Schwellwerte verzichtet würde.

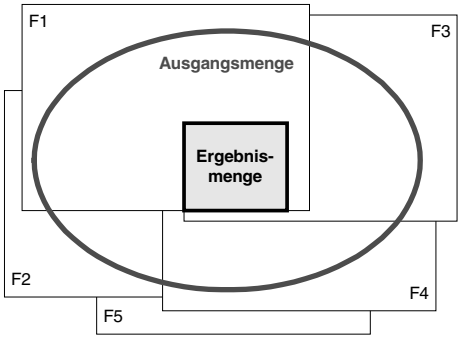

Abbildung 60: Schwellwertbestimmung durch die Prognose-Methode

5.1.1.2.2 Gewichte-Methode

Diese Methode zur Bestimmung von Schwellwerten beruht auf der Verwendung der Gewichte, dem Maß für die Bedeutung der Features. Dabei werden die Schwellwerte gemäß Gleichung 40 abgeleitet:

$$t_f^{SOM} = 1 - h\left(\frac{w_f}{\sum_{i=0}^{F} w_i}\right) \tag{40}$$

Hier sind t_f^{SOM} und w_f der Schwellwert beziehungsweise das Gewicht für Feature f und F die Anzahl der Features. h() ist eine passende lineare Funktion mit positiven Koeffizienten. Der Schwellwert für ein Feature ist also um so kleiner, je größer sein Gewicht ist, das heißt für

wichtige Features werden nur geringe Abweichungen vom Suchbild zugelassen. Da bei dieser Methode keine Untersuchung der voraussichtlichen Rückgabemengen stattfindet, ist es möglich, daß durch ein so generiertes Modell eine leere Ergebnismenge retourniert wird, wenn eben keine zum Suchbild ausreichend ähnlichen Bilder in der Datenbank vorhanden sind.

5.1.1.2.3 Kombinierte Schwellwertbestimmung

Neben den beiden Basismethoden können lineare Kombinationen als Schwellwerte benutzt werden (vgl. Gleichung 41).

$$t_f^{COMB} = \frac{t_f^{CONST} a + t_f^{SOM} b}{a + b} \tag{41}$$

Hierbei charakterisieren die Parameter a und b die Gewichtung der Basismethoden. Im Rahmen der durchgeführten Tests wurden verschiedene Verhältnisse von a und b ausprobiert (vgl. Abschnitt 5.1.3).

5.1.2 Implementierung

Die Algorithmen zur Generierung von Suchmodellen wurden für alle Features implementiert, die auch im Reihungs- und Gewichtungsalgorithmus implementiert sind, da Informationen dieser Module zur Herleitung der Modelle verwendet werden. Die einzelnen Funktionen wurden in einer eigenen Funktionsbibliothek abgelegt, in der auch die Bedingungen für markante Eigenschaften und die Verknüpfungen zwischen Features definiert sind. Die Herleitung der Modelle erfolgt durch die Funktion doFeatureTest(), der über Parameter angegeben wird, welche Methoden zur Feature- und welche zur Schwellwertbestimmung verwendet werden sollen; die einzelnen Teilalgorithmen sind durch eigene Funktionen realisiert. Nach der Herleitung eines Suchmodelles erfolgt die automatische Bestimmung passender Gewichte und abschließend die performance-optimierte Reihung durch den Reihungsalgorithmus.

In der Webschnittstelle für die Suche (vgl. Abbildung 61) wurden (während der Testphase) in der Sektion Suchmodell-Definition Auswahlboxen für die Auswahl der gewünschten Teilalgorithmen hinzugefügt. Nachdem eine Suche durch die Auswahl eines Suchbildes initiiert wurde, wird in der Ergebnisseite ein vorgeschlagenes Suchmodell ausgegeben, das durch Klicken auf einen angegebenen Link evaluiert werden kann. Im angegebenen Beispiel wurde die Verwendung eines Modells mit fünf Schichten vorgeschlagen (Tabelle "Suggested query model"), wobei vier Farbfeatures und das Segmentierungs-Feature gewählt wurden (Spalte 2). Das regionalisierte Farbenanzahl-Feature wird übrigens nur deshalb im Modell benutzt, weil es mit dem (ähnlichen) regionalisierten Farbabstufungen-Feature verknüpft ist

(Eintrag "r v" in Spalte 5). Die anderen Features wurden aufgrund ihrer markanten Eigenschaften ausgewählt (Eintrag "r a" in Spalte 5).

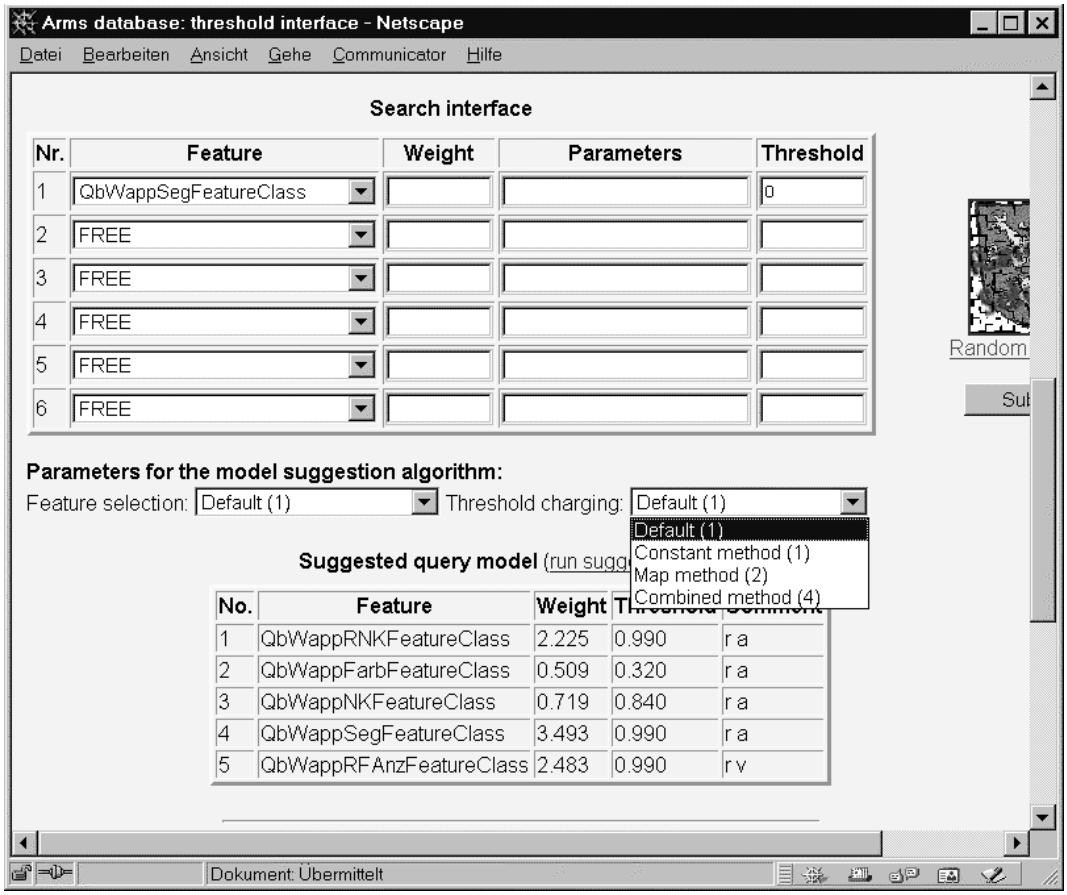

Abbildung 61: Suchmaske für die Auswahl von Generierungsmethoden für Suchmodelle

5.1.3 Auswertungen

Die Methoden zur Ableitung von Suchmodellen wurden anhand von Precision und Recall beurteilt. Dabei wurde für jede mögliche Kombination von Einzelmethoden eine Serie von Testsuchen durchgeführt. Dazu wurde die erweiterte Webschnittstelle der Testumgebung (vgl. Abschnitt 3.1.3) benutzt. Im folgenden werden zuerst die Einzelergebnisse für die Methoden zur Featureauswahl und Schwellwertbestimmung betrachtet und anschließend die besten Kombinationen mit den von einem menschlichen Experten erzielten Ergebnissen verglichen.

Um zu Aussagen über die Qualität der Featureauswahl-Methoden zu gelangen, wurde jede Methode mit jeder Schwellwertmethode kombiniert und getestet. Danach wurden für jede Featureauswahl-Methode die Mittelwerte für Precision und Recall über alle Schwellwert-Bestimmungs-Methoden berechnet. Abbildung 62 zeigt die Mittelwerte von Precision und Recall für die Eigenschaften-, die Cluster-Methode und die kombinierte Featureauswahl im Vergleich zur Benutzung aller Features für jede Suche.

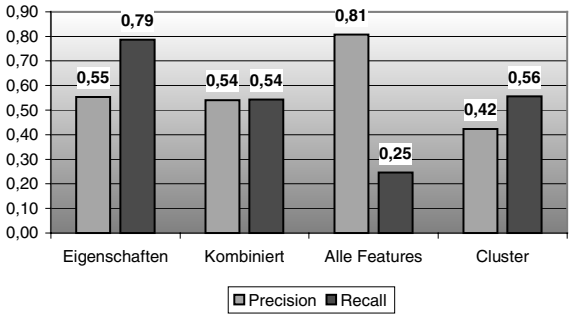

Abbildung 62: Ergebnisse der Feature-Methoden

Dabei erweist es sich, daß bei der Eigenschaften-Methode von allen der mit Abstand beste Recall erreicht wird, jener der Cluster-Methode ist um 23 Prozent schlechter und Kombinationen führen eher zu einem noch schlechteren Ergebnis. Werden alle Features zur Suche benutzt, ergibt sich zwar eine Ergebnismenge mit relativ hoher Precision, der Recall liegt jedoch in einem für eine Datenbank mit künstlichen Bildern nicht akzeptablen Bereich.

Abbildung 63 zeigt Precision und Recall für die vorgestellten Schwellwert-Berechnungsmethoden. Evaluiert wurde die Prognose-Methode, die Gewichte-Methode und die kombinierte Schwellwertbestimmung.

