

Otto-von-Guericke-Universität Magdeburg



Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme

Diplomarbeit

Iterative Anfrageformulierung in graphischen Anfragesprachen für Multimedia-Datenbanken

Verfasser:

Anke Schneidewind

28. März 2002

Betreuer:

Prof. Dr. Gunter Saake, Universität Magdeburg, ITI

Dr. Ingo Schmitt, Universität Magdeburg, ITI

Dipl.-Inf. Thomas Herstel, Universität Magdeburg, ITI

Universität Magdeburg
Fakultät für Informatik
Postfach 4120, D-39016 Magdeburg
Germany

Schneidewind, Anke:

*Iterative Anfrageformulierung in graphischen
Anfragesprachen für Multimedia-Datenbanken*
Diplomarbeit, Otto-von-Guericke-Universität
Magdeburg, 2002.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vi
1 Einleitung	1
1.1 Motivation	3
1.2 Einsatzgebiete von Anfragesystemen für Bilddaten	5
1.3 Aufbau der Arbeit	5
2 Grundlagen	7
2.1 Multimedia-Datenbanksysteme	7
2.2 Medienobjekte	9
2.3 Multimedia Retrieval	11
2.4 Beschreibung von Bildern durch Metadaten	12
2.4.1 Attribute	13
2.4.2 Freitext	13
2.4.3 Merkmale	14
2.4.4 Inhaltsobjekte	16
2.4.5 Vergleich der Beschreibungsverfahren	16
2.5 Ähnlichkeit zwischen Bildern	17
2.6 Vergleich und Bewertung von Retrieval Systemen	19
3 Grafische Anfragen an Bild-Datenbanken	22
3.1 Retrieval	22
3.1.1 Daten-Retrieval (Data Retrieval)	23

3.1.2	Text-Retrieval (Information Retrieval)	23
3.1.3	Bild-Retrieval (Image Retrieval)	24
3.1.4	Vergleich von Daten-Retrieval mit Text- und Bild-Retrieval	26
3.2	Benutzerschnittstelle und allgemeine Informationsvisualisierung	27
3.3	Visualisierung von Anfrage und Ergebnismenge	30
4	Iterative Veränderung der Anfrage	34
4.1	Abstraktionsgrad der Anfragereformulierung	35
4.2	Ausführung der Iteration im System	37
4.3	Verändern der Wichtigkeit während der Iteration	38
4.3.1	Wichtigkeitsänderung anhand Nutzerverhalten	39
4.3.2	Dimensionswichtung	39
4.4	Auswahl der Merkmale für eine neue Anfrage	40
4.5	Browsen und Navigieren	42
4.6	Relevanz Feedback	43
4.6.1	Bewertung durch den Nutzer	44
4.6.2	Systemseitige Verarbeitung	44
4.6.3	Reformulierungsmethoden	46
4.7	Kontext-Feedback	47
5	Anfragesysteme	50
5.1	SEMCOG	50
5.1.1	Architektur	51
5.1.2	Indexierung	53
5.1.3	Benutzerschnittstelle	54
5.1.4	Iterative Anfrageformulierung	54
5.2	CHARIOT	57
5.2.1	Architektur	57
5.2.2	Indexierung	58
5.2.3	Benutzerschnittstelle	59

5.2.4	Iterative Anfrageformulierung	59
5.3	El Niño	61
5.3.1	Architektur	61
5.3.2	Indexierung	61
5.3.3	Benutzerschnittstelle	62
5.3.4	Iterative Anfrageformulierung	62
5.4	CBIR-VU	64
5.4.1	Architektur	64
5.4.2	Indexierung	64
5.4.3	Benutzerschnittstelle	65
5.4.4	iterative Anfrageformulierung	66
5.5	VIQING	67
5.6	CHABOT	68
5.7	VIBE	69
5.8	VIPER	69
6	Auswertung	70
6.1	Bewertung	70
6.2	Klassifikation	72
6.3	Diskussion	74
7	Zusammenfassung	76
	Literaturverzeichnis	78
	Thesen	84

Abbildungsverzeichnis

1.1	Beispiel zu den menschlichen Fähigkeiten zur Kantenverfolgung (Quelle [vC93])	3
1.2	Beispiel zu den menschlichen Fähigkeiten zur Mustererkennung (Quelle [vC93])	3
2.1	Euklidische Distanz dargestellt im 2-dimensionalen Raum	19
2.2	Beispiel für ein Recall-Precision-Diagramm	20
3.1	Retrieval-Prozess im Merkmal-Raum	26
3.2	Informationsfluss Mensch Maschine (Quelle: [Lan94])	28
3.3	Editor für die Anordnung und Benennung von Inhaltsobjekten	31
3.4	Darstellungsformen einer Ergebnismenge zu der Anfrage q	32
3.5	2-dimensionale Visualisierung	33
4.1	Beispiel Dimensionswichtung (Quelle [Bos00])	40
4.2	Browsen auf einer Baumstruktur mit Cluster	43
4.3	Anordnen der Bilder im Kontext-Feedback	48
4.4	Prinzip der Konzept-Box	49
5.1	System Architektur von SEMCOG (angelehnt an [LCHH97b])	52
5.2	Begriffshierarchie mit Prädikaten (aus [LCHH97b])	53
5.3	Selektivität, abgeleitet aus der Hierarchie (aus [LCHH97b])	53
5.4	Benutzerschnittstelle von IFQ (aus [LCHH97b])	55
5.5	Grafische Anfrage und simultane Übersetzung (aus [LCHH97b])	56

5.6	System Architektur von CHARIOT (aus [Bos00])	58
5.7	Ergebnisliste CHARIOT	59
5.8	Benutzerschnittstelle von El Niño (aus [SGJ01])	63
5.9	Benutzerschnittstelle von CBIR-VU (angelehnt an [JKXB01])	66
6.1	Klassifikation der iterativen Anfrageformulierung in grafischen Anfragesprachen	73

Tabellenverzeichnis

2.1	Metadaten der Bildbeschreibung und ihre Abhängigkeit	12
2.2	räumliche Beziehung (Quelle: [LCHH97b])	13
2.3	Zuordnen von Unschärfe zu Inhaltsobjekten (Quelle: [Sub98])	18
3.1	Vergleich Data Retrieval mit Information Retrieval (angelehnt an [Rij79])	27
4.1	Interne Darstellung der Nutzerbewertung [Bos00]	45

Kapitel 1

Einleitung

Die Komplexität der Informationssuche hat im Laufe ihrer Entwicklung erheblich zugenommen.

Um eine effiziente Suche zu erhalten wird in relationalen Datenbanksystemen die zu speichernde Information in Form von gezielt strukturiert abgelegten Daten gespeichert. Bei Suchsystemen für Texte, den sogenannten Text-Retrieval-Systemen, erfolgt die Suche auf nur teilstrukturierten und unstrukturierten Textdokumenten und ist dadurch aufwändiger als in Relationalen Datenbanksystemen. Text-Retrieval-Systemen arbeiten mit Suchmaschinen, die Schlüsselwörter oder Volltext als Suchanfragen zulassen. Für die Erkennung und Berücksichtigung von Synonymen werden Thesauri eingesetzt. Homonyme können dagegen nur im Zusammenhang mit dem umgebenden Text genau definiert werden. Die Ähnlichkeit von Wörtern kann durch Einsetzen von entsprechenden Lexika bestimmt werden.

In Textdokumenten kann die gleiche Information unterschiedlich ausgedrückt werden. Deshalb können die relevanten Textdokumente nicht mehr mit einfachen Beschreibungen exakt bestimmt werden, sondern werden in der Regel über ihre Ähnlichkeit zu einer Anfrage gesucht.

Parallel zum Erfolg des World Wide Web und dem damit verbundenen Text-Retrieval, ist nicht nur das Interesse an der Suche nach Texten aufgrund ihrer Inhalte gestiegen [Bim99], sondern auch an der Suche nach Bild, Video und Audio anhand ihrer Inhalte. Bei der Suche auf mehreren Medien wie Bild, Audio, Text und Video werden die speziell dafür entwickelten Multimedia-Datenbanken eingesetzt. Die einzelnen Medien werden dort als Medienobjekte abgespeichert.

Die Anforderungen an Multimedia-Datenbank-Systeme vereinen die Anforderungen von relationalen Datenbanksysteme und Text-Retrieval-Systeme und gehen noch darüber hinaus. Diese Datenbanksysteme benötigen ein Management-System, das nicht nur unterschiedliche Medienobjekte unterstützt, sondern auch Werkzeuge mit denen die Inhalte der Medienobjekte extrahiert, indiziert und verwaltet werden können. Die Systeme müssen auch neue Medienobjekte im System integrieren können und darauf medienübergreifende Anfragen ermöglichen.

Neben den Schwierigkeiten, Datenmengen in der Größe, wie sie bei Video und Audio in Multimedia-Anwendungen anfallen, zu verwalten, ist eines der Hauptprobleme die Entwicklung eines effizienten Anfragesystems. Wie im Text-Retrieval ist dabei Ähnlichkeit ein Hauptkriterium bei der Suche. Im Gegensatz zu den teilweise strukturierten Textdokumenten sind die anderen Medienobjekten in der Regel nicht strukturiert.

Die Repräsentation von Informationen in Textdokumenten ist durch das Vokabular begrenzt. Im Gegensatz dazu ist die Informationsrepräsentation in einem Bild kaum eingeschränkt. Ein Gegenstand in einem Text kann mit einer begrenzten Menge von Synonymen beschrieben und wiedergefunden werden. Schwierigkeiten bei der Suche können beispielsweise durch Homonyme entstehen. Ein Gegenstand in einem Bild kann gegenüber der Darstellung im Text beispielsweise von verschiedenen Blickwinkeln aufgenommen sein oder vereinfacht als Piktogramm dargestellt werden. Auch kann der Gegenstand in einer Vielzahl von unterschiedlichen Farben präsentiert werden ohne, dass sich die Bedeutung des Gegenstandes ändert. Damit wird die Suche in einer Menge von Bildern wesentlich schwieriger.

Bei der Betrachtung von Bildern unterliegt der Mensch mehreren Einschränkungen: optischen Täuschungen aufgrund seines Wissens und einer eingeschränkten Farberkennung und Bildauflösung aufgrund des Aufbaus des menschlichen Auges [vC93]. Während ein schwarz-weißes Zeitungsbild für den Menschen deutliche Graustufen hat, wird es unabhängig davon im Computer immer korrekt als schwarz-weiß-Bild interpretiert werden. Demgegenüber ist der Mensch fähig, unvollständige Darstellungen durch sein Wissen zu ergänzen und neben der Erkennung von Gegenständen im Bild auch Szenen und Aktionen daraus zu interpretieren.

Im Gegensatz dazu stehen die Fähigkeiten eines Rechners. Mit Algorithmen können z.B. Texturen erkannt werden die durch mehr als zwei Kriterien bestimmt werden. Der Mensch ist dazu nur begrenzt fähig. Beispiele dazu sind in Abbildung 1.1 und 1.2 zu sehen. In Abbildung 1.1 können im linken Punktemuster auf Grund der Farbunterschiede sehr leicht zwei sich kreuzende Polygone hineininterpretiert werden. Im rechten Punktemuster sind die Polygone aufgrund eines weiteren Kriteriums zur Unterscheidung nur schlecht zu erkennen. Der Betrachter muss nicht nur die Farbe der Punkte, sondern auch die Position der Farbe im Punkt erfassen um eine Interpretation vornehmen zu können [vC93].

Ähnliches gilt für die Abbildung 1.2, die noch wesentlich schwererer zu interpretieren ist. Das Feld mit den Kreuzen aus der Menge der „L“ kann vom Menschen sofort erkannt werden, der ähnlich große Bereich mit „T“ fällt hingegen kaum auf.

Durch die Wahrnehmungsfähigkeit des Menschen gibt es unterschiedliche Möglichkeiten Information zu interpretieren. Durch die unterschiedliche Verarbeitung der Information von Mensch und Computer, entsteht das Problem, dass empfundene Ähnlichkeit schlecht quantisierbar ist. Je mehr Möglichkeiten es gibt die enthaltene Information darzustellen und zu interpretieren, desto größer wird die „semantische Lücke“ zwischen Mensch und Computer.

Bei der Suche nach Bildern in einer Datenbank ist es durch das Problem der un-

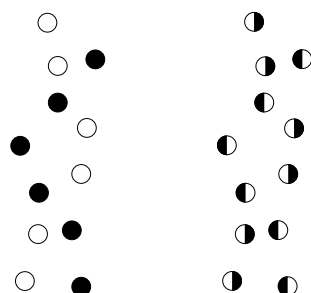


Abbildung 1.1: Beispiel zu den menschlichen Fähigkeiten zur Kantenverfolgung (Quelle [vC93])

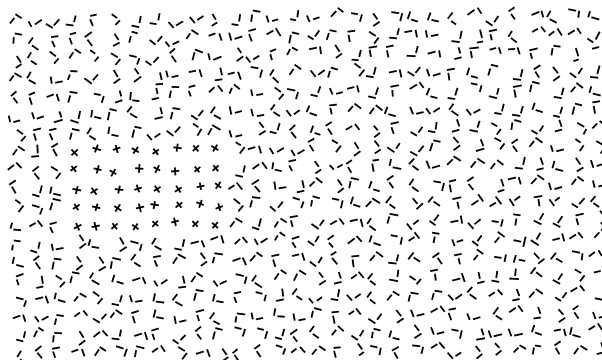


Abbildung 1.2: Beispiel zu den menschlichen Fähigkeiten zur Mustererkennung (Quelle [vC93])

terschiedlichen Interpretation sehr unwahrscheinlich, dass der Nutzer mit einer einzigen Anfrage an die Datenbank ein vollständiges und zufriedenstellendes Ergebnis erhält. Vielmehr ist zu erwarten, dass erst nach einer Folge von immer wieder neu- oder umformulierten Anfragen die semantische Lücke geschlossen werden kann. In der Vorliegenden Arbeit sollen dazu Systeme untersucht werden, welche die iterative Anfrageformulierung unterstützen.

1.1 Motivation

Multimedia-Datenbanken wurden entwickelt, um Medienobjekte wie Text, Bild, Audio und Video zu verwalten.

In relational abgelegten Daten ist die Information eindeutig und vollständig definiert und kann durch Anfragen direkt erhalten werden. Die Information aus Medienobjekten kann nur durch direkte Interpretation extrahiert werden. Der Mensch benutzt dazu seine Sinnesorgane und erlerntes Wissen. Im Rechner sind ohne Vorgaben nur bit- und pixelbezogene Informationen extrahierbar.

Für eine Informationssuche auf den Medienobjekten können umfangreiche Zusatz-

Informationen zum Medienobjekt relational gespeichert werden um damit eine zielgerichtete Suche zu ermöglichen (z.B. Autor, Erstellungsdatum, Größe usw.). Eine andere Möglichkeit ist das Extrahieren von Merkmalen aus den Medienobjekten durch das Multimedia-Datenbank-Managementsystem. Geeignetenweise werden die Merkmale so gewählt, dass sie nahe an der menschlichen Wahrnehmung liegen und berücksichtigen, wonach gesucht wird. Bezogen auf die rechnergestützte Suche, sollten Merkmale ein schnelles Vergleichen der Medienobjekte ermöglichen.

Unabhängig von der Extrahierbarkeit gibt es primitive Merkmale, die sogenannten low-level-Merkmale. Merkmale einer höheren Stufe sind high-level-Merkmale [Sch00]. Low-level-Merkmale sind beispielsweise die Farbverteilung in Bildern oder extrahierte Texturen. Mit low-level-Merkmalen können Anfragen wie folgt formuliert werden: „Gib mir alle Bilder bei denen die obere Hälfte blau und die untere gelb ist“. Mit dieser Formulierung können beispielsweise Strandbilder gesucht werden. Es kann aber auch ein Beispielbild vorgegeben werden, von dem die low-level-Merkmale extrahiert und bei der Suche mit den Merkmalen der Datenbank-Bilder verglichen werden.

Mit high-level-Merkmalen werden z.B. Gegenstände in Bildern bestimmt. Damit sind Anfragen wie „Suche alle Bilder mit einem Bus“ möglich. Die nächst höhere Stufe bei high-level-Merkmalen ist die Beschreibung von Szenen. Ein Beispiel dafür ist: „Gib mir alle Bilder mit lachenden Kindern“. Die automatische Extraktion von high-level-Merkmalen ist in der Regel nicht möglich. Deshalb beschränken sich viele Multimedia-Datenbanksysteme auf low-level-Merkmale.

Eine benutzerfreundliche Unterstützung wird durch eine visuelle Anfrageschnittstelle erreicht. Besonders die räumliche Anordnung von Bildinhalten und das Zuweisen von Merkmalen zu Regionen im Bild ist für einen Nutzer in einer visuellen Benutzerschnittstelle wesentlich leichter zu realisieren als in einer textuellen Anfragesprache wie z.B. SQL.

Bei der Suche nach Bildern bezieht sich die Vorstellung vom gewünschten Bild meistens auf das, was es darstellt und weniger auf Form oder Farbe einzelner Gegenstände. Bei der Beschreibung durch low-level-Merkmale kann das gesuchte Bild nur vage beschrieben werden. Die Ergebnismenge einer solchen Anfrage wird demnach mit großer Wahrscheinlichkeit nicht genau den Vorstellungen des Nutzers entsprechen.

Durch mehrere iterative Schritte während einem Anfrageprozess kann die menschliche Beurteilung mit in die Anfrageausführung durch das System einbezogen werden. Bei einer unbefriedigenden Ergebnismenge gibt es mehrere Verfahren, bei denen durch iterative Anfrageformulierung eine Verbesserung der Ergebnismenge erreicht werden kann. Dazu gehören z.B. das Relevanz-Feedback-Verfahren, bei dem der Nutzer die erhaltene Ergebnismenge bewertet und als neue Anfrage an das System zurück gibt. In anderen Verfahren wird die erste Anfrage des Nutzers in jedem weiteren Schritt manuell geändert.

In der vorliegenden Arbeit werden dazu Verfahren und Anfragesysteme untersucht welche die iterative Anfrageformulierung auf Bilddaten unterstützen. Im nächsten Abschnitt werden bezüglich unterschiedlicher Suchkriterien, einige Einsatzgebiete für Anfragesysteme auf Bilddaten vorgestellt.

1.2 Einsatzgebiete von Anfragesystemen für Bilddaten

Die Suche nach Bildern kann unterschiedlich motiviert sein. Es kann nach benannten Personen und Inhaltsobjekten gesucht werden, nach einer bestimmten Szene oder aber auch nach übergeordneten Eigenschaften, z.B. alle Bilder eines bestimmten Fotografen. Anfragesystemen für Bilder müssen die semantische Lücke zwischen Mensch und Computer schließen, und damit auch den unterschiedlichen Interpretationsmöglichkeiten von Bildinhalten gerecht werden. In vielen Fällen sind Datenbanksysteme mit Bildern auf ein bestimmtes Problemfeld ausgerichtet. Dadurch werden die Möglichkeiten der Suche und die vom Nutzer erwarteten Ergebnisse gegenüber einem allgemeinen System eingeschränkt.

In [Sch00] werden einige Beispiele für den Einsatz von Bild-Datenbank-Systeme aufgezählt.

Architektur- und Industriedesign: Die Visualisierung von im CAD erstellten Modellen für eine Präsentation erzeugt viele Bildobjekte, die wieder verwendet werden können und nach denen gegebenenfalls gesucht werden muss.

Journalismus und Werbung: In diesen Sparten sind Bild- und Video-Daten zentrale Informationsträger und Übermittler. Eine Suche in den oft großen Datenmengen kann sich z.B. nach einem Artikel richten, der illustriert werden muss.

Kunst: Neben der Suche nach verschollenen Kunstgegenständen (z.B. Beutekunst <http://www.lostart.de/>) sind Bild-Datenbanksysteme für die Verwaltung und Suche in Museumskatalogen zur Kunst-Recherche geeignet.

Medizin: In der Medizin können Beispielbilder für Krankheitsverläufe zu Anschauungszwecken abgespeichert werden oder gezielt in medizinischen Aufnahmen nach auffälligen Regionen wie Tumoren gesucht werden (Bilder der Kernspindeltomographie).

Sicherheit: Die Suche in Karteikarten mit Fingerabdrücken und Katalogen mit Gesichtern kann durch ein geeignetes Anfragesystem ersetzt werden.

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit richtet sich nach den einzelnen Schritten der Iteration. Das iterative Anpassen und Verfeinern einer Anfrage kann in vier rekursiven Schritten erfolgen:

1. **initiale Anfrage:** Auf einer graphischen Oberfläche werden mit unterstützenden Werkzeugen Beispielobjekte sowie Informationen über Inhalte, Merkmale und deren Beziehungen zueinander in den gesuchten Medienobjekten zu einer Anfrage zusammengestellt.

2. **Präsentation der Ergebnismenge:** Die gefundenen Medienobjekte werden präsentiert und zusätzliche Informationen zu den Ergebnisbildern und dem Datenbankinhalt angezeigt.
3. **Reformulierung der Anfrage:** Die Anfrage wird graphisch vom Nutzer oder intern anhand der Auswertung der Ergebnismenge durch den Nutzer (Relevanz-Feedback) verändert.
4. **Nächster Iterationsschritt:** Mit der neuen Anfrage wird eine neue Ergebnismenge berechnet und präsentiert.

Die Schritte werden so oft wiederholt, bis ein zufriedenstellendes Ergebnis erreicht wurde.

Die Arbeit gliedert sich wie folgt: Im Folgenden Kapitel werden zunächst die benötigten Grundlagen eingeführt. Kapitel 3 behandelt die unterschiedlichen Möglichkeiten der Anfragestellung. Außerdem werden in diesem Kapitel mögliche Visualisierungstechniken von Anfragen und Ergebnismengen betrachtet. Kapitel 4 beschäftigt sich mit der Reformulierung der Anfragen, wobei Verfahren wie Browsen, Relevanz-Feedback und Kontext-Feedback vorgestellt werden. Anfragesysteme, die einige der vorgestellten Ansätze einsetzen, werden in Kapitel 5 beschrieben. Das Kapitel 6 enthält eine Auswertung der Anfragesysteme und die Klassifikation der Ansätze. Im letzten Kapitel erfolgt die Zusammenfassung der Arbeit.

Kapitel 2

Grundlagen

Als Einführung in das Themengebiet der iterativen Anfrageformulierung werden in diesem Kapitel grundlegende Begriffe eingeführt, auf denen die Arbeit aufbaut. Entsprechend der Aufgabenstellung wird das Gebiet der Multimedia-Datenbanken (MMDB) und Anfragen kurz eingeführt.

Es wird ein kurzer Überblick über die gängigen Medienobjekte gegeben. Dabei sollen die Unterschiede der verschiedenen Techniken für Speicherung, Verwaltung und Präsentation aufgezeigt werden. Der Schwerpunkt der Beschreibungen liegt auf dem Medientyp Bild. Um die gewonnenen Erkenntnisse der Arbeit in Systeme mit anderen Medientypen (Text, Audio und Video) einsetzen zu können muss die Übertragbarkeit im einzelnen überprüft werden. In der Arbeit werden besonders die Ansätze betrachtet, die sich größtenteils unabhängig vom Medientyp einsetzen lassen.

Des Weiteren wird auf die Beschreibung der Bildinhalte durch Metadaten als Grundlage für ein Anfragesystem eingegangen. Es werden als Beschreibungsmöglichkeit für Bilder, Attribute, Freitext, Inhaltsobjekte und Merkmale eingeführt. Hierbei liegt der Schwerpunkt auf low-level-Merkmalen. Es wird ein Modell eingeführt in dem die Inhalte des Bildes in unterschiedliche Ebenen der internen Repräsentation gegliedert sind. Die Ermittlung der Ähnlichkeit von Bildern anhand der Metadaten wird in Abschnitt 2.5 behandelt.

Eine Bewertung der entwickelten Systeme ist durch Vergleich möglich. Im Bereich des Text-Retrievals gibt es dazu Plattformen, die Rahmenbedingungen hierfür schaffen.

Die im Zusammenhang mit Datenbanken benutzten Begriffe und Grundlagen orientieren sich an den Datenbankbüchern [HS95] und [HS99].

2.1 Multimedia-Datenbanksysteme

In Abhängigkeit der zu speichernden Daten und der darauf auszuführenden Operationen, haben sich unterschiedliche Datenbanksysteme entwickelt. MMDB sind auf das Bearbeiten, Präsentieren, Speichern, Suchen, Indexieren und Übertragen von Medienobjekten,

unterschiedlicher Medientypen spezialisiert (siehe 2.2). Medienobjekte sind digitalisierte Formen von Bild, Audio, Text und Video. MMDB müssen mindestens zwei unterschiedliche Medientypen unterstützen. Aufgrund der Verschiedenheit der Medientypen und der darauf auszuführenden medienabhängigen Methoden sind objektorientierte Datenbanksysteme als Grundlage für Multimedia-Datenbanksysteme (MMDBS) besonders geeignet [SST97].

Medienspezifische Daten liegen in der Datenbank in zwei Formen vor, die (unformatierten) Rohdaten und die Metadaten. Die Metadaten beschreiben die Rohdaten und werden zur Suche nach den in der Datenbank enthaltenen Medienobjekten eingesetzt.