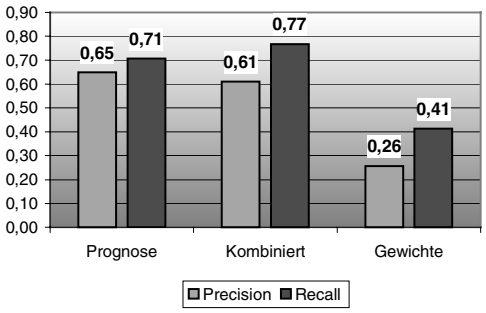

Abbildung 63: Ergebnisse der Schwellwert-Methoden

Hier erwiesen sich die Prognose-Methode und die kombinierte Schwellwertbestimmung als etwa gleich gut, der höhere Recall der letzteren wird durch eine höhere Precision der ersteren ausgeglichen. Die Gewichte-Methode erzielt keine akzeptablen Werte. Nachdem festgestellt wurde, daß die Verwendung der kombinierten Schwellwertbestimmung zu einem etwas höheren Recall führt, wurde versucht, das optimale Verhältnis der Parameter a und b (vgl. Gleichung 41) festzustellen, wobei sich für die Wappen-Testumgebung ein optimales Verhältnis von a:b = 7:6 ergab.

Nach der Evaluierung der Basismethoden anhand der Mittelwerte von Precision und Recall wurden die verschiedenen Kombinationen der Einzelmethoden untersucht. Die besten fünf Kombinationen sind in Abbildung 64 dargestellt. Da hier nicht mehr die Mittelwerte über verschiedene Basismethoden zur Feature- oder Schwellwertbestimmung betrachtet werden,

sind die Ergebnisse signifikant besser. Als beste Kombination erwies sich die Verwendung der Eigenschaften-Methode zur Featureauswahl und der kombinierten Schwellwertbestimmung; dadurch wird ein Recall von 94 Prozent bei einer Precision von mindestens 68 Prozent erzielt. Als zweitbeste Kombination ergab sich Eigenschaften-Methode und Prognose-Methode mit einer Precision von 75 Prozent aber einem etwas geringeren Recall von 80 Prozent. Die Bedeutung der Miteinbeziehung von Clusterinformation zur Schwellwertbestimmung, die bei der Evaluierung der Basismethoden noch nicht klar zu Tage getreten war, kommt damit deutlich zum Ausdruck. Dadurch wird der Recall um 15 Prozent erhöht. Als drittbeste Methode ergab sich die Verwendung von allen Features mit kombinierten Schwellwerten. Dabei wird eine Precision von 100 Prozent erzielt; der Recall liegt aber mit nur 38 Prozent weit unterhalb der Featureauswahl nach Bedingungen. Alle anderen Methoden erzielen Ergebnisse, die weit unter für den Benutzer akzeptablen Grenzen liegen.

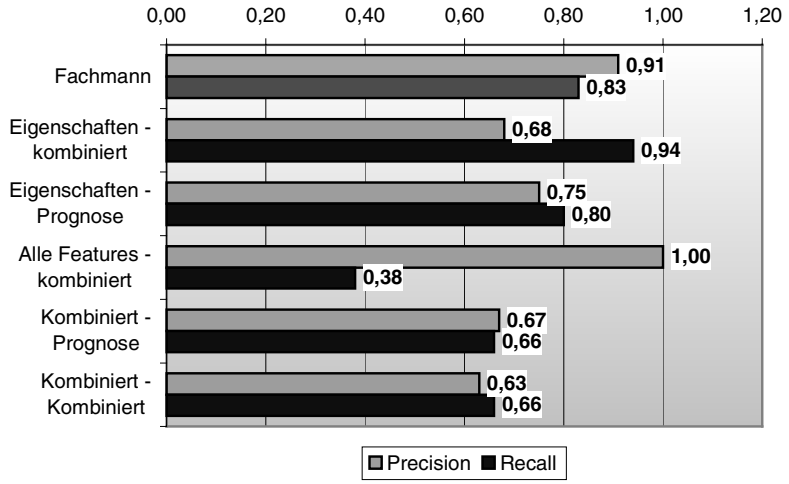

Abbildung 64: Qualität der Algorithmen zur Ableitung von Suchmodellen aus einem Suchbild

Abschließend wurde untersucht, wie gut die generierten Modelle im Vergleich zu einem Fachmann abschneiden. Dazu wurde folgendermaßen vorgegangen: es wird zuerst ein Suchmodell generiert, das der menschliche Experte dann so weit verändert und verfeinert, bis das seiner Meinung nach optimale Ergebnis erreicht ist. In Abbildung 64 ist dargestellt, daß sich so im Schnitt eine um 23 Prozent bessere Precision erreichen ließ, wobei eine Verminderung des Recalls um 11 Prozent in Kauf genommen wurde. Wollte man den Recall gleich halten, so ließe sich immerhin eine Steigerung der Precision von ca. 10 Prozent erreichen. Die automatischen Methoden zur Modellgenerierung erzielen also kaum schlechtere Ergebnisse, als ein Experte erreichen könnte.

5.1.4 Zusammenfassung

In diesem Abschnitt werden Überlegungen dazu angestellt, wie die an sich guten, in der Formulierung aber komplizierten Suchmodelle automatisch aus den Eigenschaften eines Suchbildes hergeleitet werden können. Dabei wird an die Einbettung eines solchen Ansatzes in ein System gedacht, das es dem Benutzer erlaubt, eine Suche durch einfaches Klicken auf ein Bild zu starten und ein bereits so gutes Ergebnis zu erhalten, daß die Verfeinerung durch iteratives Annähern mithilfe des Relevanzurteils des Benutzers in wenigen Schritten erfolgen kann.

Es wird überlegt, welche Daten für die Generierung zur Verfügung stehen und passende Algorithmen entworfen, die in der QBIC-Testumgebung realisiert werden können. Die Evaluierung dieser Methoden ergibt eine eindeutige Reihung der Methoden und für die besten Kombinationen, im Vergleich zu einem menschlichen Experten, ansprechende Ergebnisse.

5.2 Automatische Generierung von Suchmodellen aus mehreren Beispielbildern

Nachdem im letzten Abschnitt überlegt wurde, wie Suchmodelle aus den Informationen eines Suchbildes abgeleitet werden können, wird in diesem Abschnitt ein Ansatz dargestellt, Suchmodelle aufgrund der Eigenschaften von semantischen Gruppen von Bildern abzuleiten. Unter einer semantischen Gruppe soll dabei eine (beinahe) willkürliche Auswahl von Bildern aus der Grundmenge durch den Benutzer verstanden werden. Dieser Ansatz kommt dem Gedanken der Verwendung von Relevanzinformation nahe, wo auch versucht wird, Informationen für die Suche aus dem Relevanzurteil des Benutzers abzuleiten. Im vorgestellten Algorithmus werden allerdings nur positive Gruppenbeispiele zugelassen, da es bei einer zu erwartenden großen Grundmenge und einer im Vergleich dazu sehr kleinen semantischen Gruppe wenig Sinn macht, Informationen über Bilder, die nicht zur Gruppe gehören (von denen es sehr viele gibt), mit einzubeziehen.

Aus den Beispielen, die ein Benutzer zur Definition einer semantischen Gruppe angibt, soll einerseits eine Beschreibung der semantischen Gruppe und andererseits ein oder mehrere Suchmodelle zum Finden aller Bilder, die zur Gruppe gehören, abgeleitet werden. Der Benutzer erhält also die Möglichkeit, durch die Auswahl von Bildern eine subjektive Ähnlichkeit so zu definieren, daß das Suchsystem dazu in der Lage ist, die Liste der angegebenen Beispiele aufgrund der Bilder in der Grundmenge im Sinne dieser Ähnlichkeitsdefinition fortzusetzen.

Im folgenden Abschnitt 5.2.1 wird der grundsätzliche Lösungsansatz zur Bestimmung der Eigenschaften einer semantischen Gruppe sowie zur Ableitung passender Suchmodelle dargestellt, in Abschnitt 5.2.2 wird auf die Implementierung des Lösungsalgorithmus eingegangen, es wird unter anderem erklärt, wie der Benutzer eine semantische Gruppe bestimmen kann, wie die Suchmodelle abgeleitet werden und wie die Suche mithilfe dieser Suchmodelle erfolgt. Abschnitt 5.2.3 schließlich beschäftigt sich mit Auswertungen: es werden Beispiele für semantische Gruppen gezeigt, die Qualität des Algorithmus anhand von Precision und Recall beurteilt und allgemeine Schlußfolgerungen zur Problematik der Ähnlichkeitsdefinition aufgrund von Gruppen von Bildern gezogen.

5.2.1 Ableitung von Suchmodellen für semantische Gruppen

Da die Ableitung eines Suchmodells aus einem einzigen Bild nur die Umsetzung einer allgemeinen Ähnlichkeitsdefinition erlaubt, die unter Umständen der Ähnlichkeitsauffassung des Benutzers nicht entspricht, wurde durch die Definition semantischer Gruppen von Bildern versucht, dem Benutzer ein Werkzeug in die Hand zu geben, das es ihm ermöglicht, besondere Ähnlichkeit einfach und auf intuitive Weise definieren zu können. Eine semantische Gruppe kann dabei eine willkürliche Zusammenstellung von Bildern sein; um

gute Suchergebnisse zu erzielen, ist es aber sinnvoll, Bilder mit einem semantischen Zusammenhang zu wählen. Eine solche semantische Gruppe könnte in der Wappen-Testumgebung beispielsweise die Gruppe der bayrischen Gemeindewappen sein, denen allen das bayrische Hoheitszeichen in einem Feld des Schildes gemein ist (vgl. Abbildung 69).

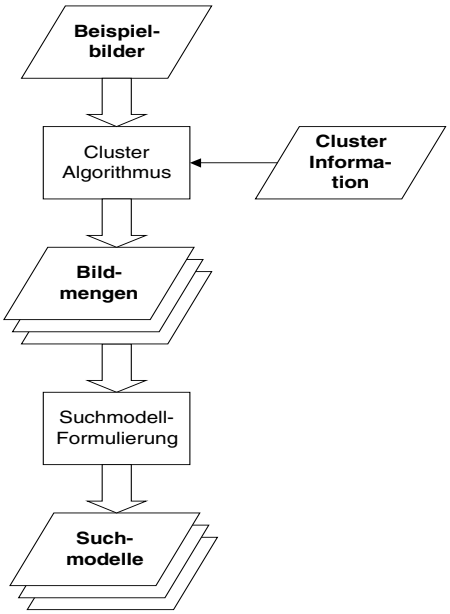

Abbildung 65: Datenfluß im Algorithmus zur Ableitung
von Suchmodellen aus mehreren Beispielbildern

Um anhand einer Gruppe von Beispielbildern eine Suche durchführen zu können, muß ein Algorithmus entworfen werden, der die Eigenschaften von Bildgruppen feststellt und daraus Suchanfragen ableiten kann. Der im folgenden dargestellte Algorithmus zur Bearbeitung semantischer Gruppen erbringt die folgenden Leistungen:

- Beschreibung der semantischen Gruppe: es wird durch eine SOM festgestellt, in welche natürlichen Cluster die angegebene semantische Gruppe zerfällt und - je Cluster - für jedes Feature Mittelwert und Varianz der Distanz der Beispielbilder zueinander berechnet. Daraus läßt sich ableiten, in welchen Eigenschaften die semantische Gruppe übereinstimmt und wenn nicht, wie groß die durchschnittliche Abweichung ist.

- Ableitung einer Gruppe von Suchmodellen, um alle Bilder in der Datenbank zu finden, die zur Gruppe gehören. Es wird für jeden Cluster ein Suchmodell erzeugt.

Abbildung 65 zeigt den Datenfluß bei der Bearbeitung semantischer Gruppen. Zuerst werden anhand der Clusterinformationen der Bilddatenbank die Beispielbilder der semantischen Gruppe in Cluster eingeteilt. Sodann wird für jede Gruppe von Beispielbildern ein Suchmodell erzeugt und ein passendes Suchbild ausgewählt. Dadurch wird die Verwendung der

Standard-Suchmaschine für Suchmodelle ermöglicht. Im folgenden ist dargestellt, wie die einzelnen Anforderungen an den Algorithmus umgesetzt wurden.

Der Algorithmus zur Behandlung semantischer Gruppen verwendet zur Feststellung der Clusterung der Beispielbilder die durch eine Self-organizing Map für den Gewichtungs-algorithmus erzeugten Clusterinformationen der Bilddatenbank (vgl. Abschnitt 4.1.2). Für jeden Cluster wird dann folgendes gemacht:

- Auswahl eines passenden Suchbildes. in diesem Ansatz zur Bearbeitung semantischer Gruppen soll eine Suche letztlich wieder durch eine Menge von Suchmodellen erfolgen. Dazu ist es notwendig, für jede Teilsuche ein möglichst passendes Suchbild auszuwählen. Im vorgestellten Ansatz wird jeweils das Zentroid der Beispielbilder eines Clusters genommen. Dazu wird die Distanz (über alle Features) von jedem Bild zu jedem anderen gemessen und jenes Bild ausgewählt, dessen Distanzsumme minimal ist (vgl. Abbildung 66). Sind nur zwei Beispielbilder vorhanden, wird zufällig eines ausgewählt.

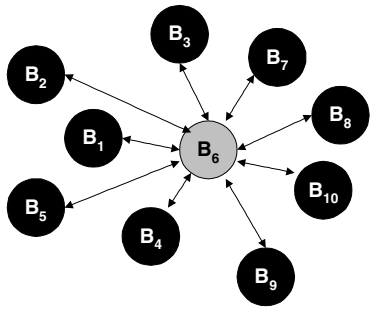

Abbildung 66: Auswahl des Suchbildes

In Abbildung 66 bezeichnen die Kreise die Beispielbilder in einem Cluster, der Kreis B_6 bezeichnet das Zentroid. Die Pfeile zwischen den Bildern repräsentieren die Distanzwerte.

- Herleitung eines passenden Suchmodells. Gibt es zu einem Cluster nur ein Beispielbild, so werden die Methoden der Generierung aus dem Suchbild verwendet (vgl. Abschnitt 5.1). Dabei werden die Features nach der Eigenschaften-Methode ausgewählt und die Schwellwerte mit der kombinierten Methode bestimmt (Methode mit den besten Ergebnissen). Jedes Teilmodell soll nur eine Ergebnismenge von ca. 1 Prozent der Ausgangsmenge produzieren. Dadurch wird die Precision der Gesamtabfrage (über alle Teilmodelle) erhöht.

  Stehen in einem Cluster zwei oder mehr Bilder zur Verfügung, wird zur Bildung eines passenden Suchmodells folgendermaßen vorgegangen: es werden alle Features benutzt, deren Mittelwert der Distanz über alle Bilder einen bestimmten, vom Distanzmittelwert über alle Features und Beispielbilder des Clusters abhängigen Wert nicht überschreitet (vgl. Bedingung 42):

$$\mu_f < k(\mu_F) \tag{42}$$

In Bedingung 42 ist μ_f der Distanzmittelwert für Feature f und μ_F der Distanzmittelwert über alle Features. k() ist eine passende lineare Funktion. In der Praxis hat es sich als günstig erwiesen, folgende Bedingung zu benützen:

$$\mu_f < 0.1\mu_F + 0.05 \tag{43}$$

Das heißt der Mittelwert der Distanz eines Features darf maximal ein Zehntel der durchschnittlichen Abweichung betragen, wobei eine Grunddistanz von 5 Prozent (da alle Features auf [0, 1] normiert sind!) akzeptiert wird. Der Schwellwert eines ausgewählten Features wird durch die in Gleichung 44 dargestellte Funktion bestimmt:

$$t_f = l(\mu_f, c_f) \tag{44}$$

Hier ist t_f der Schwellwert von Feature f, μ_f ist wieder der Mittelwert über alle Distanzen für Feature f und σ_f die Varianz über alle Distanzen. l() ist wieder eine passende lineare Funktion, wobei sich die in Gleichung 45 dargestellte Variante in einer Vielzahl von Tests als die beste erwiesen hat:

$$t_f = 1.5\mu_f + 0.01\sigma_f + 0.1 \tag{45}$$

Die Koeffizienten von Gleichung 45 bedeuten, daß der Mittelwert der Distanz einen großen Einfluß auf den Schwellwert hat, während jener der Varianz minimal ist. Eine Abweichung von lediglich 10 Prozent wird jedenfalls erlaubt. Da bei dieser Methode zur Bestimmung der Schwellwerte nicht gewährleistet ist, daß überhaupt passende Bilder in der Datenbank gefunden werden, wird das so erzeugte Modell nur als Basismodell verwendet. Wenn während einer Suche klar wird, daß die Ergebnismenge leer sein wird, wird die Ausführung der Suche abgebrochen und statt dessen ein sogenanntes Alternativmodell evaluiert, das wiederum aufgrund des oben angegebenen Generierungsalgorithmus (Featureauswahl durch die Eigenschaften-Methode und kombinierte Schwellwert-Bestimmung) aus dem Suchbild erzeugt wird. In der Testphase hat sich erwiesen, daß die Ergebnismenge des Basismodells zwar nur selten leer ist, die Verwendung des Alternativmodells Precision und Recall des Gesamtergebnisses aber doch verbessert, so daß es nicht sinnhaft erscheint, auf die Verwendung des Alternativmodells zu verzichten.

Nachdem sich in der Testphase gezeigt hat, daß das Hauptproblem dieser Methode das Erreichen einer akzeptablen Precision ist, wurde versucht, im Algorithmus explizit Unähnlichkeit zu berücksichtigen. In der Suchmaschine für Suchmodelle ist es möglich, durch die Angabe negativer Schwellwerte alle Bilder einer Datenbank zu finden, die für das

gewählte Feature eine größere Distanz als den absoluten Wert des angegebenen Schwellwertes haben. Es wurde versucht, diese Funktionalität in einem entsprechenden Algorithmus zu nutzen, wobei sich zwar eine Verbesserung der Precision ergab, der Recall aber gleichzeitig überproportional zurückging. Daher wurde bei den unten angegebenen Tests auf die Berücksichtigung von Unähnlichkeit verzichtet.

Ein weiteres Problem des Algorithmus ist seine Komplexität: für die Bestimmung der Eigenschaften von n Bildern in einem Cluster ist die Ordnung O(n) = n!. Da normalerweise aber nur wenige Bilder zu einem bestimmten Cluster gehören, ist dieses Problem in der Praxis kaum relevant.

5.2.2 Implementierung

Der Algorithmus für semantische Gruppen, in der Testumgebung als eigene C/C++-Bibliothek realisiert, wurde für alle Features außer dem bedingten Farbhistogramm implementiert, da der Featurevektor dieses Features mit dem des Tinktur-Histogramms übereinstimmt. Allerdings werden nur beim Basismodell alle verbleibenden 18 Features genutzt, bei der Generierung des Alternativmodells durch das Suchbild werden nur die Features verwendet, die für den Generierungsalgorithmus implementiert wurden (vgl. Abschnitt 5.1.2). Da die Suchmaschine NetSrv (vgl. Abschnitt 3.2.5.2) nur ein einzelnes Suchmodell evaluieren kann, wurde eine Meta-Suchmaschine entwickelt, die die Einzelergebnisse der NetSrv-Abfragen zu einer globalen Ergebnismenge verschmilzt. Abbildung 71 zeigt einen typischen Screenshot einer Ergebnisseite.

Bei der Entwicklung einer passenden Webschnittstelle für die Verarbeitung semantischer Gruppen mußten folgende Probleme gelöst werden:

- Wie sollte die Definition der semantischen Gruppe erfolgen? Es mußte ein Weg gefunden werden, einerseits die Grundmenge benutzeradäquat anzuordnen und andererseits die bereits ausgewählten Bilder übersichtlich darzustellen.