Rohdaten von Medienobjekten wie Bild und Audio besitzen keine semantisch, direkt nutzbare, innere Struktur. Die Schwierigkeit, allgemeingültige, beschreibende Eigenschaften aus diesen Medienobjekten zu gewinnen, hat unter anderem dazu geführt, dass sich „bedarfsorientierte“ spezielle Datenbank-Managementsysteme entwickelt haben, die nur bestimmte Medientypen abdecken. Die Spezialisierung auf einzelne Anwendungsgebiete erzeugen eine genau definierte Umgebung, in der nur bestimmte Inhalte gesucht werden und dadurch die Inhalte genauer modelliert werden können. Für den steigenden Bedarf an Anwendungen, die grafische Daten verarbeiten, reichen jedoch solche spezialisierte Systeme oft nicht aus. Das bestätigt auch die Sonderstellung von allgemeinen Bild-Datenbanksystemen, die zum Teil unabhängig von MMDB entwickelt werden. Bild-Datenbanken werden beispielsweise im Bereich des World Wide Web und im Verlagswesen eingesetzt und verbreiten sich dort immer mehr [SMMP00]. Insbesondere im Web kann der Benutzerkreis keinem speziellen Gebiet mehr zugeordnet werden. Der Informationsbedarf und die Suchstrategien die zum Finden von Information eingesetzt werden, lassen keine vereinfachenden Rahmenbedingungen für ein Datenbanksystem zu.

Die Schwierigkeit, die Suche und Arbeit auf unterschiedlichen Medien-Objekten (aufgrund unterschiedlicher Ansprüche) zusammenzuführen, hat auch zur getrennten Entwicklung einzelner Medien-Datenbanksysteme geführt. Die unterschiedlich entwickelten Strategien können dabei untereinander übernommen werden.

Durch grafische Anfragesprachen werden Anfragen und Ergebnismengen visualisiert. Die Visualisierung unterbindet aber die Integration der Datenbank in Anwendungen oder Programme, wie sie bei textuellen Anfragesprachen möglich und bekannt ist. Dementsprechend wird in einigen Fällen, ähnlich der herkömmlichen Anfragesprachen, zuerst eine textuelle Anfragesprache entwickelt, auf die dann eine visuelle Darstellung der Sprache aufsetzt. Dadurch lassen sich beide Sprachdarstellungen einsetzen. Vorteil der textuellen Sprache ist auch, dass sie näher an der Verarbeitung mit dem Computer liegt.

Während die Anfragesprache SQL mit wenigen Befehlen auskommt und für relationale Datenbanksysteme (RDBS) ausreicht, benötigen MMDBS entsprechend der Medientypen erweiterte Anfragesprachen, die komplexer und dadurch schwieriger zu erlernen sind. In der Regel sind textuelle Anfragesprachen für MMDBS nicht unabhängig für alle Medienobjekte definiert [Pas99].

Bei MMDB und Datenbanken allgemein kann ein hoher Aufwand betrieben werden, um Daten für eine effiziente Suche abzuspeichern. Da die manuelle Eingabe von Daten

sehr aufwändig ist und eine Bildbeschreibung mitunter mehrere Seiten umfassen kann, ist das Ziel, ein optimales Aufwand/Leistungsverhältnis zu erhalten. Mit einem möglichst geringen manuellen Aufwand der Dateneingabe, soll eine möglichst einfache und effiziente Suche erreicht werden.

2.2 Medienobjekte

Die vom Rechner abgespeicherten Daten entsprechen in vielen Fällen nicht den Informationen, die ein Anwender in den einzelnen Medien sucht. Werden Bilder in der Datenbank durch Farb-Merkmale beschrieben, kann z.B. nicht direkt mit dem Begriff „Auto“ gesucht werden. Das unterscheidet unter anderem MMDB von relationalen Datenbanken (RDB) und ist einer der Gründe, warum keine direkte Abfrage im MMDBS nach der benötigten Information möglich ist (kein exact match). Für die einzelnen Medienobjekte gibt es in der Regel mehrer (komprimierte) Repräsentationsformate (die systeminterne Abbildung des Medientyps). Gängige Formate für Bilder sind JPEG, GIF und das unkomprimierte Format BMP.

Im folgenden werden die Medientypen Text, Audio, Bild und Video (Animation) vorgestellt. Textdaten werden in MMDB meist als strukturierte Textdokumente abgelegt. Anhand der Struktur kann in vielen Fällen noch Information extrahiert werden. Wörter in Titeln z.B. sind in der Regel wichtiger und können beim Retrieval höher bewertet werden [Guo99]. Audiodaten werden nach der Umwandlung durch Sampling (abtasten, Zeitintervalle), Quantisierung (Wertintervalle) und Codieren (den Frequenzen binäre Werte zuweisen) als Audio-Objekte abgelegt.

Eine besondere Art von Bildern ist die Vektorgraphik. Vektorgraphiken werden durch Parameter für mathematische Formeln oder durch vordefinierte Modelle beschrieben. Sie sind leicht veränderbar und benötigen nur wenig Speicherplatz, da nur die Parameter und Modelle zur Herstellung der Graphik gespeichert werden. Eine Suche auf diesem Bildtyp unterscheidet sich stark von der Suche auf Pixelbildern [Guo99]. Daher sollen Vektorgraphiken in der vorliegenden Arbeit nicht weiter betrachtet werden.

Im Gegensatz zur Vektorgraphik steht das Rasterbild, als eine Menge an Bildpunkten (Pixel) ohne vorgegebene Struktur. Die Digitalisierung erfolgt ähnlich wie bei Audiodaten. Im Rohzustand, in Form eines Bitmap, ist die Repräsentation aller Rasterbilder gleichartig, unabhängig von der Originalquelle. Die Pixel sind in Linien und Spalten angeordnet. Regionen im Bild können durch eine Menge von x,y-Koordinaten für die enthaltenen Pixel bestimmt werden. Dichte und Anzahl der Pixel definieren die Größe des Bildes.

Die Farben die zur Darstellung für der einzelnen Pixel eingesetzt werden, sind durch ein Farbsystem festgelegt. Mit Primärfarben als Achsen eines Koordinatensystems sind Farbräume die Grundlage für die Extraktion der Farbverteilung aus Bildern. Ein gängiger Farbraum ist das RGB-Farbmodell mit Rot Grün und Blau als Primärfarben und Achsen des Systems [Sch00].

Eine typische Speichergröße für Farbwerte pro Pixel sind 24 Bit, wobei die Primärfarben mit jeweils 8 Bit dargestellt werden. Diese Werte bilden die unteren Grenzen der menschlichen Wahrnehmung von Farbunterschieden und sind anhand von Tests zustande gekommen [Guo99]. Damit wird sichergestellt, dass digital gespeicherten Bilder „farbecht“ erscheinen und auch bei der Präsentation auf der Nutzeroberfläche genauso aussehen wie das Originalbild. Zur Präsentation der Bilder während einer Suche, werden in der Regel wesentlich kleinere Beispielbilder eingesetzt. Dazu werden von den gespeicherten Bildern zusätzlich Miniaturbilder abgelegt, die sogenannten Thumbnails.

Das Video als letztes der vorgestellten Medientypen ist eine Folge von sogenannten Frames in zeitlicher Abhängigkeit. Frames sind entweder Rasterbilder oder Vektorgrafiken. Die Indexierung und das Retrieval von Animationen bzw. Video sind dem des Bild-Retrievals ähnlich. Der Unterschied ist, dass eine Zeitachse zur Beschreibung der Daten zusätzlich berücksichtigt werden muss.

Die vorliegende Arbeit bezieht sich auf die Suche in Bilddaten. Eine Verwirklichung von medientypübergreifenden Ansätzen für Retrieval und Indexierung ist von den Überschneidungspunkten der Repräsentation und Präsentation einzelner Medientypen abhängig. Bei der Präsentation unterscheiden sich Bild und Video nur durch die Zeitkomponente. Der Einsatz von Thumbnails für die Präsentation, wie sie bei Bild üblich ist, ist auch für Video geeignet. Mit Hilfe von Thumbnails können wesentlich mehr Bilder oder Frames gleichzeitig angezeigt werden, als es mit Originalen möglich wäre. Mit der gleichzeitigen Präsentation von mehreren Bildern kann z.B. der Inhalt einer Ergebnismenge wesentlich leichter erfasst werden als bei einer Einzelbetrachtung der Bilder. Der Informationsverlust durch die Verkleinerung ist in der Regel unbedeutend für die Beurteilung der Relevanz einzelner Bilder. Für Text hingegen sind Thumbnails ausgeschlossen. So ist auch die gleichzeitige oder simultane Präsentation von Text oder Audio nicht sinnvoll, da sie in dieser Form nicht vom Nutzer ausgewertet werden können.

Eine bei allen Medienobjekten übliche Repräsentations-Methode ist es, den Informationsinhalt durch extrahierbare Merkmale zu beschreiben. Die durch Algorithmen extrahierten Werte liegen als Vektor vor. Eine hohe Anzahl von Werten bedingt einen hochdimensionalen Raum, in dem die Medienobjekte als Punkte abgebildet werden. Auf den Merkmalen kann eine inhaltsbasierte Suche erfolgen: Das sogenannte Content-Based-Retrieval. Die inhaltsbasierte Suche ist in der Regel eine Ähnlichkeitssuche, in der eine Anfrage den gewünschten Inhalt beschreibt. Die Anfrage wird in Form eines Vektors mit den Vektoren der Medienobjekte der Datenbank verglichen und so die ähnlichsten Medienobjekte ermittelt.

Der Zugriff auf die Bilder der Datenbank erfolgt anhand ihrer extrahierten Merkmale. Somit sind Merkmale die Grundlage der Indexierung [Guo99]. Merkmale müssen demnach so gewählt werden, dass sie neben der wahrnehmungsorientierten Repräsentation der Inhalte, auch eine effiziente Suche ermöglichen. Der Zugriff auf Merkmale wird in der Regel durch geeignete Zugriffsstrukturen unterstützt.

Medienobjekte unterliegen oft Urheberrechten, was bei der Erstellung einer MMDB und bei der Nutzung der angebotenen Anfragemöglichkeiten auf die Objekte nicht ver-

nachlässigt werden darf. In [FS95] wird ausführlich auf diese Problematik eingegangen.

2.3 Multimedia Retrieval

Für die Suche in MMDB wird ein Suchverfahren benötigt, das an unterschiedliche Repräsentationsformen der Medienobjekte, an hochdimensionale Zugriffsstrukturen und Indexierungsverfahren angepasst ist. Dabei müssen die Anforderungen an eine Datenbank-Anfragesprache berücksichtigt werden. In [Pas99] werden dazu 15 allgemeine Anforderungen an eine Datenbank-Anfragesprache vorgegeben und 6 Anforderungen die sich speziell auf Multimedia-Anfragesprachen beziehen.

Anforderungen sind z.B. die Ad-hoc-Formulierung oder auch die Deskriptivität der Anfragesprache, sie fordern Anfragesprachen in denen nicht den Suchvorgang beschrieben werden muss sondern nur die zu suchenden Daten. Andere Anforderungen sind Effizienz, Erweiterbarkeit und Berechnungsvollständigkeit. Zusätzliche Anforderungen an Multimedia-Anfragesprachen sind unter anderem inhaltliche, zeitliche und räumliche Anfragen so wie die Möglichkeit unscharfe Anfragen zu stellen.

Bei der Suche auf den Medienobjekten ist in vielen Fällen nur eine vage Vorstellung über den dargestellten Inhalt vorhanden. Dabei sind in der Regel die Bedingungen die an den Inhalt gestellt werden unterschiedlich wichtig für den Nutzer (z.B. ist es dem Nutzer wichtig, dass ein UFO in dem Bild vorkommt, aber es ist ihm nicht so wichtig, dass es grau ist). Mit dem Einsatz von unscharfen Anfragen können z.B. durch unscharfe Prädikate, Relevanzwerte und gewichtete Anfrageterme vage Vorstellungen umschrieben und gewichtet werden.

Für die Entwicklung von Retrievalsystemen werden in [Sch97] zwei Verfahren angegeben, die im Multimedia-Retrieval kombiniert werden.

Das erste Verfahren bedingt speicherorientiertes Retrieval. Nur ein Abspeichern von Daten an einem bestimmten Ort ermöglicht das Wiederfinden. Entsprechend dieses Prinzips wird der Schwerpunkt der Entwicklung auf das Finden und Organisieren solcher Speicherorte gelegt, d.h. wo wird Dokument gespeichert und wo wird Referenz gespeichert. Mit diesem Ansatz werden Informationsstrukturen erzeugt, auf denen gesucht werden kann.

Das zweite Verfahren ist die probabilistische Sortierung. Die Antwort auf eine Anfrage ist ein Sortieren der Medienobjekte einer Sammlung. Die Sortierung erfolgt nach der abgeschätzten Nützlichkeit des Medienobjektes für den Nutzer anhand der Anfrage. Die Abschätzung erfolgt auf der für das System zugreifbar gemachten Information.

Bezogen auf die Darstellung der Inhalte von Medienobjekten durch extrahierte Merkmale existiert ein Multimedia-Retrievalmodell in Form eines Vektorraum-Modells. Es besteht aus einer Indexierungsmethode und einer Retrievalfunktion. Die Indexierungsmethode generiert Beschreibungsvektoren und die Retrievalfunktion generiert einen Wert anhand der Anfrage, der die abgeschätzte Nützlichkeit eines Medienobjektes ausdrückt. Der erhaltene Wert ist ein Maß für die Ähnlichkeit zwischen dem Anfragevektor und

dem Medienobjektvektor der Datenbank.

Neben der Beschreibung von Inhalten durch Merkmale gibt es bei der Suche in Bilddaten weitere Beschreibungsmöglichkeiten durch andere Metadaten. Die anderen Arten der Beschreibung unterstützen spezielle Verfahren der Suche in Bilddaten. Dazu werden im nächsten Abschnitt neben den Merkmalen weitere Bild-Beschreibungsverfahren vorgestellt.

2.4 Beschreibung von Bildern durch Metadaten

Für eine rechnerunterstützte Suche auf Bildern müssen diese durch Metadaten so repräsentiert sein, dass sie mit einer Nutzeranfrage verglichen werden können. Eine ideale, allgemeingültige Beschreibung eines Bildes erlaubt die Rekonstruktion des gesamten Bildes und das Ausdrücken jedes denkbaren Ähnlichkeitskriteriums [SJ00]. Dabei sollte die Repräsentation eine reduzierte und redundanzfreie Darstellung der wesentlichen, enthaltenen Information [WI98] sein. Gründe für eine reduzierte Darstellung ist, eine schnelle und effiziente Suche bei möglichst kleinem Speicherverbrauch. Bildinhalte sind nach [Bim99] wahrnehmbare Eigenschaften wie Farbe, Textur, Kontur und räumliche Beziehungen (sogenannte inhaltsabhängige Metadaten), semantische Primitive wie Gegenstände, Funktionen und Szenen, aber auch Eindrücke, Emotionen und Bedeutung (sogenannte inhaltsbeschreibende Metadaten). Inhaltsunabhängige Metadaten sind Urheber, Bildformat, Bildabmessungen, Aufnahme datum und andere direkt ermittelbare Einstellungen. Eine Zusammenfassung der Metadaten ist in Tabelle 2.1 aufgelistet.

Abhängigkeit	Metadaten
inhaltsabhängig	Farbe, Textur, Kontur und räumliche Beziehungen
inhaltsbeschreibend	Gegenstände, Funktionen, Szenen, Eindrücke, Emotionen und Bedeutung
inhaltsunabhängig	Urheber, Bildformat, Bildabmessungen, Aufnahme datum, Photograph

Tabelle 2.1: Metadaten der Bildbeschreibung und ihre Abhängigkeit

Übliche Repräsentationen von Bildbeschreibungen sind nach [Guo99]:

- Attribute,
- Freitext,
- Merkmale und
- Inhaltsobjekte.

Inhaltsobjekte sind Gegenstände im Bild dargestellt sind und als Einheit in der Datenbank verwaltet werden.

2.4.1 Attribute

Attribute sind (in der Regel relational) abgelegte Daten zu Bildern und deren Inhalt. Die relationale Beschreibung von Bildern umfasst normalerweise alle inhaltsunabhängigen Attribute. Inhaltsbeschreibende Attribute können Ort der Aufnahme oder räumliche Beziehungen zwischen dargestellten Inhaltsobjekten sein. Ein Beispiel für das Ablegen von räumlichen Beziehungen als Attribute ist in Tabelle 2.2 zu sehen. Dabei wird zu je zwei Inhaltsobjekten ein räumliches Beziehungs-Attribut zugewiesen.

Bild ID	Objekt 1	räuml. Beziehung	Objekt 2
001	Mann	links von	Auto
002	Baum	rechts von	Auto
003	Mann	links von	Flugzeug
004	Mann	links von	Fahrrad
005	Mann	rechts von	Flugzeug
...

Tabelle 2.2: räumliche Beziehung (Quelle: [LCHH97b])

Es ist aber auch möglich, inhaltsabhängige Attribute zu speichern. So kann z.B. die Farbverteilung im Bild anhand eines vorgegebenen Vokabulars beschrieben werden. Die Begriffe des Vokabulars müssen dabei den Farbton und Vorkommen abbilden, z.B. „viel Rot“. Ein Vorteil des Einsatzes von Attributen ist, dass effiziente RDBS genutzt werden können. Nachteilig ist hingegen die Unvollständigkeit und die auf die definierten Attribute begrenzte Suche. Bilder die zur Beschreibung andere Attribute als die vorgegebenen brauchen, können nicht gesucht werden.

2.4.2 Freitext

Freitext ist die Beschreibung einzelner Bilder oder auch Bildgruppen in Form von Wörtern, in zusammenhängenden Sätzen oder einer Stichwortsammlung ([SJ00]). Die Verwaltung und Bearbeitung von Freitext und die Suche in Freitexten erfolgen mit den Methoden des IR. Freitext eignet sich besonders für das Beschreiben von high-level Bildinhalten, z.B. Szenenbeschreibungen oder auch subjektive Sinneseindrücke wie Emotionen.

Eine aus dem Text-Retrieval bekannte Indexierung von Texten erfolgt in drei Schritten. Im ersten Schritt werden Wörter (Terme) aus den Texten entfernt die geringe oder keine direkte Aussagekraft haben (z.B. Artikel, Konjunktionen usw.) sogenannte Stoppwörter.

Im zweiten Schritt erfolgt das Reduzieren der übrigen Wörter auf die Grundform (was im Englischen wesentlich leichter ist da meist nur eine Endung entfernt werden muss), das sogenannte Stemming.

Der dritte Schritt ist das Zählen der Häufigkeit der einzelnen Wörter. Aus der Häufigkeit der Wörter wird ein gewichteter Term-Vektor erzeugt. Jeder Term entspricht einer Dimension im Vektorraum. Die Häufigkeit des Terms bestimmt die Position in der dazugehörigen Dimension. Die Anzahl der eingesetzten, unterschiedlichen Wörter in einer Datenbank bestimmt die Anzahl der Dimensionen, bzw. die Länge des Vektors.

Eine typische Anzahl an Dimensionen im Vektor-Raum ist 10^4 . Einzelne Dokumente benutzen jedoch nur eine kleine Untermenge des gesamten möglichen Vokabulars. Für sie reichen in der Regel 10^2 -Dimensionen. Die Häufigkeitswerte für die restlichen Dimensionen sind Null, da der entsprechende Term nicht vorhanden ist [SMMP00].

Texte zu Bildern könne je nach Herkunft auch automatisch extrahiert werden. Wird das Bild z.B. einer WWW-Seite entnommen, kann zugehöriger Text wie beispielsweise der Alternativ-Text oder Bild-Titel als Annotation abgelegt werden (siehe [Bim99]). Eine andere Möglichkeit besteht darin, direkt nach Text im Bild zu suchen. In [LE00] wird die Texterkennung in Bildern behandelt. Dazu sind z.B. besonders Bilder aus einem Filmabschnitt geeignet, da sie sehr viel textuelle Informationen bezüglich dem vorliegenden Film enthalten.

2.4.3 Merkmale

Merkmale, auch Features genannt, sind automatisch extrahierbare Bildeigenschaften die in Form von Vektoren für jedes Bild abgelegt werden. Im Folgenden werden die automatisch extrahierbaren low-level-Merkmale betrachtet.

Entsprechende der Wahrnehmung kann beispielsweise zwischen den Merkmalen Textur, Farbe und Kontur unterschieden werden. Für jedes dieser Merkmale stehen meistens mehrere Extraktionsverfahren zur Verfügung. Jedes Extraktionsverfahren erzeugt eine interne Repräsentation des Merkmals, eine sogenannte *Merkmalsrepräsentation*. Zu den häufig genutzten Extraktionsverfahren für Textur gehören unter anderem der Gaborfilter und Co-ocurrence Matrix ([RHM98]). Farben können durch Farbhistogramme und Farbmomente repräsentiert werden. Ein Beispiel für die Beschreibung von Konturen ist der *Smallest Enclosing Circle*.

In [RHM98] wird die Modellierung von Bildinhalten durch Merkmale formalisiert. Ein Bildobjekt I wird dargestellt durch:

$$I = I(D, F, R) \tag{2.1}$$

- D sind Rohdaten des Bildes
- $F = \{f_i\}$ ist eine Menge von Merkmalen (z.B. $F = \{Farbe, Textur, Kontur\}$)

- $R = \{r_{ij}\}$ ist eine Menge von Merkmalsrepräsentationen, in Abhängigkeit der Merkmale in F (z.B. $R = \{Farbhistogramm, Farbmomente, co - occurrenceMatrix, Gabor - Filter, SmalestEnclosingCircle\}$)

Jede Merkmalsrepräsentation (MR) kann durch einen Vektor dargestellt werden. Die Werte des Vektors sind die einzelnen Merkmalsausprägungen (MA). Der Vektor ist beschrieben durch $r_{ij} = [r_{ij1}, \dots, r_{ijk}, \dots, r_{ijK}]$ in dem K die Länge des Vektors der Merkmalsrepräsentation, bzw. die Anzahl der Merkmalsausprägungen ist. Jede Merkmalsausprägung stellt eine Dimension in einem Vektorraum der Merkmalsrepräsentationen dar. Wird F als Vektor abgelegt, entsteht durch Ersetzen der Merkmale mit den Merkmalsrepräsentationen und durch das Ersetzen der Merkmalsrepräsentationen mit den Merkmalsausprägungen ein Vektor, der alle Merkmale eines Bildes durch seine Werte repräsentiert. Die Werte des Vektors können je nach Extraktionsmethode stark voneinander variieren.

Durch die Definition von Merkmal, Merkmalsrepräsentation und Merkmalsausprägung, liegen hinter jeder Bildbeschreibung drei Ebenen: Die erste Ebene enthält die Merkmale, die zweite Ebene die Merkmalsrepräsentationen und die Dritte die einzelnen Merkmalsausprägungen.

Durch Einteilen eines Bildes in vordefinierte Bereiche (z.B. vier Quadranten: Nord, Süd, West, Ost), können Merkmale auch lokal extrahiert werden. Damit können Inhaltsobjekte anhand ausgeprägter Merkmale grob lokalisiert und unterschieden werden. Zusätzlich können ihnen dadurch räumliche Beziehungen zugeordnet werden.

Entscheidend für die Suche mit Merkmalen ist, eine Merkmal-Kombination zu finden, die der menschlichen Wahrnehmung nahe kommt. Die Merkmale müssen die Bilder einer Datenbank voneinander unterscheidbar macht und auf der Ähnlichkeit berechnet werden kann. Die Extraktion eines Farbhistogramms als Merkmal für Schwarz-Weiß-Bilder ist beispielsweise ungeeignet.

Bei einer entsprechenden Auswahl von Merkmalen können Merkmals-Vektoren in einer Größenordnung von 10^3 entstehen. Besonders viele Merkmalsausprägungen sind z.B. im Anfragesystem Viper [SMMP00] verwirklicht: 85 000 einzelne Merkmalsausprägungen. Im Gegensatz zu den Vektoren im Text-Retrieval sind in der Regel die meisten Werte der Vektoren der Bilder ungleich Null [SMMP00].

Merkmale, die schnell extrahiert werden können ermöglichen ein Einbinden von digitalen Bildern als Beispielbilder in eine Anfrage. Durch die schnelle Extraktion können während der Laufzeit die nötigen Merkmale zur Suche aus dem Beispielbild gewonnen werden.

Ein Nachteil von Merkmalen ist, dass viele Merkmalsrepräsentationen nicht mehr direkt interpretierbar sind. Während sich ein Farb-Histogramm [RT01] relativ leicht visualisiert und interpretiert lässt, ist der Mensch kaum mehr in der Lage aus einer Co-Occurrence Matrix Bezüge zum Bild herzustellen. Zudem ist der Mensch auf Grund seiner eingeschränkten Fähigkeiten nicht fähig einzelne Farben zu unterscheiden und einzelne Farbtöne eines gesuchten Bildes genau zu beschreiben. Unscharfe Anfragen werden

damit bei der Suche auf Merkmalen zum allgemeinen Fall.

Die Aufschlüsselung eines Bildes in einzelne Merkmale eignet sich nicht nur zum Vergleich von Bildern, sondern auch für eine flexible Anfrageformulierung. Merkmale können wie Attribute in Anfragen eingesetzt werden, um so einzelne Merkmale von Bildern hervorzuheben bzw. auszuschließen [Bos00].

2.4.4 Inhaltsobjekte

Inhaltsobjekte können manuell oder semiautomatisch bestimmt werden. Der Zugriff kann über eine Ontologie erfolgen oder aber auch durch relational abgelegte Bezeichnungen die sich auf ein Inhaltsobjekt beziehen.

Eine semiautomatische Objekterkennung ist durch Segmentierung und Suchen von Regionen möglich (z.B. mit *Split and Merge* oder *Region Growing*). Aus den als Inhaltsobjekt erkannten Regionen können zum Wiederfinden und Vergleichen mit anderen Inhaltsobjekten Merkmale extrahiert werden. Die Merkmale sind

Die Beschreibung von Bildinhalten mit Hilfe von Ontologien in Form von Objekthierarchien, kommt der Objektorientierung vieler Bild-Datenbanken entgegen. Innerhalb der Ontologie ist die Ähnlichkeit durch Vater-Sohn-Beziehungen bestimmt, die gegebenenfalls auch bezüglich ihrer Ähnlichkeit bewertet sein können.