- Wie sollten die Eigenschaften einer semantischen Gruppe dargestellt werden?

- Welche Möglichkeiten sollte es zur Evaluierung der generierten Suchmodelle geben?

Zur Lösung des ersten Problems wurde ein ikonischer Index verwendet (vgl. Literaturüberblick). Dieser wurde aufgrund jener Bilder gebildet, die in der Bilder-SOM am nächsten zum Mittelwertvektor liegen. In der Wappen-Testdatenbank war eine flache Struktur ausreichend, für größere Datenbanken wäre aber vermutlich eine hierarchische besser geeignet. Abbildung 67 zeigt eine typischen Screenshot der Auswahlmaske. In der ersten Sektion werden die bereits ausgewählten Bilder angezeigt, in der zweiten der Inhalt des aktuellen Clusters und in der dritten der ikonische Index; die Bilder werden, abhängig vom Zweck der Sektion, in unterschiedlichen Größen dargestellt. Der Nachteil dieser Methode ist,

daß viele Gruppen über mehrere Cluster verstreut sind und es manchmal nicht ganz einfach ist, die gewünschten Bilder sofort zu finden.

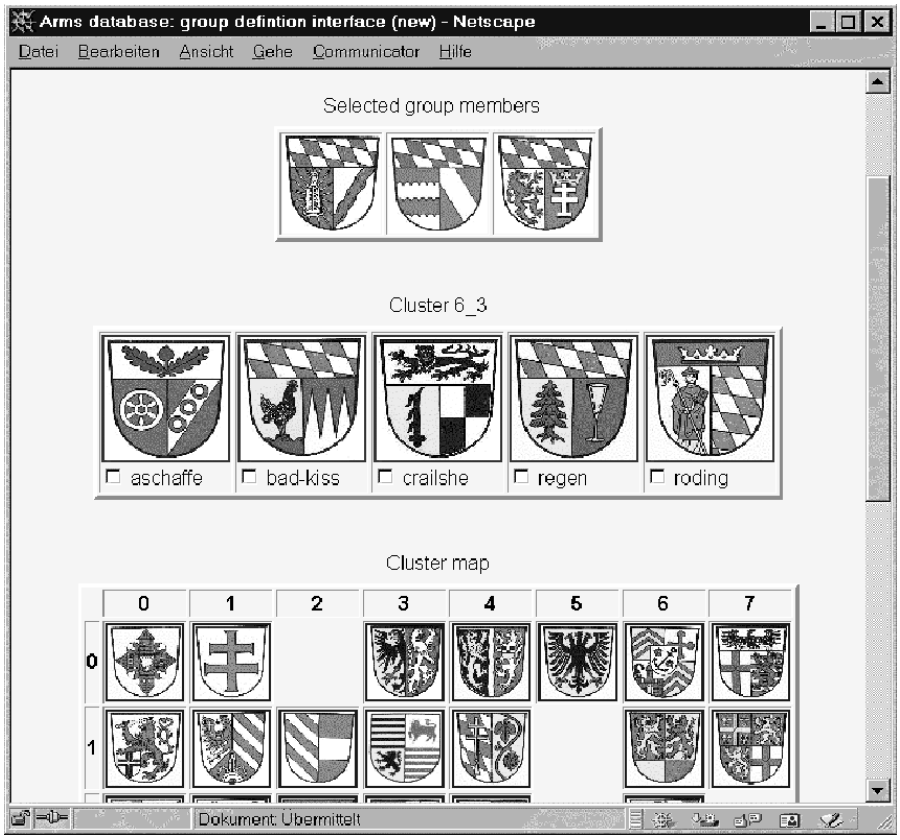

Abbildung 67: Benutzerschnittstelle für die Bildung semantischer Gruppen

Die Anzeige der Eigenschaften einer semantischen Gruppe erfolgt, nachdem in der Auswahlmaske der Submitbutton gedrückt wurde. Es wird je Cluster eine zweispaltige Ausgabe erzeugt: in der ersten Spalte werden untereinander die zu diesem Cluster gehörenden Beispielbilder sowie die Eigenschaften dieser Untergruppe anhand von Mittelwert und Varianz der Distanzen zwischen den Bildern dargestellt (vgl. Abbildung 68). In der zweiten Spalte werden das Basis- und das Alternativmodell dargestellt. Nach jedem Cluster folgt eine Zeile mit Auswahlbuttons. Dadurch können folgende Funktionen aktiviert werden:

- Anzeige der globalen Eigenschaften der semantischen Gruppe. Es erfolgt vor der Berechnung der Distanzwerte keine Zerlegung in Cluster. Dadurch lassen sich sehr schön die Gemeinsamkeiten der semantischen Gruppe ablesen: gemein ist eine Eigenschaft (Feature) allen Bildern der Gruppe dann, wenn der Distanzmittelwert für dieses Feature gering ist. Hier kommt aber die ungünstige Ordnung des Algorithmus leider stark zum Tragen, das heißt die Antwortzeiten sind bei semantischen Gruppen von zehn und mehr Bildern in der Testumgebung eher hoch.

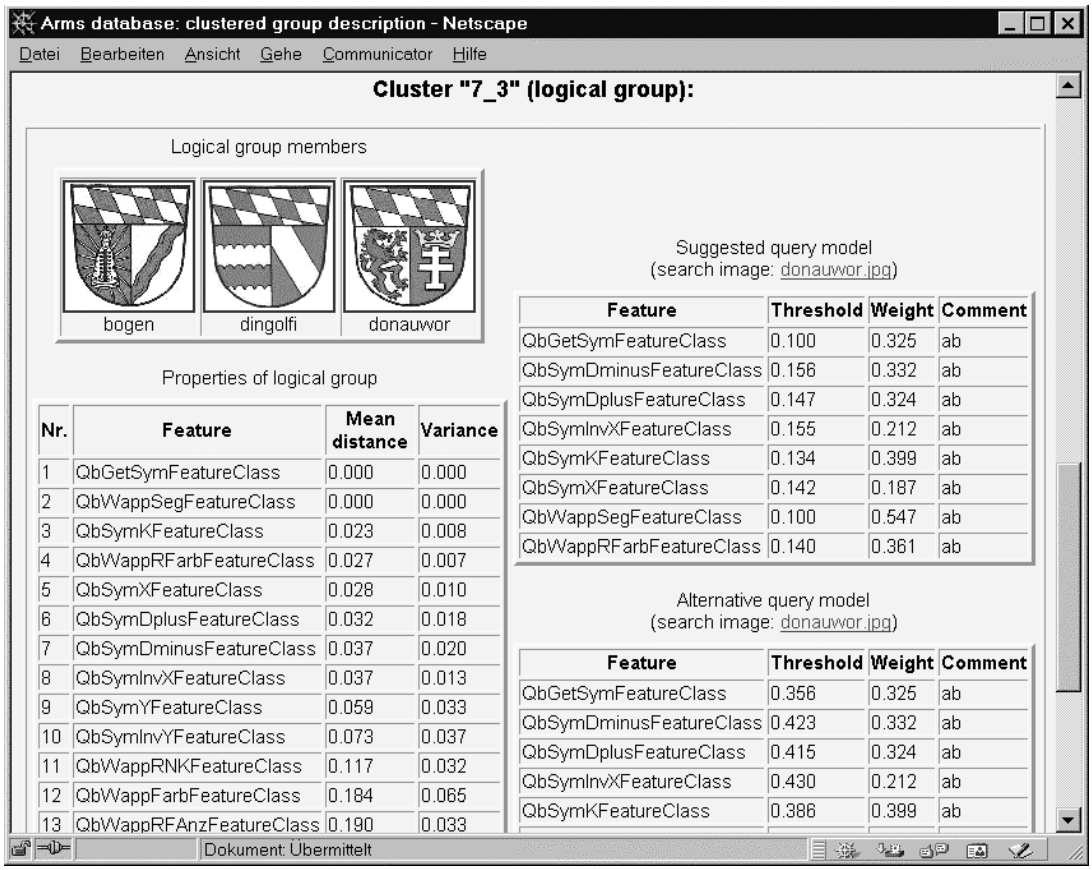

Abbildung 68: Screenshot einer Gruppenbeschreibung

- Evaluierung des Suchmodells für diesen Cluster. Es wird nur das Basismodell und eventuell das Alternativmodell ausgeführt. Anhand der Ergebnismenge kann man den Beitrag eines Suchmodells zur globalen Ergebnismenge beurteilen.

- Evaluierung aller Suchmodelle und Bildung einer globalen Ergebnismenge. Dazu wird die Perl-basierte Meta-Suchmaschine benutzt, die neben einer globalen Ergebnismenge auch Statistiken und Ergebnisse je Teilmodell erzeugt (vgl. Abbildung 71).

5.2.3 Auswertungen

Im folgenden werden die Tests beschrieben, die zur Evaluierung des oben angegebenen Algorithmus für semantische Gruppen durchgeführt sowie die Ergebnisse, die dabei erzielt wurden. Die Messung der Qualität des Algorithmus erfolgte anhand von Precision und Recall.

Bevor die Qualität des Algorithmus für semantische Gruppen evaluiert werden konnte, mußten erst die Koeffizienten der in den Gleichungen 42 und 44 angeführten linearen Funktionen passend bestimmt werden. Dazu wurden eigene Testreihen durchgeführt und nacheinander passende Werte für die Koeffizienten bestimmt. Um die ermittelten Werte zu überprüfen, wurden außerdem stark abweichende Koeffizienten getestet. Die gefundenen (und in den Gleichungen 43 und 45 dargestellten) Werte sind nicht notwendigerweise optimal, liefern jedenfalls aber gute Ergebnisse.

Abbildung 69 zeigt eine Auswahl von Beispielbildern für die semantische Gruppe der bayrischen Gemeindewappen, wie sie in der Testdatenbank vorkommen. Diese Wappen zeichnen sich im wesentlichen durch eine ähnliche Segmentierung und vor allem durch das bayrische Hoheitszeichen im oberen Feld des Schildes aus.