Zwischen Inhaltsobjekten im Bild bestehen räumliche Beziehungen die relational abgelegt werden können.

2.4.5 Vergleich der Beschreibungsverfahren

Es gibt mehrere Nachteile bei der Annotation durch Freitext und Attribute. Das Einfügen von Annotation oder Attributen ist wesentlich aufwändiger, als die automatische Merkmalsextraktion, da sie vom Nutzer manuell erfolgen muss. Damit ist das Verfahren auf Systeme begrenzt, die mit einer „vertretbaren“ Anzahl an Medienobjekten auskommen. Mit Freitext und Attributen sind weiterhin nicht alle Bedeutungs- und Interpretationsmöglichkeiten eines Bildes erfassbar. Dies gilt besonders bei emotionalen Eindrücken, die das Bild erzeugt, da diese abhängig vom Betrachter und auch dem Bearbeiter bei der Annotation sind [SJ00]. Außerdem sind Freitext und Attribute nicht besonders geeignet, Ähnlichkeit zu modellieren. Es ist beispielsweise schwer, mit Worten die Ähnlichkeit von Personen zu beschreiben.

Automatische Objekterkennung in Bildern ist im allgemeinen nicht möglich und ist, wie Attribute und Texte, auf die Angaben beschränkt, die manuell oder semiautomatisch extrahiert bzw. beschrieben wurden. Die Repräsentation von Merkmalen und Text durch Vektoren ermöglicht die Anwendung der linearen Algebra.

Merkmale sind eine gängige Form der Bildbeschreibung und die Suche auf ihnen wird in vielen Systemen unterstützt. Merkmale können nicht direkt Emotionen ausdrücken. Indirekt erfolgt dies jedoch durch bewerten der Merkmale. Es gibt beispielsweise eine

Skala, die eine Einteilung der Farben in die Bereiche warm, kalt und neutral vornimmt und somit einen Gesamteindruck des Bildes ausdrücken kann [Bim99].

In der Praxis werden oft mehrere Bildbeschreibungsmechanismen miteinander kombiniert um Vor- und Nachteile der Beschreibungen gegeneinander auszubalancieren. Ein dabei zu bewältigendes Problem ist die Abbildung der Anfrage auf die Repräsentation der Inhaltsbeschreibungen. Eine Anfrage muss demnach erst in die Datentypen Merkmal, Attribut usw. zerlegt werden. Die gewonnenen Teilanfragen können dann mit den entsprechenden Metadaten der einzelnen Bilder aus der Datenbank verglichen werden. Das Anfrage-Ergebnis ist dann eine Verschmelzung der Teil-Ergebnisse von den einzelnen Bildbeschreibungen.

2.5 Ähnlichkeit zwischen Bildern

Bei unscharfen Anfragen an eine MMDB muss die Ähnlichkeit bestimmt sein. Um anhand einer Bildbeschreibung ähnliche Bilder zu finden, wird eine Methode definiert, die zwischen zwei Bildbeschreibungen eine Ähnlichkeit berechnen kann. Entsprechend der unterschiedlichen Bildbeschreibungen durch Attribute, Freitext, Merkmale und Inhaltsobjekte, gibt es unterschiedliche Methoden für die Ähnlichkeitsberechnung.

Bei der Suche in Bildern die durch definierte Attribute beschrieben sind, kann das Ermitteln der Ähnlichkeit zwischen den Bildern durch das Einführen einer Unschärfe ermöglicht werden. Im Vektorraum der Merkmale und des Freitextes ist die Ähnlichkeit durch eine Distanz zwischen den Vektoren der Bilder festgelegt. Eine geringe Distanz zwischen zwei Vektoren entspricht einer hohen Ähnlichkeit der zugehörigen Bilder. Die Ähnlichkeit zwischen Bildinhalten, die durch Inhaltsobjekte beschrieben sind, ist eine Kombination aus räumlicher und inhaltsobjektbezogener Ähnlichkeit. Zum einen können Bilder mit Inhaltsobjekten ähnlich bezüglich der räumlichen Anordnung ihrer Inhaltsobjekte sein. Zum anderen besteht eine Ähnlichkeit zwischen den unterschiedlichen Inhaltsobjekten bezüglich ihrer Anordnung in einer Ontologie und es besteht eine Ähnlichkeit zwischen Inhaltsobjekten im Vektorraum aufgrund ihrer Beschreibung durch Merkmale.

Bestimmung der Ähnlichkeit anhand von Attributen

Auch wenn relational abgelegte Attribute in der Regel bei der Suche mit genauen Anfragen und exakten Ergebnissen (exact-match, singel-match [HS99]) eingesetzt wird, kann hier mittels Einführung einer Unschärfe Ähnlichkeit ermittelt werden. Um dies zu unterstützen kann die Anfragesprache um Fuzzy-Prädikate ergänzt und das Datenbanksystem entsprechend erweitert werden [ACL97]. Werden z.B. die Namen der Personen eines Bildes als Attribut abgelegt, kann dem Attribut zusätzlich ein oberer und unterer Wert zugewiesen werden, der ausdrückt wie wahrscheinlich es ist, dass es sich um diese Person handelt (siehe Tabelle 2.3).

Bild ID	Name	p-	p+
001	Jim Hatch	0.77	0.83
001	Dave Fox	0.17	0.23
002	John Lee	0.72	0.78
002	Ken Yip	0.12	0.18
003	John Lee	0.97	1.00
...

Tabelle 2.3: Zuordnen von Unschärfe zu Inhaltsobjekten (Quelle: [Sub98])

Durch das Einführen einer Bedingung, welche den Übereinstimmungsgrad der Anfrage mit der Ergebnismenge festlegt (unscharfe Anfrage), kann eine Ähnlichkeit bestimmt werden. Damit können nicht nur die in Tabelle 2.3 gemachten Angaben genutzt werden, sondern auch einzelne Attribute durch andere ersetzt werden. Wenn einer Anfrage „Bild mit einem Mann und einem kleinen Hund“ eine Bedingung hinzugefügt wird die eine Ähnlichkeit von mindestens 50% vorgibt sind beispielsweise auch Bilder dabei die einen großen Hund enthalten.

Ähnlichkeit von Bildern im Vektorraum

Durch die das Bestimmen einer Distanzfunktion in einem Vektorraum wird eine Metrik¹ festgelegt. Die Metrik bestimmt die Distanz zwischen Bildern in einem Vektorraum. Werden Bilder mit einer hohen Ähnlichkeit zu einer Anfrage gesucht die ebenfalls als Punkt im Raum abgebildet ist, müssen die Punkte im Raum gefunden werden die dem Anfragepunkt am nächsten sind.

Für jedes Merkmal können entsprechend der unterschiedlichen Merkmal-Repräsentation mehrere Distanzfunktionen zur Verfügung stehen, die bei der Ähnlichkeitssuche, jeweils mit dem korrespondierenden Vektor des zu vergleichenden Bildes eingesetzt werden müssen [Bos00].

Unter Einbeziehung einer Menge von Distanzfunktionen kann das Modell 2.1 für die Suche auf extrahierten Bildinhalten zu einem CBIR-Modell erweitert werden ([RHM98]):

$$CBIR = CBIR(D, F, R, M) \quad (2.2)$$

Dabei ist M eine Menge von Methoden zur Messung der Distanz ($M = \{m_{ijl}\}$). Mit m_{ijl} als Methode l für die Merkmal-Repräsentation j des Merkmals i

Eine gängige Distanzfunktion zwischen zwei Punkten im Raum ist die euklidische Distanz mit der Messung der Distanz über zwei Dimensionen mit $d(b_1, b_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ [Bim99]. Grafisch dargestellt ist die Distanz in Abbildung 2.1.

¹diejenige Struktureigenschaft eines Raumes, durch die in ihm die Entfernung (der Abstand) zweier Punkte definiert wird (aus [Zwa99])

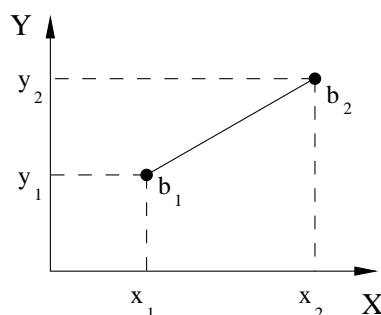


Abbildung 2.1: Euklidische Distanz dargestellt im 2-dimensionalen Raum

Die Distanzberechnung mit Euklid bestimmt mit den Vektoren nicht immer eine Metrik, in der Bilder die nahe beieinander liegen auch als ähnlich wahrgenommen werden. Für einige der Merkmal-Repräsentationen gibt es deshalb spezielle Verfahren zur Distanzmessung, die eine andere Anordnungen der Bilder im Vektorraum bewirken, Beispiele zur Distanzberechnung zwischen Farbhistogrammen sind in [RT01].

Bestimmung der Ähnlichkeit anhand von Inhaltsobjekten

Beim Einsatz von Ontologien für Inhaltsobjekte können zum Erhalten von Ähnlichkeitswerten zwischen den Inhaltsobjekten lexikalische Systeme herangezogen werden. Lexikalische Systeme, wie das der Universität Leipzig (zu finden unter <http://wortschatz.informatik.uni-leipzig.de/index.html>) beschreiben Ähnlichkeit zwischen Wörtern durch Graphen, die für die Einordnung von Inhaltsobjekten genutzt werden können. Eine weitere Möglichkeit die Ähnlichkeit von Inhaltsobjekten zu ermitteln ist es, die Ähnlichkeitsberechnung anhand der definierten und extrahierten Merkmale durchzuführen.

2.6 Vergleich und Bewertung von Retrieval Systemen

Um die Qualität eines Retrieval-Verfahrens einschätzen zu können, oder auch um unterschiedliche Systeme direkt miteinander zu vergleichen, muss eine Standardumgebung geschaffen werden, in der die Systeme getestet werden können, mit Richtlinien, die ein Vergleichen erst möglich machen. Zwei Beispiele aus dem Text-Retrieval sind OKAPI² (Open Kernel for Access to Protected Interoperable interactive services) und TREC³ (Text Retrieval Conference). Für die Suche in Bilddaten gibt es jedoch noch nichts vergleichbares, was einen direkten Vergleich nicht möglich macht.

²weiteres ist zu finden unter <http://www.cordis.lu/en/home.html>, unter http://www.igd.fhg.de/igd-8/projects/okapi/okapi_de.html

³ausführliche Information unter <http://trec.nist.gov/>

OKAPI ist eins von mehreren gemeinsamen Europäischen Programmen die durchgeführt wurden. OKAPI sieht sich als Test- und experimentelle Umgebung zum Erproben von Methoden bei der Entwicklung von Theorien im Bereich von Text Retrieval Systemen. Man will in dem Rahmen das Verständnis für das Suchverhalten von Nutzern nach Information erforschen [Rob97]

TREC will die Forschung innerhalb der Information Retrieval Gemeinschaft unterstützen, indem sie die Infrastruktur zur Verfügung stellt, die für die Auswertung der Text Retrieval Methoden notwendig ist. Durch das Zusammenbringen von Akademie, Industrie und Regierung erhoffen sie sich eine Beschleunigung des Entwicklungsprozesses. Für jede TREC wird eine große Test-Sammlung von Dokumenten und Anfragen zur Verfügung gestellt.

Zur Messung der Leistung eines Systems gibt es mehrere Bewertungsmethoden. Ein Beispiel ist die durchschnittliche Genauigkeit: Es wird gezählt wie viele relevante Medienobjekte bei 5 bei 10 oder bei 100 erhaltenen Dokumenten enthalten sind [CMRB01]. Eine weitere, häufig eingesetzte Messmethode ist das Erzeugen von Recall-Precision-Diagrammen.

Precision $P = \frac{R}{N_{Erg}}$ ist die Anzahl der erhaltenen, relevanten Medienobjekte R bezogen auf die Anzahl der Medienobjekte in der Ergebnismenge N_{Erg} , Recall $R = \frac{R}{N_R}$ ist die Anzahl der relevanten Medienobjekte bezogen auf die Datenbank [Bim99]. Der ideale Fall ist, wenn Recall und Precision jeweils bei 100 Prozent sind. D.h. bei 100 prozentigem Precision sind alle erhaltenen Medienobjekte relevant und entsprechend dem 100 prozentigen Recall sind alle relevanten Medienobjekte der Datenbank in der Ergebnismenge. In der Regel ist es jedoch so, dass bei sehr hohem Recall eine eher niedrige Precision zu erwarten ist, dies ist auf die Wahrscheinlichkeit zurück zu führen, dass bei einer genügend großen Ergebnismenge „irgendwann“ alle relevanten Medienobjekte enthalten sind. Im Gegensatz dazu sinkt bei einer höheren Precision meist das Recall, da relevante Medienobjekte, die eine etwas größere Distanz besitzen, nicht mehr in der Ergebnismenge enthalten sind. Die Precision drückt die Leistung des Systems aus, eine hohe Precision bedeutet auch eine hohe Dichte der relevanten Medienobjekte unter den ersten zurückgelieferten Medienobjekten. Mit der Messungen von Recall und Precision entstehen Diagramme wie sie in Abbildung 2.2 zu sehen sind.

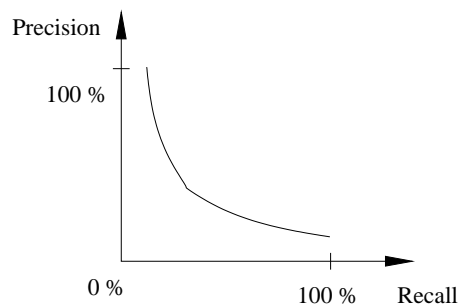


Abbildung 2.2: Beispiel für ein Recall-Precision-Diagramm

Für eine vergleichende Messung muss zu einzelnen Anfragen die richtigen Lösungen vorher bekannt sein. Ein Problem bei Bildern ist dabei, dass mitunter Testpersonen die Ergebnisse unterschiedlich interpretieren und unterschiedliche Bilder als relevant für die Ergebnismenge kennzeichnen. Eine korrekte Lösung sollte aber, für einen klaren Vergleich, widerspruchsfrei sein. Dazu sollte eine zuverlässige Standard-Datenmenge vorliegen, bei der die korrekte Lösung für ein Problem eindeutig definiert ist. Außerdem sollte die Datenbank groß genug sein, damit Statistiken mit einer ausreichenden Genauigkeit geführt werden können.

Tests dienen neben dem Vergleich von Systemen auch der Optimierung, deshalb sollten einstellbare Parameter während den Tests zusätzlich verändert werden, damit für jedes System eine optimale Lösung gefunden werden kann. Einstellbare Parameter sind z.B. die Anzahl der Farben eines Farbhistogramms (z.B. zwischen 16 und 24 Farben wechseln) oder die Anzahl der Filter mit der die Merkmalsausprägungen des Texturmerkmals ermittelt werden.

Kapitel 3

Grafische Anfragen an Bild-Datenbanken

Für die Suche auf Bildern ergeben sich entsprechend der internen Repräsentationen der Bildbeschreibungen mit Attributen, Freitext, Merkmalen und Inhaltsobjekten (siehe Abschnitt 2.4) unterschiedliche Verfahren die zur Suche eingesetzt werden können. Dazu werden in diesem Kapitel Retrieval-Verfahren vorgestellt, die anhand der vorgestellten Repräsentationen Bilder in der Datenbank Suchen. Dazu gehören das Daten-Retrieval auf relational abgelegten Daten, das Text-Retrieval für die Annotation der Bilder mit Freitext und das Bild-Retrieval auf Merkmalen. Insbesondere wird beim Bild-Retrieval auf die Wichtung von Merkmalen eingegangen. Das Einbeziehen der Wichtung ist bei einigen der Iterativen Verfahren für eine Reformulierung der Anfrage ausschlaggebend und soll deshalb in diesem Kapitel ausführlich eingeführt werden.

Ein Weiterer Punkt dieses Kapitels ist die Visualisierung der Anfrageformulierung und der Ergebnismenge. Dazu wird auf die allgemeinen Anforderungen an eine Nutzoberfläche eingegangen. Um grafische Anfragen zu realisieren müssen Methoden gefunden werden mit denen z.B. Ähnlichkeit und Merkmale visualisiert werden können.

3.1 Retrieval

Die Suche nach Bildern erfolgt auf den Metadaten der Bilder. Durch Anfragen auf den Metadaten werden Bilder „zurückgewonnen“, die der Anfrage ähnlich sind. Je nachdem, wie die Bilder durch die Metadaten repräsentiert sind, werden unterschiedliche Verfahren des Retrievals genutzt. Attribute können mit den Methoden des klassischen Daten-Retrievals (relationale Speicherung) gesucht werden. Bei Freitext zur Annotation der Bilder werden Methoden des Text-Retrievals eingesetzt. Die durch Extraktion direkt aus dem Bild gewonnen Merkmale werden mit den speziell für das Bild-Retrieval (Image Retrieval) entwickelten Verfahren durchsucht.

3.1.1 Daten-Retrieval (Data Retrieval)

Anfragen auf relational abgelegte Daten erfolgen in der Regel mit SQL (eine genaue Beschreibung der Anfragesprache ist beispielsweise in [HS95, HS99] zu finden) oder SQL-ähnlichen Anfragesprachen. Bei diesen Anfragesprachen besteht eine Anfrage in der Regel aus drei Hauptblöcken. Einen Block `select` der bestimmt, welche Attribute (Spalten in einer Tabelle) zurückgegeben werden sollen, ein weiterer Block `from`, der vorgibt, woher die Daten zu beziehen sind und ein Block mit Bedingungen, welche die Auswahl der Tupel (Zeilen einer Tabelle) bezüglich der gewählten Attribute einschränkt. Ein Beispiel dazu ist `select NAME, ORT from TAB_ADRESSEN where ORT = 'Magdeburg'`.

Die Aussage oder auch die Information die in einem relationalen Datenbankeintrag enthalten ist, ist eine Funktion der Datenbanksyntax (die durch das Datenbankschema bestimmt wird) auf der Semantik der Attribute [SJ00]. Werden in einer relationalen Tabelle die Bits eines Attributes eines bestimmten Tupels als die Zahl 2.000 interpretiert, ist bei einem entsprechenden Datenbankschema die enthaltene Information, dass der Angestellte, dessen eingetragener Name zu dem Tupel in Beziehung steht, 2.000 EURO im Monat erhält.

3.1.2 Text-Retrieval (Information Retrieval)

Im Text-Retrieval gibt es mehrere Modelle zur Suche in Texten (Annotationen der Bilder). Klassische Modelle sind, neben dem Vektormodell, das *boole'sche Modell* und das *probabilistische Modell* [BYRN99].

Das boole'sche Modell ist einfach und hat eine klare Semantik. Nachteil ist, dass ein Bild entweder relevant ist oder nicht, was das boole'schen Modell eher zu einem typischen Daten-Retrieval-Modell macht. Es wird keine ähnlichkeitsbasierte Übereinstimmung erkannt, was zu einer schlechten Leistung führen kann [BYRN99]. Im probabilistischen Modell wird demgegenüber die Wahrscheinlichkeit der Relevanz eines jeden Bildes abgeschätzt. Die Bilder der Datenbank werden fallend nach ihrer Wahrscheinlichkeit sortiert und als Liste zurückgegeben.

Baeza-Yates [BYRN99] bewertet das boole'sche Modell als das am wenigsten geeignete Modell, da keine direkte Ähnlichkeit gemessen werden kann. Vom probabilistischen Modell und Vektormodell gibt es nach Yates widersprüchliche Forschungsergebnisse die einmal das Vektormodell und in anderen Fällen das probabilistische Modell besser bewerten. Das Vektor-Modell wird aber in der Regel bei Texten als das überlegenere angesehen und vermehrt auch praktisch eingesetzt.

In der Regel kann bei einer Suche auf Text keine natürlichsprachliche Anfrage gestellt werden. Vielmehr werden Anfragen erwartet, die den Inhalt des Dokuments beschreiben, das voraussichtlich die Antwort auf das gegebene Problem enthält [BYRN99].

3.1.3 Bild-Retrieval (Image Retrieval)

Während Text- und Daten-Retrieval auf den entsprechenden Metadaten für die Suche von Bildern genutzt werden kann, gibt es für Metadaten wie Merkmale oder Inhaltsobjekte weitere Ansätze, die speziell auf die Eigenschaften der Bildbeschreibung ausgerichtet sind.

Das Ermitteln der Ähnlichkeit zwischen zwei Bildern bzw. von Bild und Anfrage, im hochdimensionalen Raum der Vektoren, ist sehr rechenaufwändig. Für eine effiziente und zielorientierte Suche werden in vielen Fällen die Dimensionen des Vektors auf eine Menge von aussagekräftigen Merkmalen oder Merkmalsausprägungen reduziert. Für die Wahl der einzusetzenden Merkmale werden Häufigkeitsverteilungen der Merkmalsausprägung in Bild, Anfrage und Datenbank genommen. Der Nutzer kann dabei durch Wichtung einzelner Merkmale und durch direkte Angabe von Werten für die einzelnen Merkmalsausprägungen einzusetzende Merkmale vorgeben. Die Wichtung ist eine Methode, den Bildbeschreibungs-Ebenen (siehe Kapitel 2.4.3) einen Relevanzwert zuzuweisen, welcher der Wichtigkeit der Ebene entspricht. Für die Wichtung der Merkmalsausprägungen werden im allgemeinen die Techniken aus dem Text-Retrieval übernommen. Eine übliche Relevanzbewertung einer Merkmalsausprägung ist die *inverse document frequency* idf (aus [RHM97]). Wird sie auf das Bild-Retrieval übertragen, ergibt sich für die Merkmalsausprägung m_{ijk} folgende Formel:

$$idf_{ijk} = \log_2 \frac{M}{df_{ijk} + 1} \quad (3.1)$$

Die Anzahl der Bilder in der Datenbank ist durch M gegeben und df_{ijk} ist die Anzahl der Bilder in der gesamten Datenbank, in der die Merkmalsausprägung m_{ijk} vorkommt. Die idf drückt die Aussagekraft einer Merkmalsausprägung aus. Merkmalsausprägungen die über die gesamte Datenbank verteilt sind und in vielen Bildern vorkommen, können keine entscheidenden Unterschiede in Bildern ausdrücken. Konzentriert sich eine Merkmalsausprägung auf ein paar wenige Bilder, steigt ihre Aussagekraft.

Bei der Berechnung der Ähnlichkeit zwischen zwei Bildern wird oft der Wert einer Merkmalsausprägung durch seine Wichtung ersetzt die sich mit der idf berechnen lässt:

$$W_{ijk} = mf_{ijk} \cdot idf_{ijk} \quad (3.2)$$

Mit der Multiplikation von idf mit der Häufigkeit mf_{ijk} der Merkmalsausprägung in dem Bild werden bei der Wichtung die Verteilung der Merkmalsausprägung in Bild und Datenbank berücksichtigt.

In der Regel werden Wichtungen auf jeder Ebene normalisiert. Das hat unter anderem zur Folge, dass sich die stark variierenden Werte, der Merkmalsausprägungen zwischen den einzelnen Merkmalsrepräsentationen nach der Änderung in eine normalisierte Wichtung in einem Intervall von $[0,1]$ befinden.

In der initialen Abfrage werden die Wichtungen durch den Nutzer angegeben. Ein Beispiel für die Wichtung von Merkmals-Vektoren ist $W_{Frb.,Hist.,rot} = 20\%$, bei dem

angegeben wird, dass im Merkmal „Farbe“ (Frb.) aus der Merkmalsrepräsentation „Histogramm“ (Hist.) die Merkmalsausprägung Rot 20% des Bildinhaltes ausmachen soll. In der Regel wird nicht vom Nutzer erwartet, die Merkmalsrepräsentation anzugeben, sondern wird automatisch durch das System ergänzt bzw. im Fall der Farbe die Ausprägung „20% rote Farbe“ auf alle Repräsentationen übertragen. Eine andere Möglichkeit ist die Wichtung des Merkmals, mit $W_{Kontur} = \frac{1}{3}$, $W_{Textur} = \frac{2}{3}$, danach ist die Übereinstimmung der Textur doppelt so wichtig, wie die Übereinstimmung der Kontur. Die Summe aller Wichtungen eines Merkmals oder einer Repräsentation ist immer 1 bzw. 100%. Gibt der Nutzer als einzige Bedingung 20% in der Farbverteilung an, sind die restlichen 80% der Farbverteilung sowie die gesamte Verteilung von Textur und Kontur beliebig.

Bei Ausführung der Anfrage werden die Wichtungen direkt in die Ähnlichkeitsberechnung mit einbezogen. Bei der Ähnlichkeitsberechnung zwischen Anfrage und Bild wird die Ähnlichkeit der korrespondierenden Merkmalsausprägungen bestimmt und deren Wichtung in die Ähnlichkeitsberechnung mit einbezogen. Die daraus berechnete Ähnlichkeit der Merkmalsrepräsentation wird in die Berechnung der Ähnlichkeit zwischen den Merkmalen von Anfrage und Bild einbezogen. Die Gesamtähnlichkeit zwischen Anfrage und Bild errechnet sich aus den Wichtungen der Merkmale in der Anfrage und der Ähnlichkeit.

Der Vorgang der über die drei Ebenen der Bildbeschreibung geht (siehe Abschnitt 2.4.3) ist in Abbildung 3.1 dargestellt, wobei die Aufspaltung von Anfrage und Vergleichsbild in die einzelnen Ebene angezeigt ist und die Zuweisung der Wichtungen in den einzelnen Ebenen dargestellt ist.