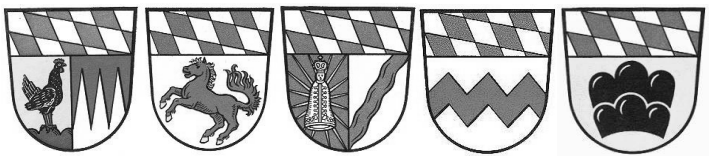

Abbildung 69: Beispielgruppe "Bayrische Wappen"

Für einen menschlichen Experten wäre es einfach, in der QBIC-Testumgebung mithilfe des Siegel-Features und der Zusatzfunktion Teilausschnitt-Suche jeweils nach dem oberen Drittel des Schildes zu suchen. Dadurch werden annähernd optimale Werte für Precision und Recall erzielt. Wenn man das Siegel-Feature z. B. noch mit dem Segmentierungsfeature kombiniert, so daß vom Siegel-Feature nur jene Bilder untersucht werden müssen, die tatsächlich über einen entsprechenden heraldischen Schnitt verfügen, läßt sich auch die Performance noch wesentlich verbessern. Algorithmisch ist es ungleich schwieriger, die passenden Bilder anhand der gegebenen Beispiele zu finden, da kein so mächtiger Erkennungsmechanismus wie der des Menschen zur Verfügung steht und folglich die leistungsfähigen Zusatzfunktionen einiger Features kaum genutzt werden können.

Der vorgestellte Algorithmus erzeugt für die semantische Gruppe der bayrischen Wappen folgende Gruppenbeschreibung:

- Ähnliche Werte für alle Symmetriefeatures: die bayrischen Gemeindewappen sind in der Mehrzahl in jeder Hinsicht unsymmetrisch. Eventuell besteht eine vertikale Symmetrie in den unteren Feldern, die aber durch die Asymmetrie des Hoheitszeichens wieder ausgeglichen wird.

- Ähnliche Segmentierung: im wesentlichen kommen nur die T-Form und eine horizontale Teilung des Schildes nach dem ersten Drittel vor.

- Ähnliche Farbverwendung: häufig verwendet werden die Tinkturen blau und rot, grün kommt praktisch nicht vor.

- Ähnliche (mittlere) Bildkomplexität.

- Große Abweichungen bestehen bei der Anzahl der verwendeten Tinkturen beziehungsweise der Farbabstufungen. Trotzdem die Farbverteilung meist ähnlich ist, variiert die Zahl der verwendeten Tinkturen in den einzelnen Bildern dennoch signifikant.

Bei weitem feiner als diese globale Gruppenbeschreibung sind die Beschreibungen für die einzelnen Teilcluster, aufgrund derer dann die Suchmodelle abgeleitet werden. Die Evaluierung dieser Suchmodelle brachte unter anderem die in Abbildung 70 dargestellten Ergebnisse. Es sind jeweils Precision und Recall in Abhängigkeit von der Anzahl der gegebenen Beispielbilder dargestellt. Der durchschnittliche Recall liegt bei 80 Prozent, während die durchschnittliche Precision ca. 60 Prozent beträgt.

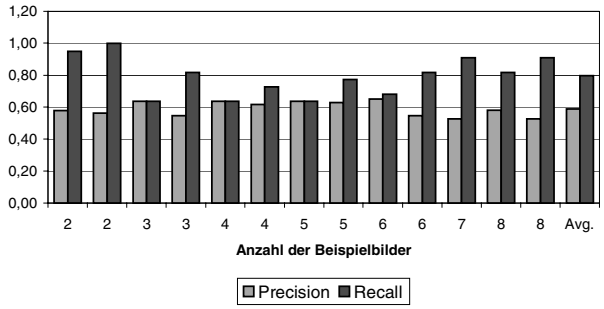

Abbildung 70: Qualität des Algorithmus zur Ableitung
von Suchmodellen aus mehreren Beispielbildern

Aus diesen Ergebnissen und den Erfahrungen während der Testphase lassen sich folgende Erkenntnisse ableiten:

1. Die Verwendung von mehr Beispielbildern verbessert nicht notwendigerweise die Qualität des Suchergebnisses. In Abbildung 70 sieht man, daß einmal sogar mit nur zwei geschickt gewählten Suchbildern ein Recall von fast 100 Prozent erreicht wird (allerdings bei einer unter dem Durchschnitt liegenden Precision), während mit sieben anderen Beispielbildern nur ein Recall von ca. 93 Prozent erreicht wird. Die Qualität der Ergebnisse ist also wesentlich von den gewählten Beispielen abhängig und eine Aussage über die Qualität des Algorithmus nur nach vielen Tests und auch dann nur schwer zu machen. Dasselbe gilt für den Endbenutzer, der letztlich auch nur dann gute Ergebnisse erzielen wird, wenn er „seine“ semantische Gruppe gut kennt und seine Beispielbilder klug wählt.

2. In den Tests wurde ein Niveau von Recall größer 80 Prozent bei einer Precision von ca. 60 Prozent erreicht. Im Vergleich zu früheren Versuchen, semantische Gruppen anhand nur eines Suchmodells zu finden, bei denen ein Recall von 51 Prozent bei einer Precision von nur 38 Prozent erzielt wurde, ist das aber eine Verbesserung des Recalls um ca. 30 Prozent und der Precision von immerhin auch 20 Prozent.

3. Das Problem der Suche nach semantischen Gruppen ist auch für künstliche Bilder nicht leicht zu lösen und es hat den Anschein, daß dort, trotzdem sich die implementierten Algorithmen sicherlich noch optimieren ließen, die derzeitigen Grenzen des CBIR liegen.

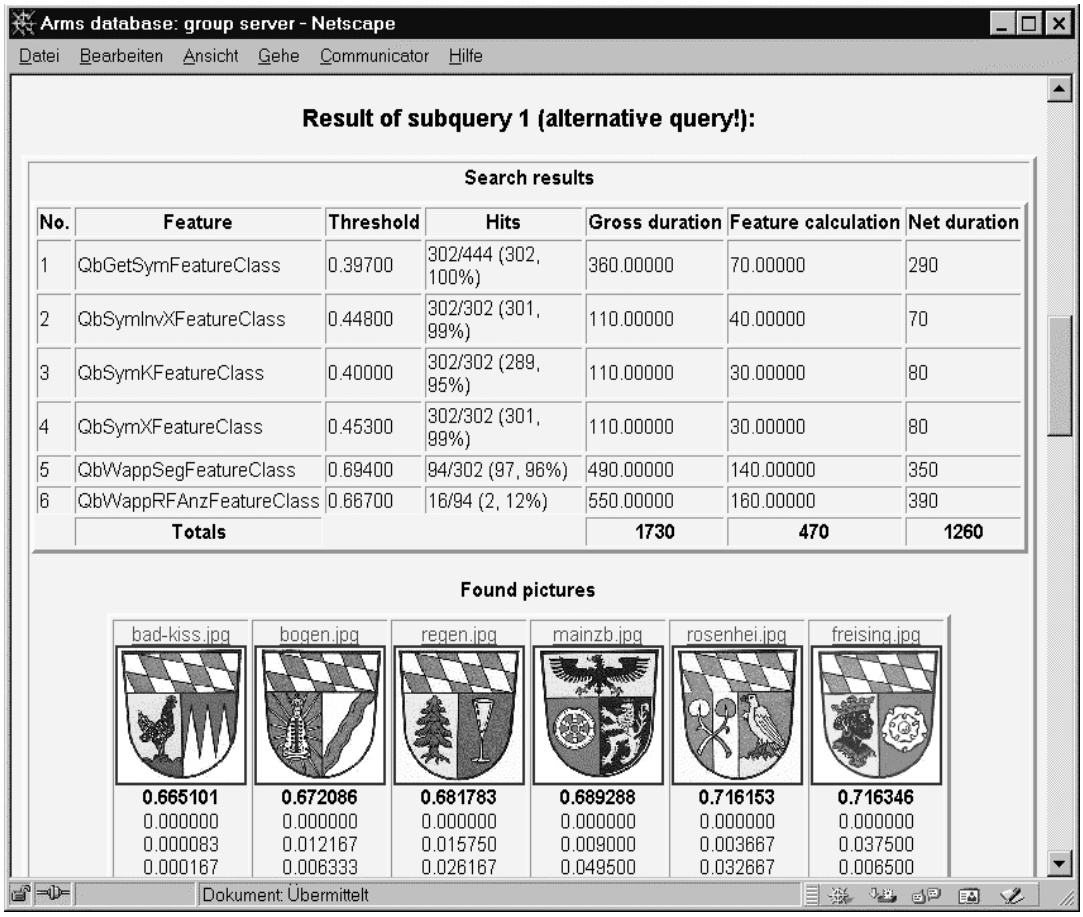

Abbildung 71: Screenshot eines Suchergebnisses

Abbildung 71 zeigt einen Ausschnitt aus einem Suchergebnis der Meta-Suchmaschine für semantische Gruppen, wobei im oberen Teil die Statistiken aller Features für den bearbeiteten Cluster und im unteren die bei dieser Teilabfrage gefunden Bilder angegeben sind. Am Beginn der Ergebnisseite befindet sich die in Abbildung 72 dargestellte globale Ergebnismenge. Hier wird zu jedem Bild der Dateiname und die Distanzsumme ausgegeben.