Folgend werden die einzelnen Rechenschritte der Ähnlichkeitsmessung beschrieben. Bei einem Bild-Vektor m und einem Anfrage-Vektor q ergibt sich die Ähnlichkeit $S_{m,n}$ mit der Distanzfunktion m_{ijl} aus der Menge M (entsprechend der Formalisierung in 2.2) und den Wichtungen der K Merkmalsausprägungen der jeweiligen J Merkmalsrepräsentationen der I Merkmale zu:

$$S_{m,q}(r_{ij}) = m_{ijl}(r_{ij}, W_{ijk}) \quad (3.3)$$

Die Ähnlichkeiten der Merkmalsausprägungen werden dann mit den Wichtungen W_{ij} der Merkmalsrepräsentationen kombiniert zur Ähnlichkeit zwischen den einzelnen Merkmalen:

$$S_{m,q}(f_i) = \sum_j W_{ij} S_{m,q}(r_{ij}) \quad (3.4)$$

die dann im letzten Schritt zur Gesamtähnlichkeit zwischen den Merkmals-Vektoren von Bild und Anfrage kombiniert werden mit

$$S_{m,q} = \sum_i W_i S_{m,q}(f_i) \quad (3.5)$$

Die letzten beiden Schritte 3.4 und 3.5 können nach [RHM98] zu

$$S_{m,q} = \sum_i \sum_j W_{ij} S_{m,q}(r_{ij}) \quad (3.6)$$

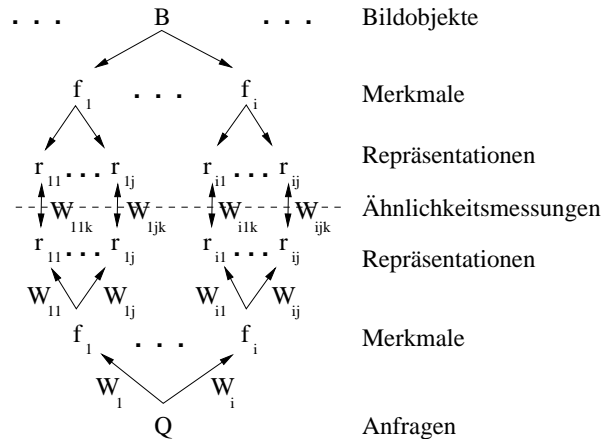


Abbildung 3.1: Retrieval-Prozess im Merkmal-Raum

Im Gegensatz zur Informationssuche mit den Methoden des Daten-Retrievals, kann mit dem Text- und Bild-Retrieval keine direkte Gewinnung von Information aus den Daten und Metadaten erfolgen. Die durch Merkmale im System interpretierbare Bedeutung der Inhalte beruhen auf der Syntax der Merkmale, so ist die Semantik einer Sonne und eines Fußballs an Hand des Kontur-Merkmales gemessen gleich [SJ00].

3.1.4 Vergleich von Daten-Retrieval mit Text- und Bild-Retrieval

Im folgenden sollen Text- und Bild-Retrieval (zusammengefasst unter dem Begriff *Multimedia-Information-Retrieval* mit dem Daten-Retrieval verglichen (angelehnt an [Rij79]) werden. Als Überblick sind die Unterschiede in der Tabelle 3.1 zusammengefasst.

Im Daten-Retrieval erfolgen Anfragen z.B. auf RDB. Anfragen an RDB sind abgestimmt auf eine bekannte Strukturierung der Inhalte. Die gesuchte Information erschließt sich aus der Struktur des definierten Datenschemas und den Werten der enthaltenen alphanumerischen Daten. Die Ergebnismenge enthält exakt die Information, die in der Anfrage beschrieben wurde. Bei digitalen Medien wie Bildern, Audio usw., auf die im "Rohzustand" keine Anfrage wie im Daten-Retrieval erfolgen können, ist die Suche abhängig von den ermittelten Merkmalen oder einer zugehörigen Beschreibung, die aufgrund von Unvollständigkeit und/oder Mehrdeutigkeit der daraus interpretierbaren Information, nur angenäherte bzw. unscharfe Ergebnismengen erlaubt, in der dann relevante Bilder sein können.

Eine RDB enthält Informationen, die explizit aufbereitet und eingepflegt wurden. Das Daten-Retrieval basiert auf dieser vollständigen Informationsrepräsentation. Bei Bilddatenbanken ohne Annotationen muss nicht einmal die Person, welche die Bilder einpflegt über Wissen der Bildinhalte verfügen, da die Merkmalsextraktion automatisch erfolgen kann. Merkmale beschreiben Eigenschaften der Bildinhalte, jedoch in der Regel, nicht die Information, nach der gesucht wird.

	DR	MMIR
Ergebnismenge und Anfrage	exakt	unscharf
zugrundeliegende Information	vollständig	beschreibend, unvollständig
Anfrageablauf	ein Schritt	iterativ
Fehlertoleranz	keine	abhängig
Verfahren	deterministisch	probabilistisch, unscharf ...

Tabelle 3.1: Vergleich Data Retrieval mit Information Retrieval (angelehnt an [Rij79])

Im Multimedia-Information-Retrieval ist die Suche ein Prozess, der oft erst nach mehreren Anfragen zum gewünschten Ziel führt, wogegen im Daten-Retrieval auf die direkt Anfrage nur die eine mögliche Ergebnismenge ausgegeben wird. Fehlendes Wissen über die gesuchte Information führen beim Daten-Retrieval zum Scheitern der Anfrage. Beim Multimedia-Information-Retrieval hingegen kann während des Suchprozesses fehlendes Wissen aus relevanten (vom Nutzer markierten) Bildern in Form von Merkmalen gewonnen werden. Demnach ist die Fehlertoleranz unter anderem von der Gewinnung und Beurteilung der ergänzenden Merkmale und Bilder abhängig.

3.2 Benutzerschnittstelle und allgemeine Informationsvisualisierung

Benutzerschnittstellen sind Systeme zur Informationsübermittlung zwischen Programm und Benutzer. Durch eine grafische Benutzerschnittstelle kann das Verändern von Variablen und das Ausführen von Befehlen durch direkte, grafische Manipulation und Interaktion am Bildschirm erfolgen. Vorteile der grafischen Benutzerschnittstelle resultieren aus den besonders visuellen, kognitiven Fähigkeiten des Menschen. Der Mensch kann visuelle Information schnell aufnehmen und sie über seine Fähigkeiten der Reduzierung, Mustererkennung und Verarbeitung auch schnell nutzen (aus [Lan94]). Die Abgabe von Information bewältigt der Mensch jedoch wesentlich langsamer. Die Bilanzen des Informationsflusses sind in Abbildung 3.2 zu sehen. Bei ständiger visueller Rückmeldung erhält der Benutzer eine bessere Kontrolle und einen besseren Überblick über durchgeführte Aktionen [Jan93]. Die grafische Benutzerführung sollte intuitiv erfolgen, d.h. der Nutzer sieht, welche Aktionen bisher erfolgt sind, in welchem Ausführungsschritt er

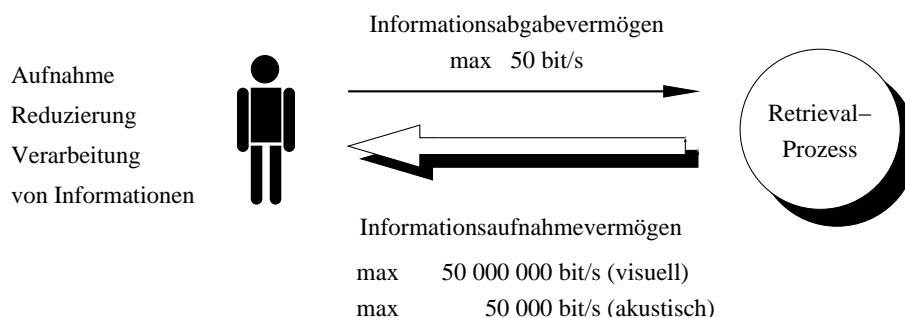


Abbildung 3.2: Informationsfluss Mensch Maschine (Quelle: [Lan94])

sich befindet, welche Schritte er als nächstes machen kann, wie er sie machen kann und was er dadurch erreicht [Thi94].

Entsprechend den Vorgaben gibt es mehrere Kommunikationsparadigmen. Dazu gehören grafische Manipulation, Zeiger- und Fenstertechniken und der Einsatz von Icons. Die Auswertung der Eingaben bei direkt manipulativen Benutzerschnittstellen erfolgt sofort (interaktionsfrei) oder nach weiteren Eingaben durch den Nutzer (asynchron) über Dialoge [Jan93].

Bezogen auf MMDB gibt es für die einzelnen Medienobjekte in der Regel, für die Präsentation und Bearbeitung eigene Anwendungen mit eigenen Benutzerschnittstellen. Für eine Multimedia-Benutzerschnittstelle, welche die Funktionen der einzelnen Schnittstellen vereint, müssen die unterschiedlichen Techniken, zur Bearbeitung der Medienobjekte in eine allgemeine Form übertragen werden. Damit soll eine einheitliche Steuerung mit möglichst wenigen Werkzeugen geschaffen werden.

Das in [Lan94] vorgestellte Seeheim-Modell für grafische Benutzerschnittstellen für Prozesssteuerungen lässt sich für dieses Problem einsetzen. Das Modell trennt die Benutzerschnittstelle in Präsentationskomponente, Anwendungsschnittstelle und Dialogkontrolle. Dabei werden Bildschirmverwaltung und Interaktionstechniken von der Präsentationskomponente übernommen. Die Dialogkontrolle ist für die Konvertierung der Aktionen des Nutzers in eine interne Darstellung zuständig, aus der die gewünschten Operationen des Nutzers erkennbar sind [Lan94]. Die Anwendungsschnittstelle koordiniert die Verbindung zwischen der Benutzerschnittstelle und den anderen Komponenten des Systems. Sie stellt die Nutzersicht auf das System dar mit der Beschreibung von internen Strukturen und Funktionen.

Die Abbildung der Funktionen ermöglichen die Interaktion zwischen Mensch und Maschine. Es werden dazu in [Lan94] drei Arten der Kommunikation angegeben: Funktionen, die vom Benutzer im System ausgelöst werden (Anfrage), Funktionen die in der Benutzerschnittstelle durch das System ausgelöst werden (Ergebnispräsentation) und eine Kombination der beiden ersten Kommunikationstypen (direktes Umsetzen von Benutzeraktionen mit Präsentation auf der Benutzerschnittstelle).

Bei der Bedienung der Benutzerschnittstelle kann zwischen signalorientierter und informationsorientierter Bedienung unterschieden werden. Signalorientierte Bedienung

beinhaltet das Anzeigen der ausgeführten Aktionen des Nutzers und des veränderten Ergebnisses. Die Aussagekraft in Bezug auf das gewünschte Ergebnis muss der Nutzer selbst abschätzen. Dazu muss der Nutzer sich ein Bild von der Ausführung der Anfrage im Datenbanksystem machen können. Das erfolgt entweder durch Erlernen der Aussagekraft der Ergebnisse durch den Nutzer (was während der iterativen Schritte eines Anfrageprozesses erfolgt) oder ist durch entsprechendes Fachwissen über die interne Ausführung der Anfragen bereits vorhanden.

Bei der informationsorientierten Bedienung wird eine Zusammenfassung der Signale (Aktion und Ergebnis) und die Ausführung abgebildet und beides für den Nutzer zu greifbar und verständlich dargestellt. Dazu gehört, dass die Information die dem Nutzer präsentiert wird auf ein notwendiges Maß reduziert wird. Das Anzeigen von ausschließlich relevanter Information ist der Idealfall einer Informationsreduktion. Übertragen auf Multimedia- und Bild-Datenbanken bedeutet das: Kein wertorientiertes Einstellen einzelner Gewichte, sondern zielorientiertes Vorgeben übergeordneter Größen oder das Beschränken auf die entscheidenden Einflussgrößen für die gewünschte Ergebnismenge.

Zu einer problem- oder situationsbezogenen Präsentation und Interaktion muss das System entsprechende Informationen vom Nutzer erhalten [Gap95]. Die Benutzeroberfläche muss demnach nicht nur so aufgebaut sein, dass eine möglichst selbsterklärende Ergebnispräsentation erfolgt, sondern auch so, dass Eingaben erfolgen können, welche die benötigten Informationen an das System vermitteln.

Die Darstellung sollte knapp, genau und übersichtlich sein, so dass die enthaltene Information erkennbar dargestellt wird. Erwähnenswert ist noch die Forderung von [FS95], dass der gesamte Bedienungsablauf kontinuierlich sein soll, damit der Nutzer nicht das Gefühl hat, "den Faden zu verlieren".

Langmann [Lan94] teilt die Visualisierung der Information in zwei Phasen. Die erste ist die konzeptionelle Phase mit der Modellbildung mit *unabhängigen Variablen* und *abhängigen Variablen*. Die unabhängigen Variablen ermöglichen die Einwirkung auf den Prozess. Im Fall der iterativen Anfrageformulierung sind die unabhängigen Variablen die Variablen die zur Erstellung der Anfrage notwendig sind. Der Prozess der damit beeinflusst wird, ist die Anfrageausführung.

Die abhängige Variablen sind die Zielgrößen. Bei Anfragen an eine Datenbank ist die abhängige Variable die Ergebnismenge mit allen Präsentationsangaben. Das dazugehörige Modell beschreibt die Abhängigkeiten der Variablen voneinander.

In der zweiten Phase erfolgt die Visualisierung. Die konzipierte Darstellung wird in konkrete Bildobjekte umgewandelt und die Variablen werden mit Bildobjekten verbunden für die Bilddynamik. Für eine direktmanipulative Oberfläche müssen die Variablen so dargestellt werden, dass sie einfach und intuitiv verändert werden können.

Die Präsentation der Variablen wird auf kombinierbare Graphikprimitive zurückgeführt [Gap95]. Graphikprimitive sind Achsen für Metriken, Tabellen für listenförmige Ausgabe von Daten und eventuell auch Kurvendarstellungen für voneinander abhängigen Größen. Dazu kommt ein beliebiges Einbinden von Graphiken, die Zusammenhänge "auf den ersten Blick" erkennen lassen.

Bedienung und Darstellung sollte in wenigen Fenstertypen mit gleichen Bedienungsmodellen erfolgen. Damit wird eine geringe Einarbeitungszeit erreicht und durch Standardisierung der Benutzerinteraktion Bedienungsfehler vermieden [Gap95].

Benutzerschnittstellen von MMDBS variieren sehr stark. Da z.B. viele MMDBS unterschiedliche Inhaltsbeschreibungen für die Suche verwenden und dadurch unterschiedliche Variablen zur Grundlage haben. Außerdem werden für die unterschiedlichen iterativen Verfahren der Anfrageformulierung unterschiedliche Werkzeuge benötigt. Werden z.B. Anfragen manuell reformuliert, werden Dialoge für die Veränderung von Anfragebedingungen als Unterstützung eingesetzt. Wird im Gegensatz dazu für die Iteration ein Bewertungsverfahren wie das Relevanz-Feedback eingesetzt muss nur die Ergebnismenge aufbereitet werden und zwar so, dass der Nutzer eine Bewertung an ihr durchführen kann.

Fragen die für die Visualisierung geklärt werden müssen, ist vor allem die Reduktion der Information: Welche Merkmale oder andere Bildinhaltsbeschreibungen sind für den Nutzer bei der Anfragerreformulierung relevant? Wie groß soll die Ergebnismenge werden und wie wird sie angeordnet damit der Nutzer möglichst schnelle eine Aussage zur Ergebnismenge machen kann. Ein weiterer Punkt ist die Frage wie der Nutzer die Variablen verändert, wie er z.B. durch direkte Manipulation Einfluss auf die Wichtung nimmt. Im nächsten Abschnitt werden dazu einzelne Präsentationsmöglichkeiten betrachtet.

3.3 Visualisierung von Anfrage und Ergebnismenge

Die für die initialen Anfragen und die Anfragerreformulierungen benötigten Angaben können über grafische Oberflächen gemacht werden. Die abhängigen und unabhängigen Variablen für eine Reformulierung einer Anfrage sind unter anderem auswählbare Merkmale, Inhaltsobjekte und Begriffe. Die Variablen bedingen oft weitere Variablen, z.B. sind mit den Merkmalen die veränderbaren Wichtungen verbunden und für Inhaltsobjekten muss die räumliche Anordnung als Variable veränderbar sein.

Unter Umständen kann auch zwischen unterschiedlichen Distanzmessungen gewählt werden bzw. den zur Distanzmessung eingesetzten Metriken. Die unterschiedlichen iterativen Verfahren der Anfrageformulierung bedingen auch verschiedene Variablen zur Einstellung (Bewertung, verändern von Werten, hinzufügen von Anfragebedingungen). Für eine veränderbare Darstellung der Ergebnismenge sollten Einstellung bezüglich Anzahl der Ergebnisbilder, ihre Position und Größe beeinflussbar sein.

Die Visualisierung von relational abgespeicherten Daten kann über ein Formular erfolgen. Jedes Attribut einer Relation wird dabei durch ein Textfeld im Formular repräsentiert. Der Nutzer kann die erwarteten oder gesuchten Größen von Hand eintragen. Bei Attributen mit vorgegebenem Vokabular wie z.B. ja/nein oder Farbnamen, können z.B. Auswahllisten angeboten werden.

Inhaltsobjekte benötigen einen Editor in dem räumliche Anordnungen vom Nutzer vorgegeben werden können. Dabei müssen die Inhaltsobjekte in einer abstrahierten Form

dargestellt werden. Manche Systeme bieten das erstellen von Icons für die Inhaltsobjekte an oder es wird ein allgemeines Symbol für ein Inhaltsobjekt verwendet (z.B. Kreis) dem eine entsprechende Bezeichnung zugeordnet wird. Ein Beispiel für einen solchen Editor ist in Abbildung 3.3 zu sehen

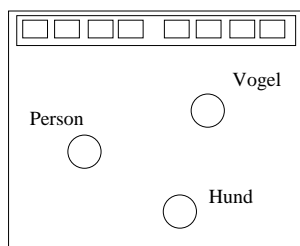


Abbildung 3.3: Editor für die Anordnung und Benennung von Inhaltsobjekten

Für Merkmale besteht die Möglichkeit unterschiedliche Texturen und Konturen durch Beispielbilder darzustellen unter denen der Nutzer dann auswählt. Bei der Auswahl einer Farbe sind Farbpaletten, die auch in Zeichenprogrammen eingesetzt werden, geeignet.

Durch die Auswahl einer Merkmalsausprägung (z.B. kariert, rund, rot) werden den entsprechenden Merkmalen Wichtungen bezüglich der Merkmalsausprägungen zugeordnet. Für eine allgemeinere Wichtung von Bildinhalten besteht die Möglichkeit, dass der Nutzer prozentuale Verteilungen als Wichtung vorgibt. Beispielsweise kann mit der Angabe „Textur 20% und Farbe 40%“ festgelegt werden, dass bei der Ähnlichkeitsberechnung die Textur mit einer Wichtung von 20% eingehen soll. Die Farbe ist mit einer Angabe von 40% wesentlich wichtiger. Mit anderen Worten, Bilder die in der Textur mit der Anfrage übereinstimmen werden in einer Ergebnisliste weiter hinten angeordnet als Bilder die in der Farbe übereinstimmen.

Die Visualisierung des hochdimensionalen Anfrageraumes bzw. die damit verbundene Ergebnismenge ist nicht direkt möglich, da nicht mehr als 3 Dimensionen simuliert werden können. In der Regel wird eine Ergebnismenge mit den gefundenen Bildern als eine nach der Ähnlichkeit der Bilder sortierten Liste von Thumbnails ausgegeben. Entspricht der Anfrageraum der gesamten Datenbank können auch Zugriffsstrukturen als Visualisierungsform eingesetzt werden.

In einer Liste können Bilder, die im Raum sehr nahe beieinander liegen, getrennt voneinander auftauchen, da Ähnlichkeit nur von einem gegebenen Punkt aus gemessen wird. Somit gehen andere Beziehungen verloren.

In Abbildung 3.4 ist dazu rechts ein Anfrageraum mit zwei Dimensionen angezeigt. Der Buchstabe q kennzeichnet den Anfragepunkt im Vektorraum. Die mit b gekennzeichneten Punkte stellen die Vektoren der Bilder in der Datenbank dar. Die Bilder sind durch Gruppierungen in Cluster mit Großbuchstaben eingeteilt. In der links dargestellten Liste ist zu sehen, wie sich zusammengehörige Bildgruppen über die ganze Liste verteilen. Grund dafür ist, dass die Bilder nur nach ihrer Ähnlichkeit zum Anfragepunkt sortiert werden und so die Nachbarschaft zu anderen Bildern nicht berücksichtigt wird.

Beim Einsatz von Clustern (Abbildung 3.4 Bildmitte) für die Visualisierung tritt das Problem auf, dass Bilder, die sich ähnlich sind, zwar in Gruppen zusammengefasst werden, aber Bilder die am Rand des Clusters liegen mitunter näher zu Bildern des Nachbar-Clusters liegen als zu Bildern, die im selben Cluster sind. Abhängig davon wie die Cluster gebildet werden und wie die Zugriffsstruktur aufgebaut ist, können einzelne Cluster weit auseinander liegen die eigentlich dicht beieinander liegen.

Nur die räumliche Anordnung (Abbildung 3.4 rechts) kann die komplexen Bezüge aufzeigen. Bei der Reduktion des hochdimensionalen Raumes, auf einen 2-dimensionalen (bzw. pseudo-3-dimensionalen) Raum für die Benutzeroberfläche geht Information verloren. Die reale Anordnung der Bilder wird durch die Reduktion der Dimensionen entweder verzerrt oder geht ganz verloren.

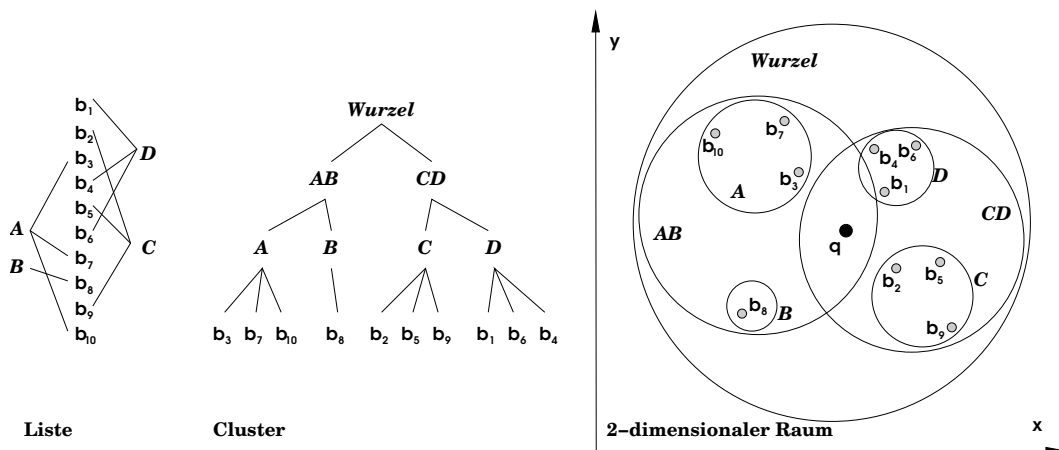


Abbildung 3.4: Darstellungsformen einer Ergebnismenge zu der Anfrage q

Durch das Projizieren bzw. transformieren von einzelnen Dimensionen in eine z.B. 2-dimensionale Ansicht werden nicht nur die realen Distanzen zwischen den Bildern verzerrt, sondern es gehen auch Beziehungen zwischen den Bildern verloren, da eine vollständige Integration aller Dimensionen nicht möglich ist [RT01]. Für die Reduktion der Dimensionen gibt es z.B. das *Multi-dimensional Scaling*. In [RGT97] wird dazu ein Verfahren vorgestellt das bei möglichst kleinem Informationsverlust möglichst viele Dimensionen auf wenige reduziert.

Eine ideale Reduzierung erzeugt eine Anordnung der Bilder zueinander, die der menschlichen Wahrnehmung entgegenkommt. Ein Beispiel dazu ist in Abbildung 3.5 dargestellt. In der Anordnung hat die Textur von rechts nach links eine gröbere Körnung und von unten nach oben lässt sich eine zunehmende Ausrichtung der Textur erkennen.

Durch anbieten von mehreren Darstellungen können unterschiedliche Schwerpunkte bei der Präsentation gesetzt werden. Beispielsweise können die Bilder auch nach unterschiedlichen Farbverteilungen angeordnet sein. Farbverteilungen können beispielsweise auch direkt als Achsen für die Präsentation eingesetzt werden. In dem Fall erfolgt kein *Multi-dimensional Scaling* sondern werden Dimensionen des Vektorraums einfach entfernt.

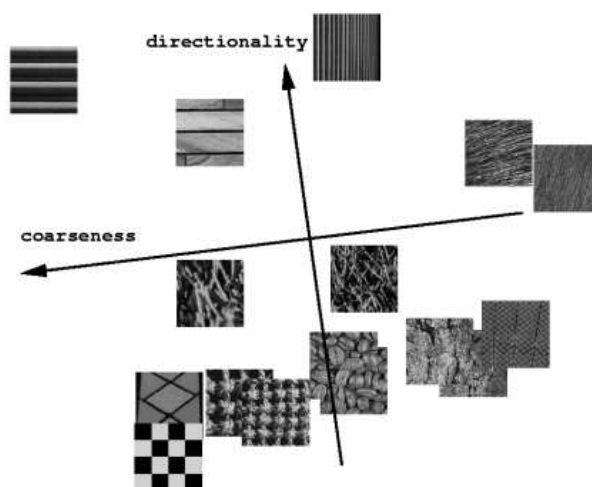


Abbildung 3.5: 2-dimensionale Visualisierung

Die 2-dimensionale Darstellung kommt der menschlichen Beurteilung entgegen. Die Beurteilung ist subjektiv und steht immer in dem Kontext, in dem sie gemacht wird. Ein sechsstöckiger Wohnblock in der Umgebung von Einfamilienhäusern, wird als sehr groß wahrgenommen. Befindet sich der Wohnblock aber in der Umgebung von Wolkenkratzern ist er eher ein kleines Gebäude.

Der Kontext, in dem die Bilder in der Datenbank stehen, ist abhängig von den extrahierten Merkmalen und der eingesetzten Distanzmessung [SGJ01]. Durch den Vektor der Merkmale und die Distanz ist die Anordnung der Bilder im Raum festgelegt (Metrik). Jenachdem, mit welchen Merkmalen die Ähnlichkeit bestimmt wird oder mit welcher Funktion die Ähnlichkeit berechnet wird, verändert sich der Kontext. Werden Bilder einzeln betrachtet, fehlt der Kontext in dem sie in der vorgegebenen Metrik stehen. Dadurch ist es relativ schwer, eine angemessene Änderung der Anfrage durchzuführen [SJ00] (ist der sechsstöckige Wohnblock bezüglich des gesamten Datenbankinhaltes klein oder groß?).

Ein besonderer Vorteil der räumlichen Darstellung ist, dass wesentlich mehr Bilder angezeigt werden können, ohne dass der Nutzer die Übersicht verliert. Bezüglich der Übersicht ist eine 2-dimensionale Anzeige der Bilder geeigneter da der Nutzer alle Bilder gleichzeitig sieht, was bei drei Dimensionen nicht immer der Fall ist. Außerdem ist das drehen einer 3-dimensionalen Darstellung wesentlich rechenaufwändiger als das Neuberechnen der Darstellung mit zwei anderen Dimensionen. Es muss nicht jedes Bild einzeln betrachtet werden, da Bilder schon geclustert erscheinen und der Zusammenhang direkt visuell aufgenommen werden kann ([RT01]). Das bedeutet auch, dass allein durch die Menge bedingt, höhere Werte für Recall erhalten werden können. Eine hohe Precision ist für Ergebnislisten mit wenigen Bildern wichtiger als für die 2-dimensionale Darstellung. Bei einer Precision von 20 % sind in einer Liste mit 10 Bildern nur 2 relevante Bilder dabei, bei einer 2-dimensionalen Darstellung mit 50 bis 100 Bildern sind es 10 bis 20.

Kapitel 4

Iterative Veränderung der Anfrage

Die Reformulierung der jeweilig aktuellen Anfrage in einzelnen Iterationsschritten ist anhand der zur Verfügung stehenden Variablen möglich. Dazu gehört das Verändern, Hinzufügen oder Entfernen von Bedingungen der Anfrage bezüglich der Inhaltsbeschreibung der Bilder, Schlüsselwörter und Berechnungsmethoden für Ähnlichkeit und Reformulierung. Mit jedem Iterationsschritt werden durch Nutzer und/oder System unterschiedliche Veränderungen durchgeführt, die zu einer Präsentation von neuen Ergebnissen führen. Bezogen auf die Anordnung der Bilder in einem Merkmalsraum kann sich dadurch die Position des Anfragepunktes verändern oder auch die Ausmaße des aufgespannten Anfrageraums. Je nachdem, wie direkt der Einfluss auf die Veränderung im Merkmalsraum oder auf die Anfrage ist, ergibt sich eine Abstufung im Abstraktionsgrad der Anfrageformulierung.

Ein für das Bildretrieval besonderer Vorteil der Iteration ist die Möglichkeit Rechenzeiten während eines Anfrageprozesses zu reduzieren. Dazu werden beim Einsatz von Beispielbildern schnell berechenbare Merkmale im ersten Schritt aus den Bildern der Anfrage extrahiert und in weiteren iterativen Schritten durch Merkmale von Bildern aus der Datenbank ergänzt, die schon beim Einfügen der Bilder in die Datenbank extrahiert wurden.

Die Merkmale aus der Datenbank werden aufgrund ihrer Häufigkeitsverteilung in Bildern der Anfrage und der Datenbank und anhand der Benutzerangaben während den einzelnen Iterationsschritten ausgewählt [SMMP00].

Auf das Problem der Auswahl von Merkmalen und Merkmalsausprägungen wird in Abschnitt 4.4 näher eingegangen. Da der Rechenaufwand sehr groß ist, wenn alle Merkmalsausprägungen eines Bild-Vektors in die Ähnlichkeitsberechnung mit einbezogen werden, werden in der Regel die Dimensionen des Vektors reduziert. Ein gutes Reduktionsverfahren bewertet Merkmalsausprägungen mit hohem Informationsgehalt hoch und entfernt Merkmalsausprägungen die nicht relevant für die Ähnlichkeit von ausgewählten Bildern ist.

Mit der Iteration als Suchstrategie bieten sich Verfahren an, die mit einer anfänglichen sehr groben oder auch nur partiell definierten Anfrage beginnen können und die in folgen-

den Schritten durch System und Nutzer verfeinert und ergänzt werden. Der Nutzer kann sich somit bei der Anfrageformulierung auf Merkmale beschränken die der menschlichen Wahrnehmung nahe kommen und die er direkt in Beziehung mit der gewünschten Ergebnismenge setzen kann. In unterschiedlichen Systemen ist dazu die Systemunterstützung unterschiedlich stark ausgeprägt, was zu unterschiedlichen Abstraktionsformen der Anfrageformulierung führt.

Im vorliegenden Kapitel werden vier Ansätze zu iterativen Verfahren behandelt:

- Browsen/Navigieren,
- manuelle Anfrageformulierung,
- Relevanz-Feedback und
- Kontext-Feedback

Kriterien, nach denen die Ansätze betrachtet werden, sind der Abstraktionsgrad der Anfrageformulierung (hierbei liegt der Schwerpunkt auf Nutzeraktionen) und die Ausführung im System. Als weiterer wichtiger Punkt werden Methoden zur iterativen Veränderung der Wichtung und deren Auswirkung auf die Ergebnismenge betrachtet.

Die Abschnitte 4.5 bis 4.7 gehen auf die Ansätze Browsen, Relevanz-Feedback und Kontext-Feedback ein.

4.1 Abstraktionsgrad der Anfragereformulierung

Wird der Zugriff des Nutzers auf Daten der Datenbank betrachtet, können unterschiedliche Abstraktionsstufen der iterativen Anfrageformulierung unterschieden werden. Eine steigende Abstraktion des Eingriffs bedeutet für den Nutzer, dass er immer weniger von interner Strukturierung und Verwaltung wissen muss und die Suche sich immer mehr auf Kriterien stützt, die den Suchvorgaben des Nutzers entsprechen.

In Abhängigkeit der Herangehensweise bei der Suche in einer Bildmenge, lassen sich die vier iterativen Verfahren zu vier unterschiedlichen Abstraktionsebenen zuordnen. Das Browsen entspricht der untersten Ebene, gefolgt von manueller Anfrageformulierung, Relevanz-Feedback und Kontext-Feedback.

Browsen/Navigieren (1.Abstraktionsebene)

Auf der untersten Ebene erfolgt bei der Suche der direkte Zugriff auf systeminterne Zugriffsstrukturen und somit direkt auf die abgelegten Bilder.

Diese Strategie wird benötigt, wenn nur eine vage Vorstellung von dem gesuchten Bild vorliegt. Die Bilder der Datenbank können direkt, d.h. ohne inhaltsorientierte Anfrage, durchsucht werden.

Beim einfachen Browsen oder auch Navigieren durch die Bilder der Datenbank bestimmt der Nutzer direkt die Position und die Ausmaße des Anfrageraums bzw. der präsentierten Ergebnismenge. Eine gute Benutzerführung ist abhängig von einer geeigneten Präsentation der Zugriffsstrukturen. Entsprechend dem Zugriff auf die Bilder kann der Datenbankinhalt schrittweise durchforstet werden, ohne ein Ziel vorgeben zu müssen. In Abschnitt 4.5 erfolgt eine genauere Beschreibung des Browsens.

manuelle Anfrageformulierung (2.Abstraktionsebene)

Die nächst höhere Form der Abstraktion ist die Beschreibung der gesuchten Bilder anhand von wählbaren Parametern die sich auf die Metadaten der Bilder beziehen. Dazu müssen dem Nutzer die Parameter bekannt sein oder durch die Benutzeroberfläche bekannt gemacht werden. Diese Art der Suche erfüllt direkt die Ad-hoc-Bedingung von Anfragesprachen wie es aus relationalen Anfragesprachen bekannt ist (siehe Abschnitt 2.3).

Die einfachste Variante der manuellen Iteration ist, das Ergebnis in einem neuen oder separaten Fenster anzuzeigen. Dabei bleibt die abgeschickte Anfrage erhalten und die grafisch erstellte Anfrage kann direkt mit der erzeugten Ergebnismenge verglichen werden. Diese Strategie wird von CHABOT [OS95] und MOQL [OXO98] verwirklicht. In diesen Systemen liegt es am Nutzer, die Anfrage nach seinem Ermessen zu verändern, um die gewünschte Zielmenge zu erhalten. Eine Verbesserung dieses Ansatzes stellt eine erweiterte Antwort des Systems auf die Anfrage dar, die zusätzlich zur Ergebnismenge weiterführende Information enthält. Dazu können neben Ähnlichkeitsangaben Informationen über Alternativen zu einzelnen Bedingungen der Anfrage gehören. Damit soll dem Nutzer Information über die allgemeinen Datenbankinhalte zur Verfügung gestellt werden. Alternativen sind z.B. das Auflisten von Inhaltsobjekten die ähnlichen zu denen aus der Anfrage des Nutzers sind. Bei der Angabe „Kind links oben im Bild“ könnte dann ein Alternativvorschlag „Mensch links oben im Bild“ oder „Kind rechts im Bild“ sein.

Durch direkten visuellen Vergleich von Anfrage und Ergebnismenge kann z.B. die Korrektur oder Anpassung von Farbangaben in der Anfrage leichter erfolgen. Manuelle Iteration erfolgt in der Regel im Zusammenhang mit der manuellen Formulierung einer initialen Anfrage, die dann in den iterativen Schritten verändert wird.

Die manuelle Iteration kann nur dann effektiv ausgeführt werden, wenn eine Vorstellung von der Zielmenge vorliegt, die so genau ist, dass klare Aussagen über die zu erwartenden oder bezüglich der Ergebnismenge zu ändernden Bildinhalte gemacht werden können.

Relevanz-Feedback (3.Abstraktionsebene)

In der dritten Abstraktionsebene ist nur noch begrenztes Wissen über die Parameter notwendig. Der Nutzer gibt an das System eine Beurteilung der gelieferten Ergebnisse, dazu bewertet er einzelne Bilder oder Bildinhalte aus der Ergebnismenge.

Beim Relevanz-Feedback wird in der Regel nur eine grobe oder partielle Definition

als initiale Anfrage erwartet. Das Relevanz-Feedback ermöglicht den effizienten Einsatz großer Merkmalszahlen und wahrnehmungsfremder Merkmale, da die Verfeinerung durch Wichtung und die Erweiterung durch Merkmale vollständig vom System übernommen wird. Der Nutzer hat nur noch eine bewertende Funktion. Damit ist der Abstraktionsgrad der iterativen Anfrageformulierung höher als bei der manuellen Iteration. Die interne Berechnung neuer Anfragen erfolgt durch Anpassen der Berechnung an den Nutzerbedarf. Das Erlernen des Nutzerbedarfs kann intern erweitert werden, indem das Benutzerverhalten beobachtet wird und daraus Suchstrategien entwickelt werden. Auf das Einbeziehen des Nutzerverhaltens wird in Abschnitt 4.3.1 näher eingegangen und in Abschnitt 4.6 wird das Relevanz-Feedback näher behandelt.

Kontext-Feedback (4. Abstraktionsebene)

Die letzte der Abstraktionsebenen, das Kontext-Feedback, ist entsprechend der Bezeichnung kein Bewerten von einzelnen Inhalten, sondern ein Vorgeben von übergeordneten Zusammenhängen. Die Relevanz von einzelnen Bildern wird durch das System anhand der Nutzeraktionen bestimmt.

Der Nutzer ordnet nur auf der Bedienoberfläche Bilder nach seinem persönlichen Ähnlichkeitsempfinden an. Dabei wird generell auf einer 2- oder 3-dimensionalen Darstellung der Ergebnismenge gearbeitet. Diese Darstellung ist besonders geeignet um manuell Ähnlichkeit in Form von Distanzen zwischen den Bildern auszudrücken. Auf das Kontext-Feedback wird in Abschnitt 4.7 näher eingegangen.

4.2 Ausführung der Iteration im System

Vergleicht man die Möglichkeiten der Iterationsstrategien (Browsen, manuelle Anfrageformulierung, Relevanz-Feedback, Kontext-Feedback), ergeben sich unterschiedliche Methoden der Ausführung. Bei der Navigation erfolgen nach der initialen Anfrage keine weiteren Anfragen an das System. Es werden auch keine Änderungen an Wichtungen oder Merkmalszusammenstellung durchgeführt.

Durch welche Merkmale, Inhaltsobjekte oder andere Beschreibungen die Bilder repräsentiert werden, wird in der initialen Anfrage festgelegt und vom System für die Präsentation aufbereitet. Damit erfolgt eine einmalige Anordnung der Bilder zueinander, bezogen auf die Ähnlichkeit zur Anfrage. Dem Nutzer wird ein Teilausschnitt der Datenbank um den Anfragepunkt präsentiert. Durch Navigieren und Browsen kann der Nutzer verdeckte oder nicht angezeigte Bilder sichtbar machen. Dabei wird der Ausschnitt durch verschieben oder skalieren angepasst und der neue Ausschnitt vom System für die Präsentation berechnet.

Bei einer manuellen Veränderung der initialen Anfrage sind die einzelnen iterativen Anfragen voneinander unabhängig. Jede Anfrage wird wie eine neue Anfrage behandelt und ausgeführt. Wichtungen müssen direkt auf Elemente der Anfrage bezogen werden und in ihrer Größe definiert sein.

Im Relevanz-Feedback werden Wichtungen während der Iteration vom System berechnet und intern in eine neue Anfrage integriert. Bei der Bewertung von Bild oder Bildinhalten durch den Nutzer geht es in der Regel nicht um genaue Werte für die Wichtung, sondern nur um Größenordnungen. Ein typisches Relevanz-Feedback-System ist CHARIOT, das in Abschnitt 5.2 vorgestellt wird.

Je nach Relevanz-Feedback-Methode werden vorhergehende Anfragen in die Neuberechnung mit einbezogen oder auch nicht. Wird jeweils die letzte Anfrage in die Neuberechnung mit einbezogen, erfolgt indirekt das Einbeziehen aller vorherigen Anfragen. Im Laufe der iterativen Suche und der damit verbundenen ständig neuen Bewertung der jeweiligen Zwischenergebnisse, können vorhergehende Bewertungen an Bedeutung verlieren. In einigen Systemen werden aber auch vereinzelt ältere Bewertungen direkt mit in die neue Anfrageformulierung einbezogen (wie z.B. in [SMMP00]), da einige Reformulierungs-Methoden bei einer höheren Anzahl an Bewertungen bessere Ergebnisse erzielen [Bos00]. Dabei kann danach unterschieden werden, ob z.B. nur die positiv oder nur die negativ bewerteten Bilder mit in die Berechnung einbezogen werden oder alle Bewertungen.

Das System CHARIOT [Bos00] merkt sich nur negative Bewertungen, um nach jeder Berechnung der Ergebnismenge die einmal als nicht relevant markierten Bilder zu entfernen. Werden eine oder mehrere vorhergehende Anfragen einbezogen, kann eine kontinuierliche Veränderung der Wichtung einzelner Größen erfolgen. Durch Anpassung des jeweils aktuellen Ergebnisses an die Vorstellung einer gewünschten Ergebnismenge sinkt der Einfluss einer einmal gemachten Bewertung mit der Anzahl der dazwischen liegenden Iterationsschritte.

Beim Kontext-Feedback ist in der Regel das Relevanz-Feedback integriert. Dabei werden Bilder nicht nur verschoben, sondern auch gleichzeitig bewertet. Während sich beim Relevanz-Feedback die Bewertung auf die Wichtung von Elementen bezieht, soll eine vom Nutzer vorgegebene Distanz beim Kontext-Feedback vom System nachmodelliert werden. Damit gibt der Nutzer eine Ähnlichkeit zwischen Bildern vor die das System auf andere Bilder übertragen soll.

Aus den Distanzen müssen rückwirkend Wichtungen ermittelt werden, die in einer neuen Anfrage einen Anfrageraum aufspannen, in dem die Distanzen möglichst den Vorgaben entsprechen. Mit den ermittelten Wichtungen, wird eine neue Anfrage ausgeführt, die neben den angepassten Distanzen in der Regel auch neue Bilder ermittelt.

4.3 Verändern der Wichtung während der Iteration

Die manuelle oder auch die durch das System berechnete Veränderung der Wichtung für die drei Bildbeschreibungs-Ebenen (Abschnitt 2.4.3), Merkmal (M), Merkmalsrepräsentation (MR) und Merkmalsausprägung (MA), bewirkt eine Verschiebung des Anfragepunktes im Vektorraum. Neben der Wichtung der Bildbeschreibungs-Ebenen, gibt es noch weitere Möglichkeiten, Einflussgrößen über Wichtung zu werten. Unter anderem

gibt es die Wichtung einzelner Dimensionen im Vektorraum und die Wichtung der Bilder in einer Anfrage, die aus mehreren Beispielbildern besteht oder aber auch die Wichtung der Inhaltsobjekte. Bei den Wichtungsverfahren kann noch unterschieden werden, ob die Veränderung der Wichtung nur auf die Reformulierung der Anfrage Einfluss hat oder ob die Wichtungen der Bildbeschreibungen in der Datenbank verändert werden.

4.3.1 Wichtungsänderung anhand Nutzerverhalten

Ein Beispiel für die automatische Veränderung der Wichtung der MA, ist das Einbeziehen des Nutzerverhaltens in die Wichtungsberechnung.

Im Relevanz-Feedback werden in der Regel die Wichtung der MA in jeder Anfrage direkt an die Bewertung des Nutzers angepasst. Im Gegensatz dazu, wird beim Einbeziehen des Nutzerverhaltens nicht die neue Anfrage gewichtet, sondern die Bilder der Datenbank [MMS⁺00a]. Dazu wird die Nutzerbewertung der Bilder in der Ergebnismenge auf die Wichtung der MA abgebildet.

Im Laufe von iterativen Anfrageprozessen wird die Summe der positiven und negativen Bewertungen für jedes MA gebildet und in ihre Wichtung einbezogen. Dazu wird jede Anfrage in Bildpaare zerlegt. Bei einer Anfrage mit n Bildern ergeben sich $\frac{n(n-1)}{2}$ Bildpaare. Die Idee ist, dass Bilder die gemeinsam positiv bewertet wurden (positives Bildpaar) wichtige MA gemeinsam haben müssen. Bildpaare hingegen, bei denen ein Bild positiv und das andere negativ bewertet wurde, können nur MA gemeinsam sein, die keine Bedeutung haben. Gemeinsame MA in einem positiven Bildpaar erhalten eine positive Markierung während gemeinsame MA in einem negativen Bildpaar eine negative Markierung erhalten. Die Summe der Markierungen wird normalisiert und in einen Faktor F_{ijk} für die Wichtung W_{ijk} jeder MA umgerechnet.

Bei MA, die nur negative Markierungen erhalten, wird dieser Faktor Null, mit der Folge, dass die MA nicht mehr berücksichtigt wird. Erhält eine MA nur positive Markierungen, verdoppelt sich die Wichtung. Der Faktor kann datenbankabhängig, nutzerabhängig oder sitzungsabhängig eingesetzt werden. Beim datenbankabhängigen Einsatz wird der Faktor kontinuierlich bei jeder Iteration auf den Daten der Datenbank angepasst. Bei der Nutzerabhängigkeit werden für jeden Nutzer eigene Faktoren abgespeichert, während Faktoren, die sich auf eine einzelne Sitzung beziehen, nach jeder abgeschlossenen Sitzung gelöscht werden.

4.3.2 Dimensionswichtung

Bei der Dimensionswichtung gibt der Nutzer an, welche MA ihn besonders interessieren. Die Angabe verändert die Skalierung der Dimension. Der Raum wird dadurch „gekrümmt“. In Abbildung 4.1 ist der ursprüngliche Anfrageraum als Kreis abgebildet. Durch höheres wichten der Dimension x und verringern der Wichtung von Dimension y wird der Anfrageraum zu einer Ellipse. Bilder mit höheren und niedrigeren Werten für x , werden zusätzlich in die Ergebnismenge aufgenommen, während Bilder mit der MA

y nur noch eingeschränkt aufgenommen werden bzw. in der Ergebnismenge nicht mehr vorkommen.

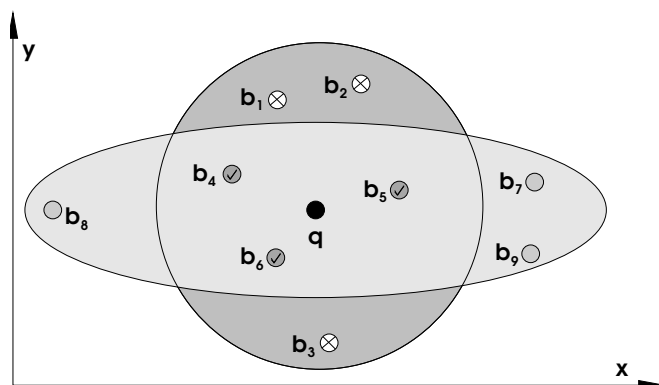


Abbildung 4.1: Beispiel Dimensionswichtung (Quelle [Bos00])

Die Dimensionswichtung ist eine gängige Methode im Relevanz-Feedback und wird an Hand dem Beispiel in Abbildung 4.1 in 4.6.3 genauer beschrieben.

4.4 Auswahl der Merkmale für eine neue Anfrage

In einer initialen Anfrage Q_{ini} , in der die Merkmalsausprägungen (MA) manuell angegeben werden, ist Q_{ini} ein dünn besetzter Vektor, bei dem bis auf die angegebenen MA alle Dimensionen den Wert Null haben. Im Gegensatz dazu stehen die dicht besetzten Vektoren der Bilder. Auch wenn die vorhandenen MA exakt bestimmt wurden, ist nach der Clustertheorie [CMRB01] die Ähnlichkeit der für die Suche relevanten Bilder untereinander höher ist als die Ähnlichkeit der relevanten Bilder zum Anfrage-Vektor. Durch einbeziehen von MA aus der Ergebnismenge von Q_{ini} wird aus dem dünn besetzten Anfrage-Vektor ein neuer, dicht besetzter Anfrage-Vektor Q_{neu} berechnet. Dabei besteht die Möglichkeit, die Bilder, aus denen die Merkmale bestimmt werden sollen, vom Nutzer wählen zu lassen (siehe Relevanz-Feedback 4.6) oder andernfalls die Auswahl dem System zu überlassen. Bei der Bildauswahl durch das System spricht man von einem Pseudo-Feedback, bei dem eine vorgegebene Anzahl der ersten zurückgelieferten Bilder vom System automatisch als relevant angesehen wird. Nach der ersten Erweiterung des Anfrage-Vektors, werden in weiteren Schritten die ausgewählten MA angepasst. Durch die jeweils neu hinzukommenden relevanten Bilder können neue relevante Merkmale hinzukommen, vorhandene sich in der Wichtung ändern oder sogar irrelevant werden.

Die einzelnen Schritte der Erweiterung sind nach [CMRB01]: 1. Suche der MA für die Erweiterung, 2. Bewerten der MA, 3. neue Anfrage berechnen. Die drei Schritte können in einer einzigen Berechnungsmethode zusammengefasst werden. Beim „einfachen“ Relevanz-Feedback ist eine klassische Methode die Berechnung durch die Formel von Rocchio [Bos00]. In Experimenten wird jedoch versucht, die Erweiterung durch Einsatz unterschiedlicher Algorithmen für die einzelnen Schritte zu optimieren.

Eine einfache Methode der MA-Auswahl ist es, alle MA aus den ersten Bildern zu verwenden. Dagegen spricht der erhöhte Rechenaufwand und nach [CMRB01] ist damit die Verbesserung gegenüber Methoden mit begrenzter MA-Anzahl nur geringfügig. Eine Aufwändigere Methode ist beispielsweise, Ergebnisse ähnlicher vergangener Anfragen auszuwerten oder MA über verteilungsbasierte Analysen zu ermitteln, wie es auch bei der *inverse document frequency* (siehe Formel 3.1) eingesetzt wird.

Eine MA ist ideal für die Erweiterung, wenn sie in einem relevanten Bild häufig vorkommt und bezogen auf den gesamten Datenbankbestand eher selten ist (ähnlich der Selektivität im RDBS). Ziel ist es, MA zu finden, die den semantischen Unterschied zwischen den gesuchten Bildern und den übrigen Bildern beschreiben.

Für die Auswahl der MA, werden alle MA der relevanten Bilder anhand ihrer Wichtung, die auch neu berechnet sein kann, sortiert. Während die Wichtungsverfahren aus dem Text-Retrieval dann einen hohen Wert ergeben, wenn der gesuchte Term besonders häufig im relevanten Dokument vorkommt, muss die Häufigkeit einer MA eines relevanten Bildes möglichst dicht an der bei der Anfrage vorgegebenen Häufigkeit liegen. Ein Beispiel dafür ist, wenn ein Bild mit blauem Himmel gesucht wird und dazu die Angabe „20% blau“ in der initialen Anfrage gemacht wurde. Bilder mit einem höheren Blauanteil könnten z.B. Unterwasser-Aufnahmen sein die dann nicht relevant wären.

Einer der Hauptgründe, den ersten Schritt gesondert zu berechnen, ist mit einer schnellen Berechnungsmethode aus allen MA der dicht besetzten Vektoren eine kleine Untermenge an MA zu extrahieren. Die extrahierten MA können dann in weiteren Schritten mit genaueren, berechnungsintensiveren Methoden für eine neue Anfrage sortiert und gewichtet werden.