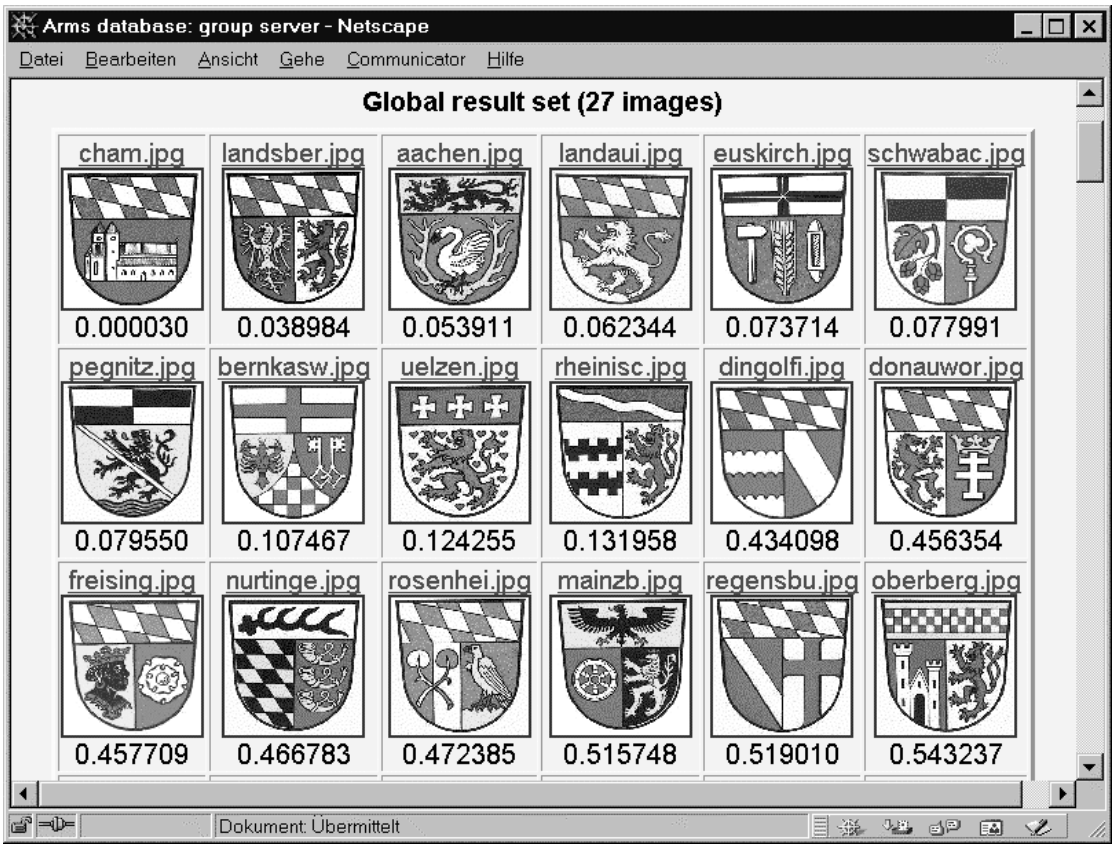

Abbildung 72: Globale Ergebnismenge einer Suche

5.2.4 Zusammenfassung

Der in diesem Abschnitt vorgestellte Algorithmus dient dazu, die Eigenschaften semantischer Gruppen zu bestimmen und daraus Suchmodelle zum Finden aller Bilder, die zur Gruppe gehören, abzuleiten. Neben den als eigene C/C++-Bibliothek implementierten Elementen des Algorithmus wurde auch auf die Gestaltung der Benutzerschnittstelle eingegangen. Die erzielten Ergebnisse liegen mit einem Recall von ca. 80 Prozent in einem akzeptablen Rahmen; eine wesentliche Verbesserung der Suchergebnisse erscheint aber nicht realistisch.

5.3 Sensitivität der Generierungsalgorithmen bezüglich der Änderungen in der Datenbasis

In diesem Abschnitt wird untersucht, wie sich die Qualität der Generierungsalgorithmen verändert, wenn zur Testdatenbank neue Bilder hinzugefügt werden. Zur Generierung von Suchmodellen werden als wesentliche Eingaben die SOM-Clusterungen der Bilder in der Datenbank und der Elemente der Featurevektoren sowie iterativ angenäherte Parameter verwendet. Durch die Prüfung der Algorithmen anhand neuer Daten wird festgestellt, wie sehr Recall und Precision von diesen Informationen abhängen. Im folgenden werden zwei Fragen für beide Generierungsalgorithmen untersucht:

- *Frage 1*: wie ändern sich Recall und Precision, wenn in der Testdatenbank n Prozent der Bilder durch neue ersetzt werden, ohne daß die Eingabedaten angepaßt werden?

- *Frage 2*: wie ändert sich die Qualität, wenn in der Testdatenbank n Prozent neue Bilder hinzugefügt werden, ohne daß die SOMs und Parameter angepaßt werden?

Im folgenden Abschnitt wird dargestellt, wie die Evaluierung durchgeführt wurde, welche Bilder verwendet wurden, welche Statistiken erzeugt wurden, etc. Abschnitt 5.3.2 zeigt, welche Ergebnisse erzielt wurden und wie diese zu interpretieren sind.

5.3.1 Vorgangsweise

Zur Evaluierung der oben angegebenen Fragen wurde eine vollautomatische Evaluierungsprozedur entwickelt, die alle Schritte von der Bildung der Testdatenbank bis zur Evaluierung der generierten Suchmodelle und Erzeugung von Statistiken durchführt. Dabei werden je Untersuchungsfrage folgende Schritte durchgeführt (vgl. Abbildung 73):

1. Generierung einer passenden Bildmenge, die n Prozent neue Bilder enthält.

2. Aufbau der Testdatenbank mit den ausgewählten Bildern. Dabei werden im wesentlichen die Featurevektoren extrahiert und die Prognosedaten für den Reihungsalgorithmus (siehe Abschnitt 4.2) erzeugt.

3. Auswahl von Suchbildern und Generierung von Suchmodellen durch die Generierungsalgorithmen. Dafür ist es notwendig, die Daten vor Beginn der Evaluierung in Gruppen einzuteilen. Die Suchbilder werden jeweils der untersuchten Bildgruppe entnommen.

4. Evaluierung der generierten Suchmodelle und Erzeugung von Statistikvektoren, die vor allen die Werte für Recall und Precision enthalten.

Wesentliche Eingaben der Evaluierungsprozedur sind die Bildmengen (bisher verwendete Bilder, neue Bilder) und ein Index, der für jedes Bild beider Gruppen angibt, zu welcher

Gruppe von ähnlichen Bildern es gehört. Für die unten angegebenen Auswertungen wurde als zweite Bildmenge neben deutschen Kreiswappen 444 deutsche Gemeindewappen herangezogen. Als Gruppe ähnlicher Bilder wurden nur die bayrischen Wappen von den übrigen unterschieden. Die Bildmenge der Gemeindewappen enthält annähernd ebenso viele bayrische Wappen wie die Menge der Kreiswappen.

Zur Beantwortung der Untersuchungsfragen wurden nacheinander Bildmengen gebildet, in denen der Anteil neuer Bilder um jeweils zehn bis zu einem Anteil von hundert (erste Frage) bzw. fünfzig Prozent (zweite Frage) ansteigt. Jede Bildmischung wurde fünfmal mit unterschiedlichen neuen Bildern gebildet. Für diese Instanzen einer Bildmischung wurden dann je Generierungsalgorithmus hundert Tests mit unterschiedlichen Suchbildern durchgeführt.

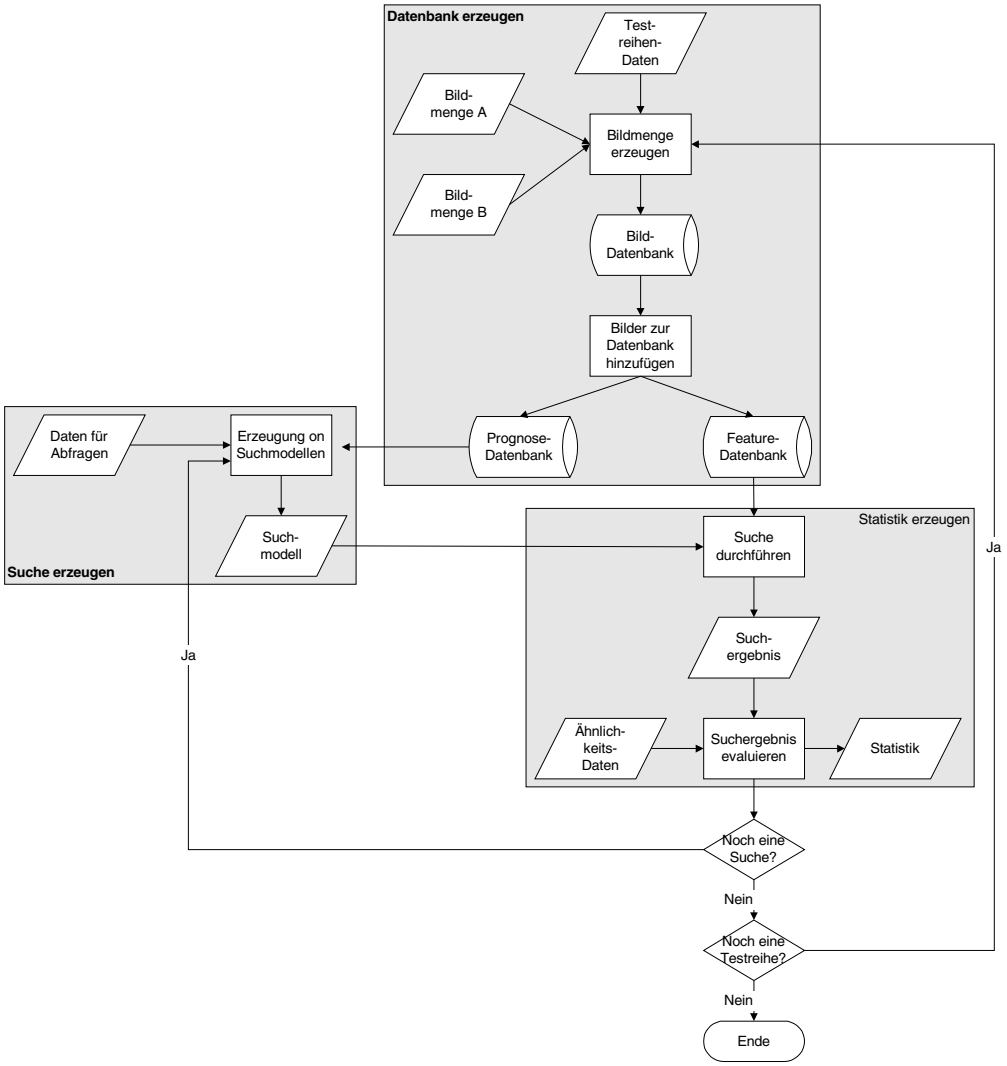

Abbildung 73: Evaluierungsprozedur für die Generierungsalgorithmen

Als Ausgabe wurde je Test ein Statistikvektor erzeugt, der folgende Daten enthält:

- Angaben zum Test: Generierungsalgorithmus, Testnummer, Bildmengenanteile, etc.