Verfahren aus dem Text-Retrieval zur Sortierung der MA benutzen unter anderem eine Formel von Rocchio oder eine χ^2 Variante von Doszkoc [CMRB01]. Bei Rocchio wird der Wert $score(m_{ijk})$ für die Sortierungsposition einer MA m_{ijk} erhalten, indem die Summe der Wichtungen W_{ijk} von m_{ijk} bei R ausgewählten Bildern gebildet wird:

$$score(m_{ijk}) = \sum_{r=1}^R W_{ijk}(r) \quad (4.1)$$

Das Verfahren von Doszkoc arbeitet mit der Wahrscheinlichkeitsverteilung der MA. Es wird dabei berücksichtigt, wie oft die MA in der Datenbank und wie oft in den relevanten Bildern vorkommt.

$$score(m_{ijk}) = \frac{p_R(m_{ijk}) - p_C(m_{ijk})}{p_C(m_{ijk})} \quad (4.2)$$

Mit der Wahrscheinlichkeit $p_C(m_{ijk})$, dass die MA in der Datenbank vorkommt und der Wahrscheinlichkeit $p_R(m_{ijk})$, dass MA in der Menge der relevanten Bilder vorkommt. Ist das Vorkommen von m_{ijk} in den relevanten Bildern groß und in der Datenbank klein, erhält m_{ijk} ein hohen Sortierungswert.

Die MA werden dann nach ihrem Sortierwert in einer Liste angeordnet und eine vorgegebene Anzahl der ersten MA für eine neue Anfrage ausgewählt und neu gewichtet. Eine Wichtungsmethode dazu ist die Kullback-Leibler-Divergenzmessung KLD , die in [CMRB01] vorgestellt wird:

$$KLD_{p_R, p_C} = \sum_k \left\{ p_R(m_{ijk}) \cdot \log \frac{p_R(m_{ijk})}{p_C(m_{ijk})} \right\} \quad (4.3)$$

KLD ist die relative Entropie zwischen den Wahrscheinlichkeiten p_R , dass die MA in der Menge der relevanten Bilder und der Wahrscheinlichkeiten, dass die MA in der Menge der Bilder der Datenbank p_C . Eine hohe Entropie drückt den Aufwand aus, der bei einer Kodierung von p_R durch p_C (oder umgekehrt) benötigt wird. Das heißt die MA m_{ijk} die am meisten zu einer hohen Entropie beitragen, haben einen hohen Informationsgehalt und tragen somit am meisten zur Unterscheidung der Bilder bei. Diese MA werden am höchsten gewichtet.

Die neu ermittelten Wichtungen werden dann zur Berechnung der neuen Anfrage eingesetzt. Eine Formel zur Anfrageneuberechnung ist z.B. Rocchio (siehe Abschnitt 4.6).

4.5 Browsen und Navigieren

Der Startpunkt beim Browsen oder Navigieren kann durch eine Initiale Anfrage oder zufällig gewählt werden. Während der Suche erfolgt keine Änderung der initialen Anfrage mehr, anstatt Bilder eines Anfrageraums in der Datenbank betrachtet der Nutzer Bilder aus Datenbankausschnitten die er durch Navigation erreicht. Der Nutzer ändert nur die Position im Vektorraum der Datenbank anhand der präsentierten Bilder auf der Nutzeroberfläche. Die Navigation kann auf Zugriffsstrukturen oder auf präsentierten Ausschnitten aus dem Datenbankinhalt erfolgen. Eingesetzte Zugriffsstrukturen sind vor allem Baumstrukturen. Bei der Präsentation aller enthaltenen Bilder stehen für die Navigation in der Regel Listen oder mehr-dimensionale Darstellungen zur Verfügung.

Beim Einsatz von Bäumen kann beginnend am Wurzelknoten, der Baum durchlaufen werden. Bei der jeweiligen Präsentation der Knoten der nächst tieferen Ebene, kann der Nutzer nach jedem Schritt entscheiden, in welche Richtung der Baum durchlaufen werden soll. Dabei repräsentieren die Knoten jeweils einen Cluster von Bildern, in dem die Bilder einer bestimmten Ähnlichkeitsbedingung genügen. Der Cluster wird dem Nutzer anhand eines Bildes präsentiert, das dem Cluster-Mittelpunkt am nächsten liegt und somit als repräsentativ gilt.

In Abbildung 4.2 sind zwei Fensterausschnitte der Benutzeroberfläche gezeigt. Der Nutzer befindet sich (im oberen Fensterausschnitt) am Wurzelknoten einer Baumstruktur, welche die Bilder b_1 bis b_{10} Clustert. Dem Nutzer werden zwei Beispielbilder für die Cluster AB und CD präsentiert. Wählt er das Cluster CD werden ihm die Beispielbilder für die Cluster C und D gezeigt.

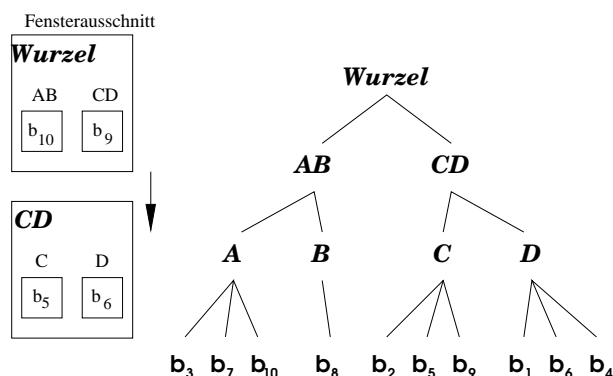


Abbildung 4.2: Browsen auf einer Baumstruktur mit Cluster

Wird nach einer initialen Anfrage, der Datenbankinhalt als eine sortierte Liste ausgegeben, kann der Nutzer diese Liste beliebig weit durchlaufen. Beim Browsen auf dem 2- oder 3-dimensional visualisierten Datenbankinhalt stehen mehrere Optionen zur Verfügung. Durch Vergrößern (zoomen) eines Fensterbereichs der Benutzerschnittstelle wird ein Datenbankschnitt detaillierter angezeigt, es sind dann in der Regel mehr Bilder aus Datenbank angezeigt. Es kann aber auch aus dem sichtbaren Bereich „herausgezoomt“ werden, für einen weiteren Überblick über den Datenbankinhalt. Zur besseren Übersicht werden dann Bilder aus dem ursprünglichen Fensterbereich entfernt und neue aus dem vergrößerten Datenbankschnitt hinzugefügt. Mit einer Pan-Funktion kann der sichtbare Ausschnitt verschoben werden, um Bilder außerhalb des Präsentationsausschnittes sichtbar zu machen. Bei einer sehr dichten Präsentation, in der sich Bilder überlappen, gibt es Funktionen wie *Fish-eye*, mit der lokal Ausschnitte vergrößert werden können.

4.6 Relevanz Feedback

Beim Relevanz Feedback können nach jeder Anfrage die Bilder des Anfrageergebnisses bewertet werden. Die Bewertung wird an das System zurückgegeben und zur Generierung einer neuen Anfrage eingesetzt. Im Prozess des Relevanz-Feedbacks werden durch die Kombination von Nutzerbewertung und automatischer Merkmalsbestimmung durch das System aus groben, undetaillierten Anfragen, verfeinerte und detaillierte Anfragen [RHM98].

In zwei Punkten können die Eigenschaften des Relevanz-Feedbacks zusammengefasst werden [SB90]:

- Es schirmt den Benutzer von den Details des Anfrageformulierungsprozesses ab und erlaubt die Konstruktion von nützlichen Suchanweisungen, ohne mit der Zusammenstellung der Bilder in der Datenbank oder der Suchumgebung vertraut sein zu müssen.

- Es unterstützt einen kontrollierten Änderungsprozess, in dem einzelne Merkmale durch Wichtung betont oder abgeschwächt werden.

Aus einfachen, initialen Beispielbild- oder Merkmals-Anfragen entstehen durch Einbinden von weiteren, auch wahrnehmungsfremden, Merkmalen und durch berücksichtigen von der Verteilung der Merkmalsausprägungen in Bild und Datenbank hochdimensionale gewichtete Bildbeschreibungen.

Im Folgenden werden Bewertungsverfahren durch den Nutzer und die Abbildung der Nutzerbewertung im System vorgestellt.

4.6.1 Bewertung durch den Nutzer

Die Bewertung des Nutzers geht bei der Berechnung der neuen Anfrage in die Wichtungen der unterschiedlichen Bildbeschreibungsebenen ein [Bos00]:

Gesamtbildbewertung: Der Nutzer beurteilt einzelne Bilder aufgrund des Gesamteindrucks vom Bild.

Merkmalsbewertung: Der Nutzer bewertet die einzelnen Merkmale des jeweiligen Bildes, z.B. Textur und Farbe.

Bewertung der Merkmalsausprägungen: Der Nutzer bewertet die einzelnen Merkmalsausprägungen jedes Bildes, z.B. den Grünanteil des Merkmals Farbe.

Durch Abstufen der Bewertung können Bildinhalte oder Bilder unterschiedlich stark gewichtet werden. Bei einer nicht abgestuften Bewertung wird ein relevantes Bild oder Bildinhalte einfach markiert. Bei einer abgestuften Bewertung wird ein Bild oder die Bildinhalte anhand einer zugehörigen Skala bewertet. Dazu gibt es die binären und die gestuften Prädikate.

Binäres Prädikat: Je nachdem, welche Möglichkeit das System unterstützt, kann der Nutzer angeben, ob ein Bild relevant ist oder ob ein Bild irrelevant ist.

Gestufte Prädikate: Der Nutzer kann unter mehreren Abstufungen wählen, z.B. sehr relevant, relevant, neutral, irrelevant, sehr irrelevant. Mehr als fünf Stufen sollten jedoch nicht vorgegeben werden. In Experimenten haben die Autoren von [RHM98] festgestellt, dass zwar höhere Genauigkeit bei mehr Stufen zu erwarten ist, aber bei steigender Abstufungszahl die Bedienfreundlichkeit sinkt und es für den Nutzer immer schwerer wird die richtige Bewertung für ein gegebenes Bild zu finden.

4.6.2 Systemseitige Verarbeitung

Die grafische Bewertung durch den Nutzer muss in einer Form abgebildet werden, die in die Berechnung einer neuen Anfrage einbezogen werden kann. Bezüglich der Ähnlichkeit bzw. Distanz heißt das, Merkmalsausprägungen, Merkmale oder ganze Bilder die

relevant(er) sind, müssen durch einen Faktor so verändert werden, dass der neue Anfrageraum hauptsächlich die Bilder enthält die als relevant markiert wurden, und solche Bilder nicht enthält die als nicht relevant markiert wurden, bzw. nur Bilder angezeigt werden, welche die relevanten Merkmale und Merkmalsausprägungen enthalten.

Bezogen auf Anfragevektor und Anfrageraum kann durch die Veränderung eine

- Verschiebung des Vektors (z.B. Verfahren nach Rocchio) erfolgen oder
- der Anfrageraum bezüglich Ausmaße und Form verändert werden (z.B. Dimensionswichtung).

Eine einfache Umsetzung der Bewertung ist es, positive Werte für relevantere Eigenschaften und steigend negative für irrelevante Eigenschaften zu wählen.

Abhängig vom System können Bilder, Merkmale und oder Merkmalsausprägungen bewertet werden. Dabei erfolgen durch eine Bewertung von Merkmalsausprägungen wesentlich detailliertere und direktere Vorgaben, als wenn nur Merkmale oder, im noch indirekteren Fall, ganze Bilder bewertet werden. Eine indirekte Bewertung (Bild oder Merkmal) kann auf alle direkteren Bewertungen (Merkmal oder Merkmalsausprägung) übertragen werden. Wenn in Gegenrichtung, eine direkte Bewertung auf eine indirektere übertragen werden soll muss eine Bewertung für das Gesamtbild oder das zugehörige Merkmal berechnet werden, die eine durchschnittliche Bewertung darstellt.

Wird z.B. die Farbe Blau sehr hoch gewertet, erhält das Merkmal Farbe eine höhere Bewertung als das Merkmal Textur und ein Bild mit sehr viel Blau (beispielsweise Himmel mit Meer) wird höher gewertet als ein Bild mit wenig Blau (beispielsweise Wald mit Himmel).

Werden die Merkmale in Abhängigkeit der Bilder bewertet kann eine Eintragung im System erfolgen wie es in Tabelle 4.1 dargestellt ist. Dabei sind fünf Bewertungsstufen

Wichtung	Vektor ₁	Vektor ₂	Vektor ₃
Objekt₁	1	1	1
Objekt₂	3	0	0
Objekt₃	-1	0	1

Tabelle 4.1: Interne Darstellung der Nutzerbewertung [Bos00]

von „sehr relevant“ über „neutral“ bis zu „sehr irrelevant“ auf die Werte [3, 1, 0, -1, -3] abgebildet. Diese Werte wurden in Experimenten als geeignete Zuordnung zu den entsprechenden Begriffen gefunden, da sie dem menschliche Empfinden der Begriffe am nächsten kommen [RHM98].

Sind alle Merkmale eines Bildes als gleich relevant gekennzeichnet erhält das Gesamtbild die gleiche Bewertung wie seine Merkmale, was der ersten Zeile in Tabelle 4.1 entspricht.

4.6.3 Reformulierungsmethoden

Im Text-Retrieval gibt es mehrere Reformulierungsmethoden, angepasst auf das jeweilige Retrieval-Verfahren. In [SB90] wird eine Methode von Rocchio als klassisches Relevanz-Feedback-Verfahren und ein probabilistisches Verfahren vorgestellt. Die beiden Verfahren bewirken eine Verschiebung des Anfrage-Punktes im Merkmal-Raum. Mit der Bewertung durch den Nutzer kann auch eine Dimensionswichtung erfolgen die eine Krümmung des Merkmal-Raumes verursacht (siehe Abschnitt 4.3.2). Daneben gibt es noch eine, durch einbeziehen der Nutzerbewertung R , erweiterte Form der *inverse document frequency* (siehe Formel 3.1) für die Berechnung der Relevanz einer MA. Neben der Häufigkeit des MA in Bild n mf_{ijkn} (kurz mf_{kn}) und in der Datenbank $\frac{M}{df_{ijk}}$ (kurz $\frac{M}{df_k}$) wird auch die Anzahl der Bilder N in der Anfrage berücksichtigt.

$$idf_k = \frac{1}{N} \sum_{n=1}^N (mf_{kn} \cdot R_n) \cdot \log_2\left(\frac{M}{df_k}\right) \quad (4.4)$$

In [MMS⁺00b] liegt die Relevanz R_n im Bereich $[-1;1]$.

Rocchio

Rocchios Formel zur Anfragepunkt-Verschiebung ist im Text-Retrieval ein Standard Algorithmus der oft als Ausgangspunkt für weiterentwickelte Algorithmen genutzt wird oder zu Vergleichszwecken mit anderen Algorithmen herangezogen wird ([Bos00, CMRB01]).

Der Algorithmus nützt positives und negatives Relevanz Feedback um den Anfragepunkt von den irrelevanten Bildern zu entfernen und zu den relevanten Bildern hin zu bewegen. Bei jedem Iterationsschritt können neue Bilder aus einer neuen "nächsten Nachbarschaft" als relevant oder irrelevant aufgenommen werden. diese Formel wird üblicher Weise zum sortieren und Wichten der Anfrage-MA benutzt.

$$Q_{neu} = Q_{orig} + \frac{\beta}{|R|} \sum_{r \in R} r - \frac{\gamma}{|R'|} \sum_{r' \in R'} r' \quad (4.5)$$

Q_{neu} ist der gewichtete Merkmal-Vektor der reformulierten Anfrage. Q_{orig} ist der gewichtete Merkmal-Vektor für die originale unexpandierte Anfrage R und R' sind die Mengen der relevanten und irrelevanten Bilder. r und r' sind die Vektoren aus R und R' extrahiert.

Bei einer Menge ausschließlich relevanter Objekte reduziert sich 4.6.3 mit R als die ersten als relevant eingeordneten Objekten zu :

$$Q_{neu} = Q_{orig} + \frac{\beta}{|R|} \sum_{r \in R} r \quad (4.6)$$

Dimensionswichtung

Bei der Dimensionswichtung im Relevanz-Feedback wird für jede Dimension der Merkmalsvektoren eine neue Wichtung berechnet. Die Wichtungen verändern die Ausdehnung des Anfrageraums. Innerhalb einer Dimensionen verändern sich die Distanzen umgekehrt proportional zur Dimensionswichtung d.h. der Raum wird verzerrt. Bei neuer Berechnung des Anfrageraums um die Anfrage können im angepaßten Raum neue Bilder gefunden werden. Die Wichtungsanpassung erfolgt mittels der Standardabweichung der relevanten Bilder von der Anfrage was auch als *Sigmamessung* bezeichnet wird.

Die Bewertung die vom Nutzer in Abhängigkeit der Bilder gemacht wird geht in die Berechnung der Dimensionswichtung ein.

In Abbildung 4.1 ist eine initiale Anfrage q in zwei Dimensionen x und y dargestellt, wobei der Nächste-Nachbarn-Suchraum im Zweidimensionalen ein Kreis ist. Bei der ersten Berechnung werden die Bilder b_1, \dots, b_6 als die nächsten Nachbarn gefunden. Davon sind die Bilder b_1, b_2 und b_3 irrelevant (von Nutzer markiert). Die Standardabweichung σ der Objekte b_4, b_5 und b_6 in Richtung Dimension x ist größer als in Richtung y , $\sigma(x) \geq \sigma(y)$.

$$\sum_{i=4}^6 (x_i - x_q)^2 \geq \sum_{i=4}^6 (y_i - y_q)^2 \quad (4.7)$$

Mit $w[x,] = \left[\frac{1}{\sigma(x)}, \frac{1}{\sigma(y)} \right]$ ist die umgekehrte Proportionalität gegeben. Die Summe der Wichtungen w_i wird zu 1 normalisiert ($\sum_{i=1}^{dim} w[i] = 1$). Der Suchraum der verfeinerten Anfrage wird bei gleichbleibender Skaleneinteilung zu einer Ellipse.

Damit die Dimensionswichtung überhaupt eingesetzt werden kann müssen mindestens zwei Objekte Relevant sein. Je Mehr Objekte relevant sind desto höher die Qualität der Wichtung, da Ausreißer an Bedeutung verlieren.

4.7 Kontext-Feedback

Für das Kontext-Feedback ist eine Visualisierung des Datenbankinhalts bzw. des Anfrageraums notwendig. Anstatt Bilder nur als relevant oder nicht relevant zu markieren, werden die Bilder des Anfrageraums vom Nutzer auf der Bildschirmfläche relativ zueinander angeordnet. Ähnlich empfundene Bilder werden nahe beieinander angeordnet. Im Gegenzug dazu werden Bilder mit geringer die Ähnlichkeit weiter voneinander entfernt positioniert. Bei einer Suche nach Haustieren können jeweils die Katzen und die Hunde nahe beieinander angeordnet werden während darauf geachtet wird, dass die Hunde- und Katzen-Ansammlung getrennt voneinander erscheinen. Das Zusammenfassen zeigt eine höhere Ähnlichkeit der Bilder an. Anstatt dass der Nutzer versucht, die Wirkungsweise der Ähnlichkeitsmessung zu begreifen, gibt er eine gewünschte Ähnlichkeit vor, die dann vom System nachgebildet wird.

Die Reformulierung der Anfrage wird dadurch bestimmt welche Information aus der Benutzeroberfläche gewonnen wird. Aus der Positionierung der Bilder und den Ein-

stellungen zur Präsentation werden die benötigten Parameter gewonnen um die Ähnlichkeitsmessung anzupassen. Einstellungen sind Zoom-Faktor, Präsentationsachsen und Bildverschiebung.

In Abbildung 4.3 ist in Bild a) die Ergebnismenge nach einer initialen Abfrage zu sehen. In Bild b) und c) wird angezeigt wie der Nutzer die Bilder zueinander positioniert und in Bild d) ist das, anhand von c), neuberechnete Ergebnis des Systems abgebildet.

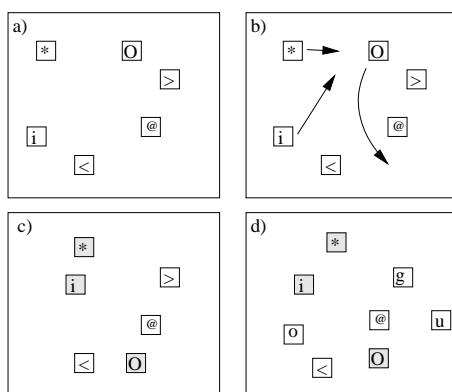


Abbildung 4.3: Anordnen der Bilder im Kontext-Feedback

Die gewählten Bilder können als Anker betrachtet werden, die einen neuen Sub-Raum aufspannen und gleichzeitig Beispielbilder für die neue Anfrage sind. Anhand der Anker wird eine neue Ähnlichkeitsmessung ermittelt und die Distanzen neu berechnet. Der letzte Schritt ist die Präsentation der neuen Berechnung auf der Nutzeroberfläche. Dabei können neue Bilder hinzukommen und alte verschwinden [SJ00].

Um durch das Verschieben der Bilder nicht nur die Wichtung zu beeinflussen, sondern auch die Position der Anfrage im Datenbankraum, muss ein Referenzpunkt für die neue Anfrage gesetzt werden können. Wird ein Bild als Referenzpunkt gewählt, ist es der Ursprung der neuen Anfrage. Das heißt, wird ein Referenzbild gewählt, wird der Raum so transformiert, dass das referenzierte Bild in der Mitte der Bildanordnung platziert wird. Das Referenzbild muss nicht zwingend ein Teil der Anfrage sein. Die Koordinaten des Referenzbildes geben gleichzeitig eine Referenzdistanz an. Alle als relevant markierten Bilder müssen so dicht an der Mitte liegen, dass sie für den Nutzer auf der Bildschirmoberfläche sichtbar werden.

Die vorgegebene Bildanordnung auf der Benutzeroberfläche und die Ähnlichkeitsberechnung werden in einem Optimierungsprozess aufeinander abgestimmt. Je nach System werden dazu einzelne Gewichte geändert, andere Merkmale gewählt oder auch unterschiedliche Metriken eingesetzt. Jedes Mal mit dem Ziel, die Berechnung so zu ändern, dass die daraus hervorgehende neue Ergebnispräsentation der vorgegebene Bildanordnung auf der Benutzeroberfläche nahe kommt. In [SJ00] wird dazu ein Optimierungsprozess durchgeführt der als Eingangsgröße die Matrix der Distanzen aller markierten Bildpaare erhält. Die veränderbaren Parameter der Ähnlichkeitsberechnung sind in einem Vektor abgelegt der schrittweise und so lange verändert wird bis ein vorgegebenes

Maß an Übereinstimmung mit der Vorgabe erreicht wird oder ein Zeitlimit überschritten wird. Der Nutzer hat keinen direkten Einfluss auf die Faktoren der Ähnlichkeitsberechnung.

Das Verändern der Bildanordnung kann durch Drag-and-Drop realisiert werden. Eine Erweiterung dieser Methode ist das Vereinen einzelner Bilder unter einem Überbegriff bzw einem Referenzbild, so dass nur das Referenzbild noch bewegt werden muss. Dazu wird ein Rahmen (eine sogenannte *Konzept-Box*) aufgezogen in den die Bilder abgelegt werden die zu einer Gruppe (Cluster) gehören sollen. Die Konzept-Box wird dann eingesetzt wie ein unabhängiges Bild auch wenn es eigentlich eine Gruppe von Bildern ist [SJ00]. Die Konzept-Boxen können als *visuellen Konzepte* abgespeichert werden. Je nach Anwendungsfall können die visuellen Konzepte, auf eine Session begrenzt sein, an einen Benutzer gebunden oder für alle Nutzer zugreifbar sein. In Abbildung 4.4 ist das Prinzip der Konzept-Box dargestellt.

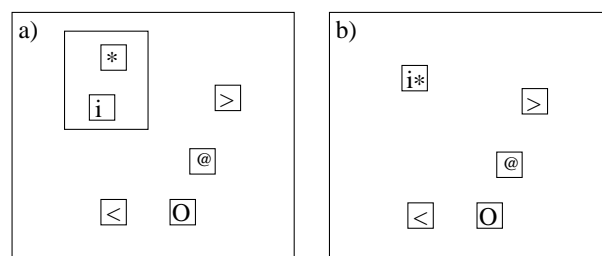


Abbildung 4.4: Prinzip der Konzept-Box

Kapitel 5

Anfragesysteme

Ziel dieses Kapitels ist es, eine für die Arbeit repräsentative Auswahl an graphischen Anfragesystemen vorzustellen. Die Auswahl ist darauf ausgelegt, möglichst viele der gefundenen Ansätze abzudecken. Neben den allgemeinen Eigenschaften werden Besonderheiten der Systeme hervorgehoben, die sie von den anderen unterscheidet. Schwerpunkt dieses Kapitels liegt auf den ersten vier Anfragesprachen. Sie kombinieren gezielt mehrere Ansätze der Iteration, um bei möglichst kurzen Antwortzeiten gute Ergebnisse zu erzielen.

In kürzeren Abschnitten werden weitere Systeme beschrieben, die Ansätze enthalten welche nicht in den ersten Anfragesystemen verwirklicht sind. Von den ersten vier Anfragesystemen wird jeweils (soweit möglich) die darunter liegende Architektur, die Indexierung, die Benutzerschnittstelle und die iterative Anfrageformulierung beschrieben. Die Beschreibungen erheben nicht den Anspruch auf Vollständigkeit. Da die meisten der Anfragesysteme kontinuierlich weiterentwickelt werden, werden immer wieder Ansätze und Ideen getestet, hinzugefügt, verändert oder auch verworfen.

In der Erläuterung der Architektur werden die einzelnen Komponenten der Anfragesysteme und ihr Zusammenwirken beschrieben. Die Beschreibung der Indexierung gibt die Möglichkeiten des Zugriffs auf die Daten an und beschreibt Zugriffsstrukturen der Systeme. Je nach System wird dazu die Zusammenstellung der Merkmale betrachtet. Von den Benutzerschnittstellen wird je ein kurzer Überblick über die visuelle Gestaltung der Oberfläche und die Benutzerführung gegeben. Die iterative Anfrageformulierung wird im letzten Teil des jeweiligen Abschnittes erläutert. Entsprechend dem Ablauf des Anfrageprozesses werden erst die vom Nutzer zu machenden Eingaben beschrieben, gefolgt von der Beschreibung der Verarbeitung und Präsentation durch das System.

5.1 SEMCOG

Das Datenbank-Managementsystem SEMCOG (SEMantics and COGnition-based image retrieval) ist eine Entwicklung der NEC C&C Research Laboratories. Für Anfragen auf

das System wurde die SQL-ähnliche Anfragesprache CSQL (Cognition and Semantic Query Language) entwickelt. Anfragen mit CSQL werden über die visuelle Anfrageschnittstelle IFQ (In Frame Query) erzeugt. Die Beschreibung der Architektur und der Indexierung ist in [LCHH97b] zu finden. In [LCHH97a] erfolgt eine detaillierte Beschreibung der Benutzeroberfläche und ihrer Bedienung.

Eine Besonderheit von SEMCOG ist die Kombination von mehreren Indexierungsarten. Die Suche erfolgt auf relationalen Tabellen, in Ontologien und im Merkmalsraum. Zudem bietet SEMCOG Alternativen zu jeder Anfrage an, die mit Hilfe von Ontologien und Datenbank-Statistiken ermittelt werden. Die Anfragerreformulierung ist manuell. Die Art der Formulierung in der initialen Anfrage unterscheidet sich nicht von der Formulierung der Anfragen in den iterativen Schritten.

5.1.1 Architektur

Die Architektur von SEMCOG ist modular aufgebaut, so dass Komponenten für neue Medienobjekte oder weitere Funktionen integriert werden können. Die drei Hauptebenen: Client, Server und Datenspeicherung, fassen die Grundfunktionalitäten zusammen. Die oberste Ebene enthält die Benutzerschnittstelle mit der visuellen Anfragesprache IFQ, die Funktionen für Anfragespezifikation, Anfragegenerierung und Ergebnispräsentation anbietet. In der zweiten Ebene erfolgt die Weiterverarbeitung der generierten Anfrage. Hier werden aus der Anfrage Unteranfragen erzeugt und an die verarbeitenden Komponenten weitergeleitet. Die Teilergebnisse werden zu einer Ergebnismenge zusammengeführt. Zudem erfolgen Analysen möglicher Alternativen für die einzelnen Anfrageelemente. Die Funktionen der dritten Ebene betreffen die Datenspeicherung. Hier werden die Bilder und die daraus extrahierten oder manuell festgelegten Daten abgelegt.

In [LCHH97b] wird die zweite Ebene in SEMCOG mit fünf Komponenten vorgestellt: *Facilitator*, *Image Data Query Processor*, *Image Semantic Editor*, *Terminology Manager* und *Textual Data Query Processor*.

Zentrale Komponente ist der *Facilitator*, er übernimmt die Koordination und Aufteilung der Anfragen auf die übrigen Komponenten. Er ermittelt zu jeder Anfrage bewertete Alternativen und macht dazu entsprechende Eintragungen in die *Datenbank-Statistiken* (siehe Abbildung 5.3), welche die Bewertung unterstützen.

Der *Image Data Query Processor* führt die Teilanfragen aus, die eine Suche im Merkmalsraum erfordern. Daneben erfolgt hier auch die Merkmalsextraktion, Objekterkennung und das Spezifizieren von räumlichen Beziehungen beim Einlesen neuer Bilder. Dazu besitzt der *Image Data Query Processor* zwei weitere Komponenten, den COIR (Content Oriented Image Retrieval) und den *Image Component Katalog*. COIR führt eine semiautomatische Objekterkennung durch. Erkannte Objekte werden dann mit einer Bezeichnung und repräsentativen Merkmalen im *Image Component Katalog* abgelegt. Räumliche Beziehungen zwischen den erkannten Objekten werden über den *Image Semantics Editor* abgelegt.

Der *Textual Data Query Processor* ist für die Ausführung aller erzeugten Teilanfragen

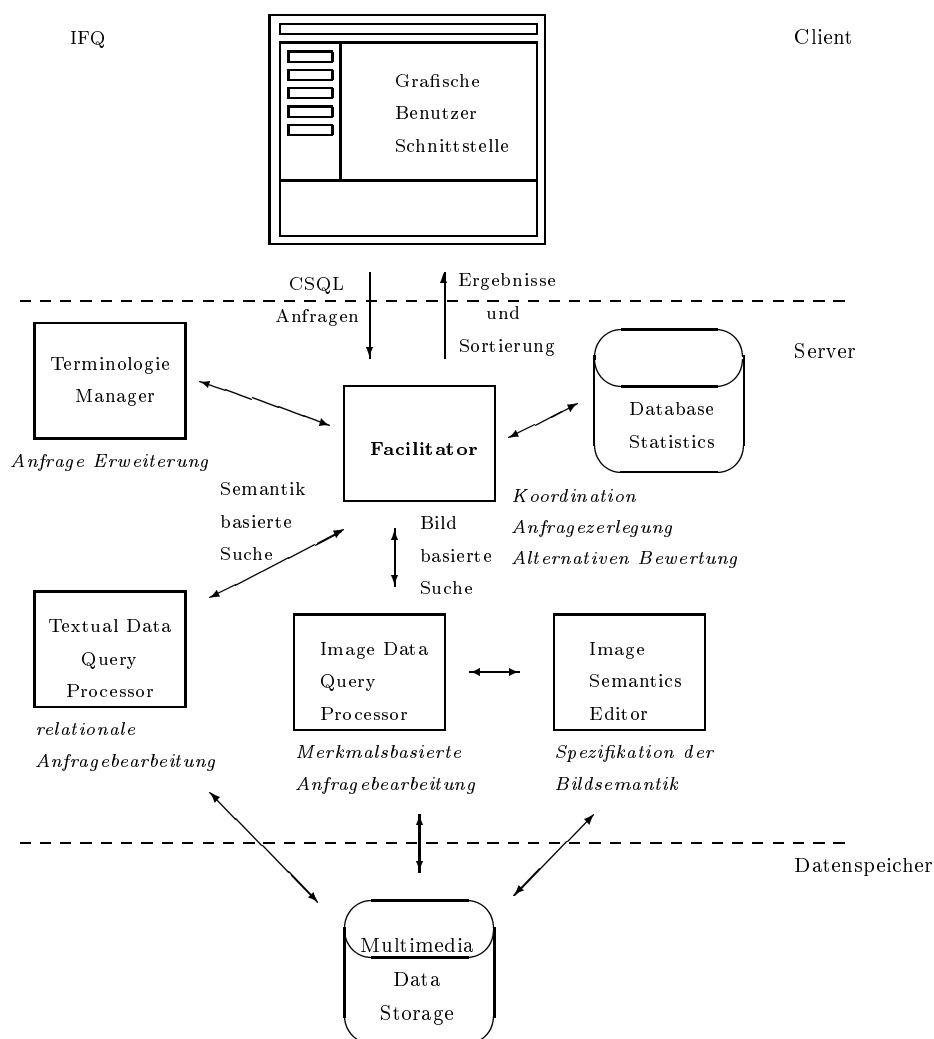


Abbildung 5.1: System Architektur von SEMCOG (angelehnt an [LCHH97b])

zuständig, die textueller Art sind. Die textuellen Teilanfragen werden zuvor vom *Facilitator* mit Hilfe des *Terminology Manager* aufbereitet. Beim Erhalt von Begriffen gibt der *Terminology Manager* Synonyme und ähnliche Begriffe mit Ähnlichkeitsangaben zurück. Die für die Ähnlichkeitsangaben zuständige Teil-Komponente des *Terminology Managers* ist ein lexikalisches Referenzierungs-System, das eine Ontologie zur Verfügung stellt, in der die Ähnlichkeit der Wörter zueinander als Wert abgelegt ist. Der *Textual Data Query Processor* ist auch zuständig für den Aufbau einer Ontologie aus den Objekten, die dem System bekannt gemacht wurden (Skizze der Architektur, siehe Abbildung 5.1).

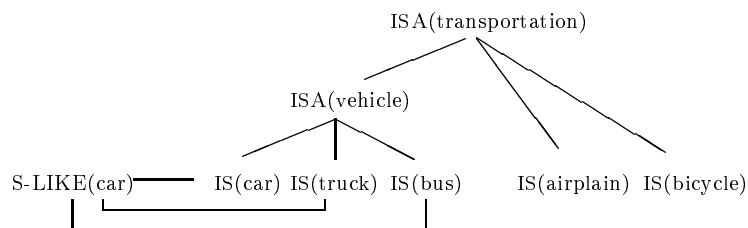


Abbildung 5.2: Begriffshierarchie mit Prädikaten (aus [LCHH97b])

Selektivitäts-Statistik		
Bedingung	Selektivität	
IS(car)	30	+ + +
IS(truck)	15	
IS(bus)	9	
IS(bicycle)	24	+ +
IS(airplane)	45	
ISA(vehicle)	54	=
ISA(transportation)	123	=
S-LIKE(car)	24	

Abbildung 5.3: Selektivität, abgeleitet aus der Hierarchie (aus [LCHH97b])

5.1.2 Indexierung

Wird ein neues Bild eingefügt, werden bildbezogene Daten wie Größe, Format, Datum usw. automatisch extrahiert. Mit COIR erfolgt semiautomatisch die Erkennung von Inhaltsobjekten im Bild (anhand Segmentierung mit Farbe und Kontur) und deren räumliche Beziehungen zueinander (mit Mittelpunkt Abstand). Den gefundenen Inhaltsobjekten werden aus dem *Image Component Katalog* Bedeutungen anhand der Merkmale zugeordnet und dem Nutzer präsentiert. Der Nutzer kann dann den Vorschlag akzeptieren oder korrigieren. Neue Inhaltsobjekte werden mit den aus dem Inhaltsobjekt extrahierten Merkmalen unter einer vom Nutzer vorgegebenen Bezeichnung abgelegt. Falls gewünscht, können auch schon definierte Inhaltsobjekte anhand neuer Merkmale korrigiert oder erweitert werden. Mit dieser Methode wird der *Image Component Katalog* während des laufenden Betriebs schrittweise aufgebaut. Damit entsteht eine anwendungsabhängige Ontologie.

In Verbindung mit COIR steht der *Image Semantic Editor*, der semiautomatisch die räumlichen Beziehungen zwischen den definierten Inhaltsobjekten in jedem Bild erkennt und nach der Bestätigung durch den Nutzer, als Relation im Datenspeicher ablegt. Dort hat auch der *Textual Data Query Processor* Zugriff auf die Beziehungen.

Die Antwort des Systems an den Nutzer und die interne Anfragerformulierung arbeiten auf der erstellten Ontologie. Sie besteht aus den möglichen Prädikaten einer Anfrage und wird aus dem Wissen des *Terminology Managers* und Begriffen des *Image Component Katalog* abgeleitet. Die Hierarchie lässt sich vom Nutzer verändern und ergänzen. Ein Beispiel einer solchen Hierarchie ist in Abbildung 5.2 angegeben. Von den Beziehungen innerhalb der erstellten Ontologie lässt sich direkt auf die Ähnlichkeit der Objekte schließen. Dabei wird zwischen zwei Prädikattypen unterschieden: dem atomaren Basis-Prädikat $IS(\text{Objekt})$ und den zusammengesetzten abgeleiteten Prädikaten $S\text{-LIKE}(\text{Objekt})$ (für "Semantic like") und $ISA(\text{Objekt})$. Das Prädikat $ISA(\text{transportation})$ ist z.B. aus den disjunkten Prädikaten $IS(\text{car})$, $IS(\text{truck})$ und $IS(\text{bus})$ abgeleitet. Die Prädikate der Hierarchie sind durch sogenannte physische Zeiger verbunden. Mit den Basis-Prädikaten werden die Objekte im Bild durch physische Zeiger indexiert. Von jedem abgeleiteten Prädikat weisen logische Zeiger auf alle zugehörigen Objekte in den Bildern. Entsprechend der physischen Hierarchie wird die Selektivitäts-Statistik gepflegt. Die Selektivität abgeleiteter Prädikate wie z.B. $ISA(\text{transportation})$ mit einer Selektivität von 54 ergibt sich aus der Addition der entsprechenden Selektivitäten der Basis-Prädikate im Fall von $ISA(\text{transportation})$ ist es die Summe aus 30 ($IS(\text{car})$) + 15 ($IS(\text{truck})$) + 9 ($IS(\text{bus})$) wie Abbildung 5.3 beispielhaft zeigt.

5.1.3 Benutzerschnittstelle

Die Benutzerschnittstelle ist in drei Bereiche eingeteilt (siehe Abbildung 5.4). Der Arbeitsbereich dient zum Entwurf und zur Veränderung von Anfragen, Inhaltsobjekten und der erstellten Ontologie. Außerdem werden die Daten des System-Feedbacks im Arbeitsbereich grafisch dargestellt. Der Bereich mit der Werkzeugleiste stellt dem Nutzer Dialoge und Editoren zur Verfügung, mit deren Hilfe eine Anfrage erstellt wird. Der Ergebnisbereich zeigt den simultan erzeugten CSQL-Code während der Anfrageformulierung an oder enthält nach Abschicken der Anfrage die ermittelten Ergebnismengen in Form einer sortierten Liste mit Thumbnails der Bilder.

5.1.4 Iterative Anfrageformulierung

Die Anfragebearbeitung erfolgt rekursiv in vier Schritten:

1. Initiale Anfrage mit Inhaltsobjekten, Beispielbildern in Form von Dateien oder direkt erzeugte Skizzen (Sketch).
2. Ausführung der Anfrage.
3. Feedback vom System in Form von Alternativtabellen, die in der grafischen Anfrage den Inhaltsobjekten hinterlegt werden.
4. Änderung der Anfrage durch den Nutzer.

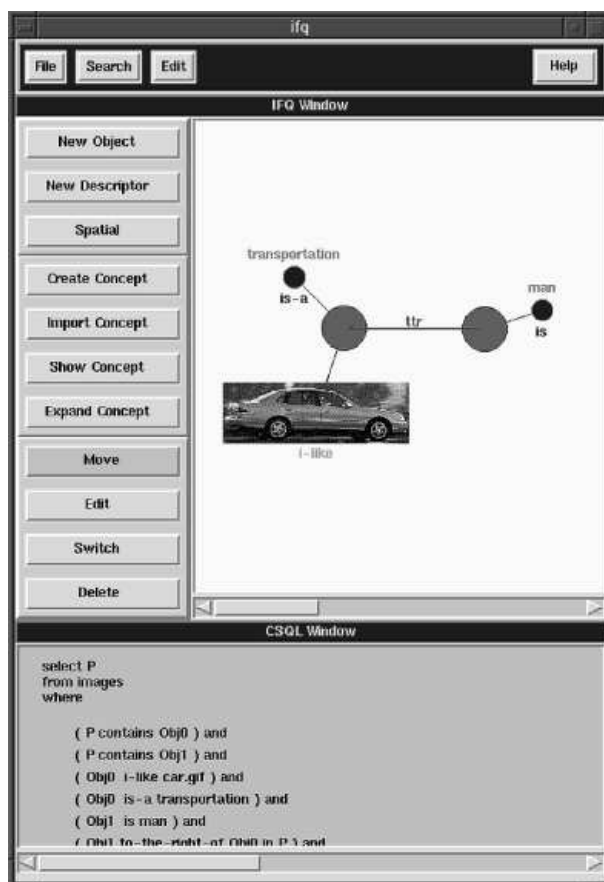


Abbildung 5.4: Benutzerschnittstelle von IFQ (aus [LCHH97b])

Über Dialoge werden Inhaltsobjekte in Form von Kreisen angegeben und mit geeigneten Prädikaten oder Beispielbildern versehen. Räumliche Beziehungen werden durch bezeichnete Linien zwischen den Inhaltsobjekten dargestellt (siehe Abbildung 5.5 oben links). Einsetzbare Prädikate sind `IS`, `ISA` und `S_like`, denen eine zugehörige Bezeichnung angehängt wird (siehe „isa Transportation“ in Abbildung 5.4). Beispielbilder können mit dem Prädikat `L_like` für „Image like“ eingefügt werden. Dabei können die Bilder eine in IFQ gezeichnete Handskizze oder eine Referenz auf eine Datei sein. Für die Bezeichnung der räumlichen Beziehungen werden Prädikate wie `above`, `below`, etc. benutzt, die wie alle anderen Prädikate aus einem Dialogfeld gewählt werden können.

Der in der Oberfläche entworfene Bildinhalt wird in die Anfragesprache CSQL übersetzt und an den *Facilitator* übergeben. Abgeleitete Prädikate werden durch die zugehörigen Basis-Prädikate ersetzt und durch Synonyme aus dem *Terminology Manager* ergänzt. Im Anschluss dazu wird die Anfrage auf die Komponenten aufgeteilt und ausgeführt. Als System-Feedback bekommt der Nutzer eine Liste der gefundenen Bilder und eine statistische Auswertung von Veränderungsmöglichkeiten in Form von Tabellen.

Alternativen werden für jedes `is-`, `isa-` und `s_like`-Prädikat der Anfrage aus der erstellten Ontologie ermittelt. Für `is(bus)` wird z.B. `isa(vehicle)` und `s_like(car)` (siehe Abbil-

dung 5.2) ermittelt. Die Alternativen sind den entsprechenden Prädikaten als Tabelle in der grafischen Oberfläche hinterlegt und können durch Auswahl des Prädikates sichtbar gemacht werden. Die Tabelle enthält zu jeder Alternative die Selektivität aus der Statistik und den Ähnlichkeitswert aus dem *Terminology Manager*. Aus der Tabelle wird ein *Erwartungswert* berechnet der in der grafischen Anfrage neben das Prädikat gesetzt wird. Er ist die Summe über alle mit der Selektivität multiplizierten Ähnlichkeitswerte.

$$Erwartungswert_{C_i} = \sum_{j=i}^n (\text{Ähnlichkeit}(C_i, R_{C_{i,j}}) \cdot \text{Selektivität}(R_{C_{i,j}})) \quad (5.1)$$

C_i ist das i -te Prädikat aus der Anfrage und $R_{C_{i,j}}$ die j -te zugehörige Alternative. Die Ähnlichkeit wird aus der Art der hierarchischen Verknüpfung und den, vom *Terminology Manager* ermittelten, Distanzen berechnet. Der Erwartungswert eines Prädikates ist ein Maß für das Potential einer Veränderung, wenn das aktuelle Prädikat durch die Alternative j ersetzt wird.

Für jedes eingefügte Beispielbild wird die Anzahl der ähnlichen Bilder ermittelt. Alle Bilder die einen bestimmten Grenzwert der Ähnlichkeit nicht unterschreiten sind ähnlich. Der Nutzer kann durch ändern der Grenzwerte Einfluss auf die Anzahl der angezeigten Bilder nehmen.

Das System-Feedback für räumliche Beziehungen wird aus dem Verhältnis der Anzahl der Inhaltsobjekt-Beziehungen, die mit der Anfrage übereinstimmen, zu der Anzahl der Beziehungen, die beide Inhaltsobjekte der Anfrage enthalten, ermittelt.

Mit Hilfe der ausführlichen Zusatzinformationen zum Ergebnis können Änderungen gemacht werden, deren Ergebnisse anhand der Alternativen und der Erwartungswerte abschätzbar sind.

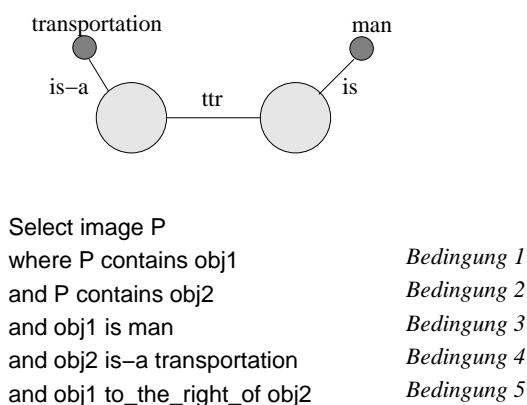


Abbildung 5.5: Grafische Anfrage und simultane Übersetzung (aus [LCHH97b])

Ein Beispiel einer grafischen Anfrage mit der dazugehörigen CSQL-Übersetzung ist in Abbildung 5.5 zu sehen. Oben ist die grafische Anfrage des Nutzers und darunter die CSQL Übersetzung, so wie sie der Nutzer in den beiden Fenstern sieht. Die Bedingungen

1 bis 5 sind durch Anpassungen in der grafischen Anfrage veränderbar. Die räumliche Bedingung „ttr“ steht für to-the-right. Angebotene Alternativen könnten sein: für Bedingung 3 IS(man) IS(woman) oder IS(human). Im Fall von IS(woman) wird die Anfrage im Anfrageraum verschoben bei dem Einsatz von IS(human) vergrößert sich der Anfrageraum, da durch das Einbinden einer höheren Hierarchiestufe der Ontologie IS(man) und IS(woman) mit eingeschlossen sind. Das Ändern anhand der angebotenen Alternativen kann so oft wiederholt werden bis die Ergebnismenge zufriedenstellend ist bzw. keine Verbesserung der Ergebnismenge erwartet wird.

5.2 CHARIOT

Die Bilddatenbank CHARIOT (Swiss (CH) Analysis and Retrieval of Image Objects) ist eine Entwicklung der ETH Zürich [Bos00]. Anhand von Beispielbildern werden ähnliche Bilder in der Datenbank gesucht. Die Ähnlichkeit wird dabei durch Vergleichen von Merkmalsvektoren bestimmt. Die gefundenen Bilder werden präsentiert und dann vom Nutzer bewertet. Das System wertet dann das Relevanz-Feedback des Nutzers aus. Eine der Besonderheiten von CHARIOT ist die dezentrale Speicherung der Bilddaten (z.B. beim Urheber). Der Zugriff des Systems und des Nutzers erfolgt ausschließlich auf die zentral gespeicherten Metadaten. Mit dieser Methode wird eine Suche nach Bildern angeboten, ohne dass das Urheberrecht verletzt werden kann.

5.2.1 Architektur

CHARIOT hat wie SEMCOG einen modularen Aufbau. Die Modularität beschränkt sich jedoch nicht nur auf die Komponenten, sondern ist auch auf der Ebene der Berechnungsmethoden innerhalb der Komponenten verwirklicht. Damit können alle Elemente bei Neuentwicklungen einzeln ausgetauscht oder ergänzt werden.

Die zentrale Komponente der Architektur ist der *Coordinator* (siehe Abbildung 5.6). Er koordiniert, ähnlich wie der *Facilitator* in SEMCOG die Kommunikation zwischen den einzelnen Komponenten. Die Geschwindigkeit des Anfragesystems wird von der Effizienz des *Coordinators* bestimmt.

Der *Client* nimmt die Anfragebilder und die Bewertung für das Relevanz-Feedback entgegen. Er stellt eine visuelle Benutzerschnittstelle zur Verfügung und übergibt die Anfragen an die *Search Engine*. In der *Feature Extraction Engine* werden für die Suche in der *Search Engine* die nötigen Merkmale extrahiert. Von der *Search Engine* werden die Anfragen für die *Index Engine* aufbereitet und die von der *Index Engine* berechneten Antwortlisten zu einer Ergebnisliste zusammengefasst. In der *Index Engine* erfolgt die Nächste-Nachbar-Suche. Sie ist der „Flaschenhals“ des Systems, da Ähnlichkeitssuche bei vielen Merkmalen sehr rechenaufwändig sein kann. Um die Zeit zu reduzieren werden die Prozesse in CHARIOT soweit möglich parallelisiert.

Das Relevanz-Feedback des Nutzers wird in der *Relevance Feedback Engine* zu einer

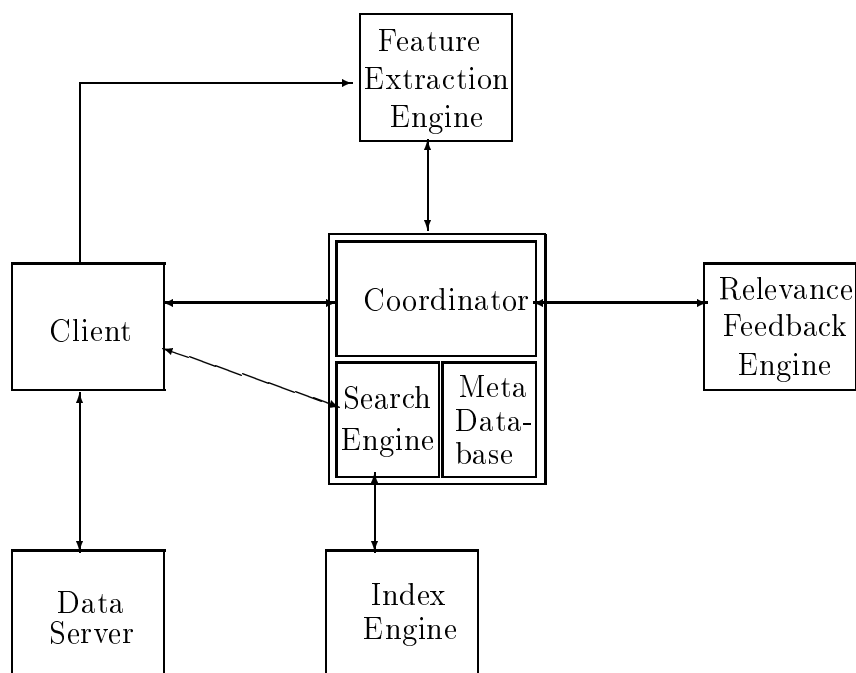


Abbildung 5.6: System Architektur von CHARLOT (aus [Bos00])

neuen Anfrage verarbeitet. Merkmale und andere Metadaten (Herkunft, Datum, Thumbnails usw.) werden in der *Meta Database* verwaltet. Die Originalbilder dagegen liegen auf dem *Data Server*.