- Informationen zum Suchmodell, gewählte Suchbilder, etc.

- Ergebnisinformation: gefundene Bilder, passende Bilder, Recall, Precision, etc.

Das Ausgabeformat wurde so gewählt, daß es leicht mit Excel und Perlscripts weiterverarbeitet werden kann. Der gesamte Prozeß der Evaluierung der beiden Forschungsfragen dauerte in der Testumgebung eine Woche. Dabei wurden insgesamt 21000 Suchmodelle erzeugt und evaluiert.

5.3.2 Ergebnisse

Dieser Abschnitt beschäftigt sich mit der Analyse der Ergebnisse des Evaluierungsprozesses. Die im Vergleich zu den vorangegangenen Auswertungen geringere Qualität der Ergebnisse ist darauf zurückzuführen, daß diesmal der Suchvorgang automatisch durchgeführt wurde. Das Interesse dieser Auswertungen liegt nicht auf dem Absolutwert sondern auf der Änderung der Qualität.

Zuerst wird die Frage behandelt, wie sich die Qualität der Algorithmen bei Ersetzung von n Prozent der Bilder ändert. In Abbildung 74 ist die Änderung von Recall und Precision für den Algorithmus zur Generierung von Suchmodellen aus einem Suchbild dargestellt. Dazu ist für jeden Anteil ersetzter Bilder der Mittelwert für den Recall und die Precision angegeben. Suchen, deren Ergebnismenge leer war (Recall und Precision gleich null), sind nicht in die Mittelwertberechnung eingeflossen, da dieser Fall nur eintritt, wenn ein sehr unpassendes Suchbild gewählt wird oder der Rechner, mit dem die Suche durchgeführt wird, nicht mehr über ausreichende Ressourcen verfügt.

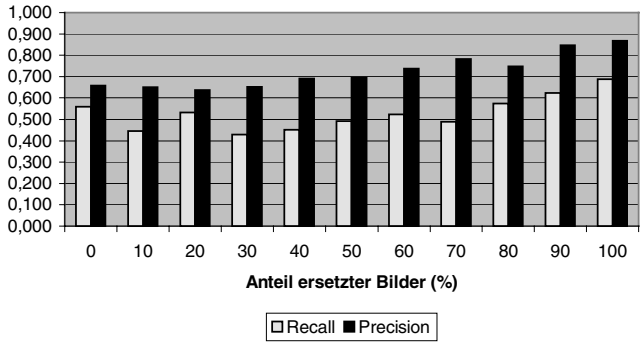

Abbildung 74: Entwicklung der Qualität des Algorithmus zur Generierung von Suchmodellen aus einem Suchbild bei konstant gehaltener Größe der Bildmenge

Es zeigt sich, daß sowohl Recall als auch Precision mit steigendem Anteil neuer Bilder trotz nicht angepaßter SOMs und nicht angepaßter Parameter nicht sinken. Eine lineare Regression ergab für beide Maße einen positiven Trend, der jedoch nicht signifikant ist (Korrelationskoeffizient R_{Recall}=0.63 und $R_{Precision}$=0.92). Das war nicht zu erwarten, zeigt aber,

daß dieser Algorithmus nicht so stark wie erwartet von den gewählten Parametern sowie den Clusterungen abhängt.

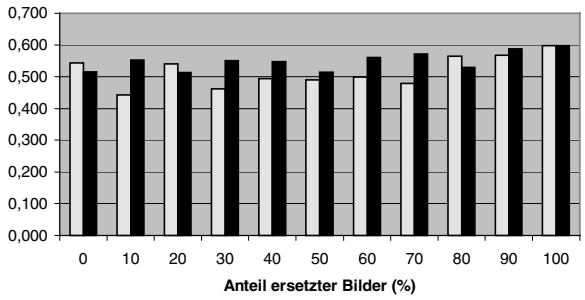

Abbildung 75: Entwicklung der Qualität des Algorithmus zur Generierung von Suchmodellen aus mehreren Suchbildern bei konstant gehaltener Größe der Bildmenge

Für den Algorithmus zur Generierung von Suchmodellen aus mehreren Beispielbildern zeigt Abbildung 75 die Änderung der Qualität. Angegeben sind wieder die Mittelwerte von Recall und Precision für alle Suchen, die zumindest ein Ergebnisbild lieferten. Es ist wiederum kein starkes Abfallen der Qualität mit steigendem Anteil ersetzter Bilder festzustellen. Eine lineare Regression ergab steigende Trends für Recall und Precision, die aber auch in diesem Fall nicht signifikant sind (R_{Recall}=0.54 und $R_{Precision}$=0.67). Insgesamt läßt sich feststellen, daß die Qualität der Generierungsalgorithmen beim Ersetzen von Teilen der Bildmenge durch neue Bilder im wesentlichen gleichbleibt.

Zur Beantwortung der Frage, wie sich die Qualität des Algorithmus zur Generierung von Suchmodellen aus einem Suchbild ändert, wenn zur bestehenden Datenbank n Prozent neue Bilder hinzugefügt werden, ohne daß die Eingabe-SOMs oder die Parameter des Algorithmus angepaßt werden, werden die in Abbildung 76 angeführten Daten verwendet. Dabei handelt es sich wieder um Mittelwerte für jeden Anteil neuer Bilder von allen Suchen, deren Ergebnismenge nicht leer ist. Es zeigt sich, daß bei schwankender Precision der Recall ungefähr gleich bleibt. Eine lineare Regression lieferte für den Recall einen leicht steigenden und für die Precision einen fallenden Trend. Diese Ergebnisse sind aber nicht signifikant (R_{Recall}=0.65 und $R_{Precision}$=0.14).

Für den Algorithmus zur Generierung von Suchmodellen aus mehreren Suchbildern ergab sich die in Abbildung 77 dargestellte Änderung der Qualität. Hier fällt die Precision deutlich ab, während der Recall gleich bleibt. Eine lineare Regression ergab wiederum für den Recall einen steigenden und für die Precision einen fallenden Trend. Diese Werte sind aber wiederum nicht signifikant (R_{Recall}=0.7 und $R_{Precision}$=0.69).

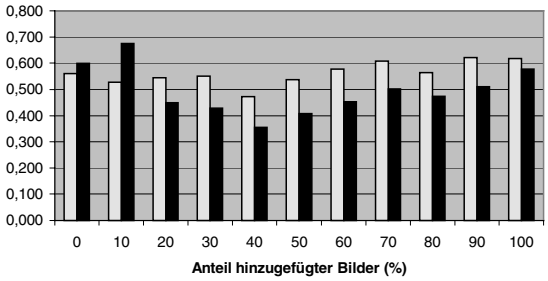

Abbildung 76: Qualität des Algorithmus zur Generierung von Suchmodellen aus einem Suchbild bei steigender Größe der Bildmenge

Die zweite Frage kann also damit beantwortet werden, daß die Precision mit steigenden Anteil neuer Bilder leicht abnimmt, während der Recall gleich bleibt. Das ist im wesentlichen darauf zurückzuführen, daß die Anzahl der Bilder in der Datenbank (444) im Prognose-algorithmus, der auch für die Thresholdberechnung verwendet wird, fest eingebaut ist und dieser Wert während der Evaluierung nicht angepaßt wurde.

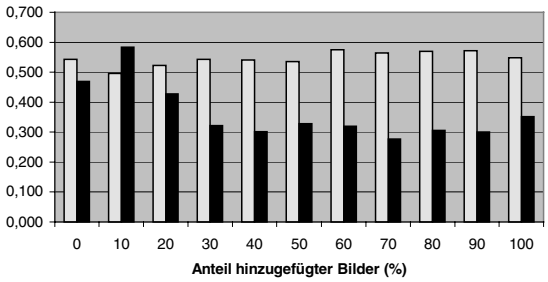

Abbildung 77: Qualität des Algorithmus zur Generierung von Suchmodellen aus mehreren Suchbildern bei steigender Größe der Bildmenge

Zusammenfassend bedeutet das, daß die Generierungsalgorithmen vor allem robust gegen große Änderungen in der Datenbasis sind.

5.3.3 Zusammenfassung

In diesem Abschnitt wurde untersucht, wie sehr die Generierungsalgorithmen abhängig vom verwendeten Datenmaterial sind. Dazu wurde eine Evaluierungsprozedur implementiert, mit der insgesamt 21000 Suchmodelle generiert und evaluiert wurden. Das Ergebnis der Auswertung zeigt, daß die Qualität der Algorithmen nicht stark vom verwendeten Daten-material abhängt.

6 Zusammenfassung

Die wesentlichen Leistungen, die im Rahmen dieser Dissertation erbracht wurden, sind:

1. Definition von Suchmodellen und Einführung eines Schwellwertes zur Bestimmung der Größe einer Ergebnismenge. Dadurch wird die Ähnlichkeitsdefinition im CBIR vereinheitlicht, eine Grundlage für die individuelle Formulierung einer für eine Suche spezifischen Ähnlichkeit geschaffen sowie durch die Schwellwerte die Qualität der Suchergebnisse wesentlich verbessert. Die Idee, viele einfache Features in Kombination zur Suche zu benutzen, stellt eine Abkehr von der bisherigen Strategie dar, mit möglichst wenigen Features Ähnlichkeit optimal zu definieren.

2. Entwicklung eines Suchsystems für Wappen. Dadurch wird gezeigt, daß es sehr wohl sinnvolle Anwendungsbereiche für reine CBIR-Systeme gibt. Für homogene Gruppen künstlicher Bilder lassen sich durch modellbasierte Ansätze sehr gute Ergebnisse erzielen.

3. Integration von Methoden zur Clusteranalyse (Self-organizing Maps) in das CBIR. Bisher wurde in CBIR-Systemen entweder auf dem Vektorraummodell oder auf Clusterung basierendes Information Retrieval angewendet. In dieser Arbeit wird versucht, die Vorteile beider Ansätze in verschiedenen Algorithmen zu integrieren.

4. Entwicklung eines Algorithmus zur automatischen Gewichtung von Features in Suchmodellen, der die Ergebnisse im Gegensatz zu früheren Systemen wesentlich verbessert und dem Benutzer die schwierige Aufgabe der Gewichtung abnimmt.

5. Entwicklung eines Algorithmus sowie der dazu nötigen Teilmodelle für Prognose, Feature-Beziehungen, etc. für die performance-optimierte Reihung von Suchmodellen. Dadurch lassen sich Suchzeiten erheblich reduzieren.

6. Entwicklung von Algorithmen zur automatischen Ableitung von Suchmodellen aus einem oder mehreren Beispielbildern. Durch mehrere Beispiele kann zudem eine subjektive Ähnlichkeit definiert werden. Die automatische Herleitung von Modellen ermöglicht die Umsetzung des benutzerfreundlichen Click & Refine – Ansatzes für CBIR-Schnittstellen.

Die entwickelten Algorithmen wurden mehrheitlich nur mit der Wappen-Testdatenbank getestet, sind aber dennoch allgemeingültig. Allerdings werden die Ergebnisse für andere Gruppen von Bildern im allgemeinen nicht an die für Wappen erreichten Ergebnisse heranreichen.

Abbildungsverzeichnis

Bibliographie

[1] Arndt, T., Petraglia, G., Sebillo, M., Tortora, G., Representing concave objects using Virtual Images, Conference on Visual Database Systems, 1995

[2] Bach, J., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., Shu, C., "The Virage image search engine: An open framework for image management", Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996

[3] Bacher, J., Clusteranalyse, Oldenbourg Verlag, München, 1996

[4] Barros, J., French, J., Martin, W., Using the triangle inequality to reduce the number of comparisons required for similarity based retrieval, SPIE Transactions, 1996

[5] Breiteneder, C., Eidenberger, H., A Retrieval System for Coats of Arms, Proc. of ISIMADE'99, 1999

[6] Breiteneder, C., Eidenberger, H., Automatic Query Generation for Content-based Image Retrieval, Proc. of IEEE Multimedia Conference, 2000

[7] Breiteneder, C., Eidenberger, H., Content-based Image Retrieval of Coats of Arms, Proc. of the 1999 International Workshop on Multimedia Signal Processing, Helsingör, 1999

[8] Breiteneder, C., Merkl, D., Eidenberger, H., Merging Image Features by Self-organizing Maps in Content-based Image Retrieval, Proc. of European Conference on Electronic Imaging and the Visual Arts, Berlin, 1999

[9] Breiteneder, C., Eidenberger, H., Performance-optimized feature ordering for Content-based Image Retrieval, X European Signal Processing Conference, Tampere, 2000

[10] DUDEN - Das Neue Lexikon, Stichworte Wappen und Wappenkunde, Mannheim, 1996

[11] Faloutsos, C., Indexing of Multimedia Data, in: Apers, P. M. G., Blanken, H. M. and Houtsma, M. A. W. (Eds.), Multimedia Databases in Perspective, Springer, Berlin, 1997, p. 219-246

[12] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P., “Query by Image and Video Content: The QBIC System”, IEEE Computer, 1995

[13] Frei, H., Meienberg, S., Schäuble, P., "The Perils of Interpreting Recall and Precision Values", in: Fuhr, N. (Eds.), Information Retrieval, Springer, Berlin, 1991, p. 1-10

[14] Fröschl, S., Faktorenanalyse, Univ. Dipl.-Arb., Technische Universität Wien, 1995

[15] Fuhr, N., Script: Information Retrieval, University of Dortmund, Dortmund, 1998

[16] Furht, B., Smoliar, S. W. and Zhang, H., Video and Image Processing in Multimedia Systems, 2nd print, Kluwer, Boston, 1996

[17] Goble, C., "Image Database Prototypes", Advances in Databases: 13th British National Conference on Databases, Springer, Berlin, 1995, p. 365-375

[18] Gong, Y., Zhang, H., Chuan, H. C., Sakauchi, M., An Image Database System with Content Capturing and Fast Image Indexing Ability, IEEE Transactions, 1994

[19] HSL and HSV: http://www.robo.mein.nagoya-u.ac.jp/~niimi/color-space/COL_23.htm

[20] Einführung in die Heraldik: http://www.sca.org.au/lochac/scribes/hrld_int.html

[21] Jacobs, C. E., Finkelstein, A., Salesin, D. H., Fast Multiresolution Querying, Proceedings of ACM SIGGRAPH, 1995

[22] Kauer, H., Vergleich klassischer Clusteranalysemethoden mit der Clusteranalyse mithilfe neuronaler Netze, Univ. Dipl.-Arb., Universität Linz, 1997

[23] Kelly, P. M., Cannon, T. M., Hush, D. R., Query by image example: the CANDID approach, SPIE Transactions on Storage and Retrieval for Image and Video Databases, 1995

[24] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., SOM-PAK: The Self-organizing Map Program Package, Helsinki, 1995

[25] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., Torkkola, K., LVQ-PAK: The Learning Vector Quantization Program Package, Helsinki, 1995

[26] Kurniawati, R., Jin, J. S., Shepherd, J. A., An efficient nearest-neighbour search while varying euclidian metrics, ACM Multimedia, 1998

[27] Laaksonen, J., Koskela, M., Oja, E., Content-Based Image Retrieval using Self-organizing Maps, International Conference on Visual Information and Information Systems, 1999

[28] Laaksonen, J., Koskela, M., Oja, E., PicSOM - A Framework for Content-Based Image Database Retrieval using Self-organizing Maps, 11th Scandinavian Conference on Image Analysis, 1999

[29] Lin, F., Picard, R. W., Periodicity, directionality, and randomness: Wold features for image modeling and retrieval, IEEE Transactions on PAMI, 1996

[30] Mehrotra, R., Gary, J., Feature-index-based similar shape retrieval, Conference on Visual Database Systems, 1995

[31] MPEG-7 - Standard: http://www.darmstadt.gmd.de/mobile/MPEG7/index.html

[32] Nastar, C., Mitschke, M., Meilhac, C., Efficient Query Refinement for Image Retrieval, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998

[33] Neudecker, O., *Großes Wappenbilder-Lexikon*, Bechtermünz Verlag, Augsburg, 1991

[34] Müller, S., Wallhoff, F., Eickeler, S., Rigoll, G., Content-based Retrieval of Digital Archives using statistical Object Modelling Techniques, Proceedings of European Conference on Electronic Imaging & the Visual Arts, Berlin, 1999

[35] Ohm, J.-R., Bunjamin, F., Liebsch, W., Makai, B., Müller, K., Smolic, A., Zier, D., A Multi-Feature Description Scheme for Image and Video Database Retrieval, IEEE Workshop on Multimedia Signal Processing, Kopenhagen, 1999

[36] Payne, J. S., Hepplewhite, L., Stonham, T. J., Evaluating content-based image retrieval techniques using perceptually based metrics, SPIE Proceedings, Vol. 3647, 1999, p. 122-133

[37] Pentland, A., Picard, R. W., Sclaroff, S., Photobook: Content-Based Manipulation of Image Databases, SPIE Storage and Retrieval Image and Video Databases II, 1994

[38] Rui, Y., Huang, T., Chang, S., Image Retrieval: Past, Present and Future, International Symposium on Multimedia Information Processing , Taiwan, 1997

[39] Rui, Y., Huang, T., Ortega, M., Mehrotra, S., Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, IEEE Transactions on Circuits and Systems for Video Technology, 1998

[40] Santini, S., Jain, R., Beyond Query By Example, ACM Multimedia, 1998

[41] Santini, S., Jain, R. Gabor Space and the Development of Preattentive Similarity, Proceedings of the International Conference on Pattern Recognition, 1996

[42] Santini, S., Jain, R., Integrated Browsing and Querying for Image Databases, to appear in: IEEE Multimedia Magazine, 1999

[43] Santini, S., Jain, R., Similarity Measures, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999

[44] Schüller, M., Selbstorganisierende Karten zur Klassifikation im Information Retrieval, Univ. Dipl.-Arb., Universität Wien, 1994

[45] Shakespeare, W., Heinrich V, London, 1592

[46] Sheikholeslami, G., Chang, W., Zhang, A., "Semantic Clustering and Querying on Heterogeneous Features for Visual Data", ACM Multimedia, 1998

[47] Smith, J. R., Chang, S., VisualSEEk: a fully automated content-based image query system, ACM Multimedia, 1996

[48] Soffer, C., Retrieval by Content in Symbolic-Image Databases, Technical Report CS-TR-3567 University of Maryland, 1995

[49] Steinböck, E., Extraktion der wesentlichen Strukturen aus einem Grauwertbild als Vorstufe für die Objekterkennung, Univ. Dipl.-Arb., Technische Universität Wien, 1988

[50] Ungerank, S., Darstellung der Faktorenanalyse in Theorie und Praxis, Univ. Dipl.-Arb., Universität Innsbruck, 1992

[51] Vass, J., Content-Based Image Retrieval by Using Color and Texture Information, http://meru.cecs.missouri.edu/mm_seminar/cont_ret.html

[52] Wang, H., Guo, F., Feng, D. D., Jin, J. S., A Signature for Content-based Image Retrieval Using a Geometrical Transform, ACM Multimedia, 1998

[53] Webers, J., Handbuch der Film- und Videotechnik, Franzis Verlag, 5. Auflage, Poing, 1998

[54] Wood, M., Campbell, N., Thomas, B., Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval, ACM Multimedia, 1998

[55] Wu, J. K., Lam, C. P., Mehtre, B. M., Gao, Y. J., Desai Narasimhalu, A., Content-Based Retrieval for Trademark Registration, Multimedia Tools and Applications, 3(3), 1996, p. 245-267