5.2.2 Indexierung

Werden neue Bilder in die Datenbank eingelesen, werden im *Feature Extraction Engine* die Merkmale extrahiert. Im Moment sind in CHARLOT insgesamt 21 Merkmale implementiert, deren Dimensionen von 9 bis 320 variieren können. Jedes Merkmal spannt einen eigenen Merkmalsraum auf, in dem gesucht werden kann. Die Metrik wird dabei immer durch die euklidische Distanz (siehe Abschnitt 2.5) repräsentiert. Die Merkmale sind in einem VA-File abgespeichert [BMSW01]. In einem VA-File werden vereinfachte und nur angenäherte Repräsentationen der Merkmale in einem Grid-File [HS99] abgebildet. Der Zugriff auf die nächsten Nachbarn erfolgt in zwei Schritten. Zuerst werden die nächsten Nachbarn im Grid-File ermittelt, dann werden auf der ermittelten, kleineren Merkmalsmenge mit den Originalmerkmalen die exakten Distanzen bestimmt.

Merkmale von CHARLOT sind z.B. Farbmomente, Farbhistogramme und Gabor-Texturen. Die Merkmale werden in unterschiedlichen Granularitätsstufen eingesetzt. Außerdem erfolgt die Merkmalsextraktion auf dem gesamten Bild und auf mehreren Bildregionen.

5.2.3 Benutzerschnittstelle

Die Benutzerschnittstelle von CHARIOT wurde in einem Web-Browser verwirklicht. In aufeinanderfolgenden Fenstern können die Rahmenbedingungen für die iterative Suche gesetzt werden. Im ersten Fenster wird zur Wahl einer bestimmten Datenbank aufgefordert. In weiteren Fenstern erfolgt die Auswahl der einzusetzenden Merkmale und falls bekannt, kann auch direkt eine OID (Object Identifier) für ein Bild aus der Datenbank angegeben werden.¹

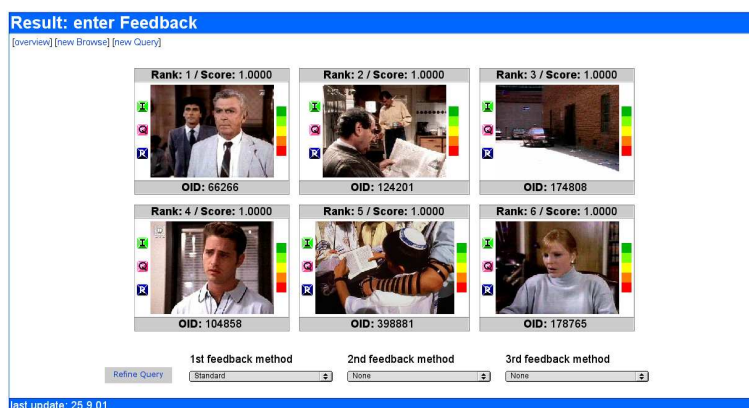


Abbildung 5.7: Ergebnisliste CHARIOT

5.2.4 Iterative Anfrageformulierung

Die initialisierende Anfrage im *Client* ist eine Menge von Beispielbildern². Beim Erstellen der Anfrage kann der Nutzer die Größe der Antwortmenge bestimmen und entsprechend der Demoversion im WWW, die bei der Suche einzusetzenden Merkmale auswählen. Damit können einzelne markante Merkmale der Zielmenge betont werden. Nach Abschicken der Anfrage übergibt der *Client* die Anfrage an die *Search Engine*.

Bei neu eingefügten Bildern, weist der *Coordinator* die *Feature Extraction Engine* an, die Merkmale der Bilder die im *Client* referenziert wurden zu extrahieren. Die extrahierten Merkmale werden vom *Coordinator* zur *Search Engine* weitergeleitet. Damit ist die Initialisierung des Anfrageprozesses abgeschlossen. In der *Search Engine* werden Unterabfragen für die einzelnen Merkmale erzeugt, die dann an die *Index Engine* übergeben wird. In der *Index Engine* werden für jedes Merkmal Ergebnislisten, die nach der Distanz aufsteigend sortiert sind, kalkuliert. Diese Listen werden an die *Search Engine* weitergereicht. Die *Search Engine* fasst die Ergebnislisten der Merkmale für jedes Bild zusammen, indem sie die durchschnittliche Distanz berechnet und die Liste nach der neuen Distanz sortiert. Die Ergebnislisten von den einzelnen Bildern werden im letzten Schritt

¹<http://simulant.ethz.ch/>

²im Prototypen kann nur ein Bild spezifiziert werden

wiederum durch Berechnung der durchschnittlichen Distanzen, zusammengefasst. Die aufgeführten Zwischenergebnisse müssen nicht berechnet werden, da CHARIOT auch das Endergebnis direkt berechnen kann. Sie werden aber für unterschiedliche Feedback-Verfahren eingesetzt.

Grafisch aufbereitet, mit Thumbnails und Ähnlichkeitsangaben aufgrund der Distanzen wird die Ergebnisliste dem Nutzer im *Client* präsentiert (siehe Abbildung 5.7).

Von jedem Bild einzeln ausgehend, kann vom *Client* aus, eine neue Anfrage gestartet werden. Entsprechen mehrere Bilder den Vorstellungen oder sind sie auch nur zum Teil relevant, kann jedes Bild an einer nebenstehenden 5-stufigen Skala bewertet werden (“highly non-relevant“, “non-relevant“, “no opinion“, “relevant“ und “highly relevant“). Wie die Nutzerbewertung in die Neuformulierung eingehen soll, kann durch sechs unterschiedliche, frei wählbare Feedback-Methoden beeinflusst werden. Das Relevanz-Feedback vom Nutzer wird mit den markierten Bildern und den ausgewählten Methoden an den *Coordinator* geleitet. Die Bewertung wird in Form einer Wichtungsmatrix (siehe Tabelle 4.1 Kapitel 4.6.2) an die *Relevance Feedback Engine* übergeben. Die ausgewählten Methoden werden von der *Relevance Feedback Engine* ausgeführt und zu einer neuen Anfrage zusammengefügt. CHARIOT bietet dazu folgende Methoden an:

- Dimensions-Wichtung mit *Sigma Measure*
- Merkmals-Wichtung
- Anfrageerweiterung durch Hinzufügen pseudorelevanter Bilder
- Anfragepunkt-Verschiebung mit
 - Rocchio und
 - RocchioPlus
- Anfrage-Wichtung durch Vergleichen sortierter Anfragelisten

Die Methode *Sigma Measure* benötigt mehrere Durchläufe, da sie für jedes Merkmal einzeln ausgeführt wird. Dazu nutzt diese Methode die Zwischenergebnisse der *Search Engine*.

Im Anschluss werden die Ergebnisse der Relevanz-Feedback-Methoden zusammengeführt und zurückgegeben. Die Reformulierung der Anfrage wird vom *Coordinator* an die *Search Engine* weitergeleitet. An der Schnittstelle zur *Search Engine* beginnt der nächste Iterationsschritt des Anfrageprozesses. Mit dem Unterschied, dass die *Search Engine* die als “nicht relevant“ markierten Bilder gesondert ablegt und sie bei jedem Durchgang aus den Antwortlisten der *Index Engine* entfernt. Somit können möglicherweise relevante Bilder “nachrücken“.

5.3 El Niño

El Niño ist eine Entwicklung des Visual Computing Laboratory an der University of California, San Diego. In El Niño sind die Suchstrategien Kontext-Feedback (siehe Kapitel 4.7) und Navigation verwirklicht. Der Nutzer beeinflusst den visuell präsentierten Datenbankinhalt, indem er die Distanzen zwischen den angezeigten Bildern manuell verändert oder mit Werkzeugen den dargestellten Datenraum verzerrt. Neben der Suche durch Navigation und Kontext-Feedback kann auf den textuellen Annotationen von Bildern und Bildgruppen gesucht werden. El Niño ist darauf ausgelegt, Bilder und Bildinformation aus dem WWW zu laden und die Bilder für mögliche Anfragen aufzubereiten. Die verwendeten Publikationen sind [SGJ01, SJ00, SJ99b, SJ99a, San99]

5.3.1 Architektur

Die Komponenten von El Niño sind der *Mediator*, mehrere *Engines*, der *Name Space* und das *Interface* auf dem die Anfragen vom Nutzer erfolgen.

Der *Mediator* übernimmt die Koordination des Anfrageprozesses. Er generiert Anfragen an die *Engines*, steuert das Kontext-Feedback und verwaltet den *Name Space*. Zum Zusammenführen von einzelnen Distanzberechnungen besitzt der *Mediator* Operatoren.

Die Berechnung der Ähnlichkeit wird von den *Engines* ausgeführt. Jede *Engine* hat eine eigene Metrik die „natürliche“ Distanzen zwischen den Bildern aufgrund der Ähnlichkeit bestimmt. Die Distanzen im aufgespannten Merkmalsraum können mit einem zugehörigen *Parameters Vector* verändert werden und zwar so, dass der Merkmalsraum sich krümmt. In El Niño sind drei unterschiedliche *Engines* realisiert: *Feature Engine*, *Textual Engine* und *Image Decomposition Engine*.

Die modulare Architektur erlaubt das Einbinden von weiteren *Engines*. Bedingung dabei ist, dass sie Distanzen zwischen vorgegebenen Bildern liefern und entweder einen *Parameters Vector* besitzen mit der die Distanzberechnung angepasst werden kann, oder anhand von vorgegebenen Distanzen die Distanzberechnung automatisch anpassen. Die Anpassung erfolgt über den *Optimizer* des *Mediators* oder, falls implementiert, über den *Optimizer* der *Engine*. Jede der *Engines* besitzt einen Extraktor für die Merkmale und eine Komponente in der die Ähnlichkeit zwischen den Bildern berechnet wird.

Der *Name Space* enthält die Referenzen zu den Originalbildern und eine ID für jedes Bild, über die der systemweite Zugriff auf die Bilder erfolgt.

5.3.2 Indexierung

Wird ein Bild in die Datenbank eingefügt, weist der *Mediator* dem Bild eine systemweit eindeutige ID zu. Die ID wird im *Name Space* mit einer Referenz auf das Originalbild abgelegt, dies ist im Fall des Downloads aus dem Web eine URL. Der *Mediator* weist jede *Engine* dazu an, die von ihr von Verfügung gestellten Merkmale aus dem Originalbild zu extrahieren.

Die textuelle Indexierung erfolgt während des Anfrageprozesses und bei der Extraktion aus dem Web.

Während des iterativen Anfrageprozesses werden *visuelle Konzepte* (siehe Abschnitt 4.7) abgespeichert die der Nutzer durch Gruppieren der Bilder in einer *Concept Box* (siehe Abschnitt 4.7) erstellt. Der Zugriff auf die visuellen Konzepte erfolgt anhand von, der *Concept Box* hinzugefügten, Begriffen, die ihren Inhalt beschreiben, z.B. „Nahaufnahmen von Gesichtern“. Eine weitere textuelle Zugriffsmöglichkeit erfolgt über das *Visual Dictionary*. Mit dem *Visual Dictionary* werden neben den visuellen Konzepten auch Annotationen von einzelnen Bildern referenziert, die in der Regel durch Extraktion aus dem Web gewonnen wurden.

Anzumerken ist, dass der textuelle Zugriff nur für den Einstieg in den iterativen Anfrageprozess vorgesehen ist. Die weitere iterative Suche erfolgt mit Kontext-Feedback und durch Navigation. Damit ist es nicht notwendig, die gesamte Datenbank zu annotieren. Anhand der Bilder die über das *Visual Dictionary* erreicht werden, werden intern die Bilder gesucht, welche die Präsentation vervollständigen.

5.3.3 Benutzerschnittstelle

Wie in CHARIOT ist die Benutzerschnittstelle für die Ausführung in einem Web-Browser implementiert. Auf der grafischen Oberfläche werden in erster Linie die Bilder präsentiert. In El Niño kann eine präsentierte Ergebnismenge zwischen 100 und 300 Bilder enthalten. Für die Wahl der Anzeige gibt es ein Auswahlfeld in dem die Ähnlichkeitsberechnung einer bestimmten *Engine* gewählt werden kann. Auch können Werkzeuge wie „Fischauge“ ausgewählt werden, um die Darstellung zu verändern. Es stehen Schieberegler für Anzahl und Größe der Bilder und für das Zoomen (für die Größe des sichtbaren Ausschnittes der visualisierten Ergebnismenge) zur Verfügung. Falls bekannt, kann auch ein Bild über die ID aus dem *Name Space* direkt referenziert werden.

5.3.4 Iterative Anfrageformulierung

Dem Nutzer steht offen, in einer ersten textuellen Anfrage mit einzelnen Stichwörtern Bildgegenstände zu beschreiben oder durch einen Zufallsgenerator eine Position in der Datenbank wählen zu lassen. Die im ersten Schritt erzeugte Präsentation dient als Einstiegspunkt in den iterativen Prozess der Anfrageformulierung.

Anhand der vorgenommenen Einstellungen werden vom Mediator aus der initialen Anfrage Teil-Anfragen erzeugt. Bei einer Anfrage mit einer Kombination aus Text und Farbe wird ein Kriterien-Graph vom *Mediator* erstellt, der eine textuelle Anfrage und eine Merkmalsanfrage mit einer UND-Verknüpfung verbindet. Die beiden Anfragen werden für jedes Bildpaar von den entsprechenden *Engines* ausgeführt. Die zurückgegebenen Ähnlichkeitswerte der *Engines* werden vom Mediator mit einem Operator für die UND-Verknüpfung verbunden. Die erhaltenen Ähnlichkeitswerte werden für die Bestimmung

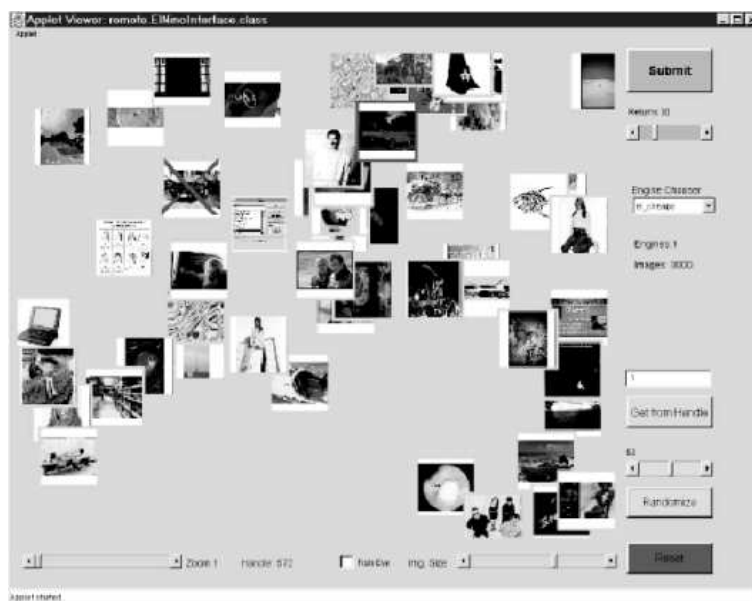


Abbildung 5.8: Benutzerschnittstelle von El Niño (aus [SGJ01])

der Distanzen auf der Nutzeroberfläche verwendet. In die Distanzberechnung fließen die Einstellungen des Nutzers, wie Zoom-Faktor und andere Darstellungsoperatoren ein.

Bei der Anfragereformulierung kann der Nutzer, mit drag-and-drop, einzelne Bilder im Arbeitsbereich auswählen und neu positionieren. Bilder, die einem Nutzerdefinierten Ähnlichkeitskriterium genügen, können einer *Concept Box* zugeordnet werden. Dazu werden sie in einem Rahmen am Rand des Arbeitsbereiches abgelegt, der benannt wird (siehe Abschnitt 4.7). Der Rahmen wird dann in Form eines Icons an der gewünschten Stelle positioniert.

Sind die gewünschten Bilder entsprechend ihrer Ähnlichkeit zueinander angeordnet, kann zusätzlich ein Referenzpunkt für die neue Anfrage gewählt werden. Der Referenzpunkt bestimmt die Ausmaße der neuen Präsentation, wobei der Referenzpunkt der Ursprung ist und nicht in die Neuberechnung mit einbezogen wird. Nach Abschicken der Anfrage werden die Distanzen zwischen den Bildern vom *Mediator* ermittelt und mit den Bild-IDs an die *Engines* weitergereicht. Wird der *Parameters Vector* durch den *Mediator* optimiert, wird nach jeder Änderung am *Parameters Vector* durch den *Mediator* eine Ähnlichkeitsberechnung von der *Engine* durchgeführt. Die zurückgelieferten Distanzen werden von einem internen Optimierer des *Mediators* mit den vorgegebenen Distanzen verglichen und gegebenenfalls der *Parameters Vector* erneut korrigiert. In den Optimierungsprozess wird der gesamte Datenbankinhalt mit einbezogen.

Ziel der Optimierung ist es, den Merkmalsraum so zu krümmen, dass bei einer möglichst geringen Abweichung von der ursprünglichen Distanzfunktion eine möglichst hohe Anpassung an die nutzergegebenen Distanzen erfolgt.

5.4 CBIR-VU

Der Name CBIR-VU leitet sich aus dem Herkunftsort des Systems ab: **C**ontent-**B**ased **I**mage **R**etrieval at the **V**isual Information Processing Laboratory at the **U**niversity of New South Wales. Schwerpunkt des CBIR-VU-Systems ist das Navigieren auf Zugriffsstrukturen. Daneben sind mehrere Retrieval-Methoden implementiert, welche die Suche anhand von Beispielbildern, Schlüsselwörtern oder anderen Vorgaben des Nutzers ermöglichen. Nach einer initialen Anfrage mit einer der Methoden, wird das Relevanz-Feedback-Verfahren für weitere iterative Schritte angeboten. Eine der Besonderheiten, von CBIR-VU ist, dass für das Retrieval keine Merkmale eingesetzt werden die aus dem Gesamtbild extrahiert werden. In [JGK97, JKX96, JKXB01] sind keine Verweise auf die darunter liegende Architektur gegeben, die deshalb im folgenden nur wenig detailliert beschrieben werden kann.

5.4.1 Architektur

Die Architektur besitzt fünf Haupt-Komponenten: Drei Suchmaschinen (jeweils eine für Textur, Kontur und Farbe), eine koordinierende Komponente und eine Benutzerschnittstelle. Die koordinierende Komponente vermittelt eingelesene Bilder zwischen den Suchmaschinen und führt die Ergebnisse der Suchmaschinen zu einer Ergebnismenge zusammen. Außerdem ist sie zuständig für die Segmentierung der Bilder. Die segmentierten Regionen werden für die Extraktion der Merkmale an jede der Suchmaschinen weiter gegeben. Während dem iterativen Anfrageprozess werden von der koordinierenden Komponente das Relevanz-Feedback unterstützt und die Berechnung der signifikanten Merkmale für die Ähnlichkeitsmessung.

5.4.2 Indexierung

Werden Bilder in das System integriert, erfolgt zunächst eine Bildsegmentation anhand homogener Regionen. Von jeder Region werden je drei Merkmale extrahiert: Kontur, Textur und Farbe. Die Indexierung der Regionen erfolgt mit einem SS^+ -Baum [JKXB01]. Die Zugriffsstruktur von Textur und Kontur wird in erster Linie zum Navigieren in den segmentierten Regionen herangezogen und erst in zweiter Instanz für die Nächste-Nachbar-Suche verwendet.

Die Nächste-Nachbar-Suche erfolgt in den meisten Fällen im gesamten Merkmalsraum der lediglich auf die aussagestärksten Dimensionen reduziert wird. In CBIR-VU sind die Merkmalsvektoren nicht mit Bildern oder Bildbereichen assoziiert, sondern mit segmentierten Regionen die dann die Schnittstelle zum Bild sind. Durch diese Vorgabe erfolgt die Suche auf möglichen Inhaltsobjekten, der Hintergrund und die Bildstruktur ist nicht im Merkmalsraum abgebildet.

Die Textur der einzelnen Regionen wird mit einem 30-dimensionalen Gabor-Filter bestimmt. Für die Farbindizierung werden Farbhistogramme eingesetzt, die nur die am

häufigsten auftretenden Farben in der Region wiedergeben. Mit der Wahl des Farbtons als Indexierungsgröße wird die Farbintensität vernachlässigt, was nach [JGK97] der menschlichen Wahrnehmung entgegenkommt. Dieses Verfahren hat sich für die Gesamtbewertung eines Bildes als ungeeignet erwiesen aber im Bereich der Regionen, wie es hier der Fall ist, ist die Methode effizient. Bei der Indexierung der Region nach der Kontur wird der „Smallest Enclosing Circle“ (SEC) eingesetzt ([JKX96]) und mit Hilfe der dem Huffman-Kodierung die aussagestärksten Längenverhältnisse zwischen dem SEC und der Kontur ausgewählt und zur Indexierung eingesetzt.

Für die Navigation werden die Mittelpunkte aller Cluster im SS^+ -Baum für Kontur und Textur ermittelt. Die, dem Mittelpunkt am nächsten liegende Region wird mit dem zugehörigen Bild als Repräsentant des Clusters abgelegt. Dadurch ergibt sich eine Baumstruktur deren Knoten durch Bilder dargestellt werden können, die die Eigenschaften der darunterliegenden Cluster als visuelle Zusammenfassung repräsentieren. Entsprechend den Eigenschaften eines SS^+ -Baums sind die Bilder der Cluster-Mittelpunkte in den höheren Ebenen näher am Datenbankdurchschnitt und liegen weiter auseinander als die Mittelpunkte der niedrigeren Ebenen. Im Bereich der Blätter erfüllen nur noch wenige Bilder die speziellen Eigenschaften der Cluster.

Unabhängig von den Regionen ist noch der Zugriff über Schlüsselwörter möglich die durch Annotation den Bildern hinzugefügt werden können.

5.4.3 Benutzerschnittstelle

Die Nutzeroberfläche ist in vier Bereiche aufgeteilt. Jeweils ein Bereich für Einstellungen an Textur, Farbe oder Kontur und einen textuellen Bereich für Eingaben und zur Dokumentation der aktuellen Nutzereinstellungen. Wird eine Textur innerhalb des Texturbereichs ausgewählt, werden ihre Werte im Feld „Texture Values“ im textuellen Bereich angezeigt und können dort auch manuell verändert werden (in soweit man die Bedeutung der Werte versteht). Für jede Retrieval-Methode gibt es einen Button der entweder eine Anfrage mit den vorgegebenen Einstellungen abschickt oder neue Dialogfenster für weitere Einstellungen öffnet, z.B. zum einlesen von Beispielbildern aus einer Datei.. Zum Navigieren auf der Zugriffsstruktur werden im Kontur- und Textur-Bereich 2×3 repräsentative Cluster-Bilder angezeigt. Die jeweils linken Bilder entsprechen dem „Eltern-Knoten“ im SS^+ -Baum und die beiden rechten entsprechen jeweils dem linken und rechten Sohn. In Abbildung 5.9 ist der Aufbau der Benutzeroberfläche angedeutet.

Die Ergebnismenge erscheint in einem Extrafenster in Form einer sortierten Liste von Thumbnails. Die Thumbnails können für eine vergrößerte Ansicht und zum editieren in einem weiteren Fenster geöffnet werden.

Die Bereiche Textur und Kontur enthalten jeweils die Beispielbilder von zwei Cluster-Ebenen, durch Auswahl eines Bildes erfolgt ein Update des Bereiches, in dem die Beispielbilder der nächsten Ebene angezeigt werden.

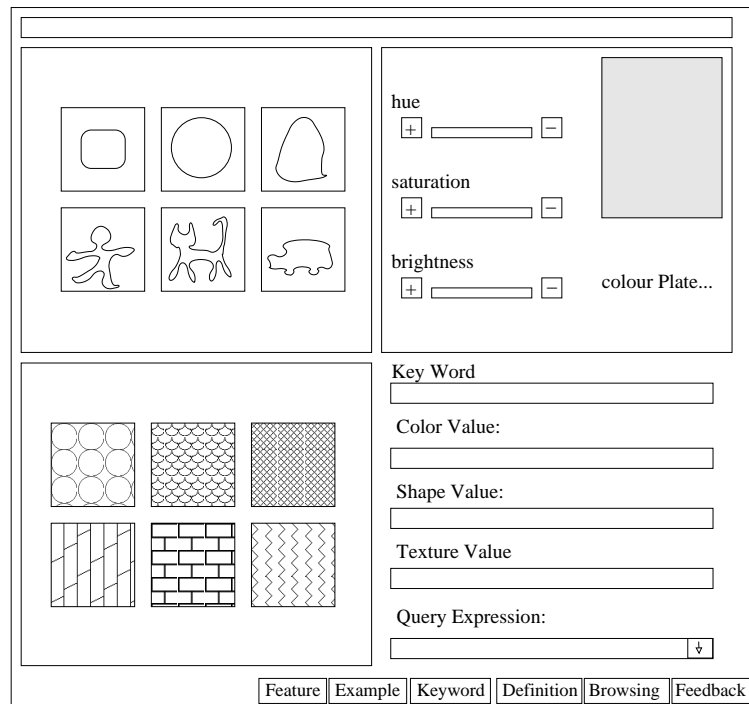


Abbildung 5.9: Benutzerschnittstelle von CBIR-VU (angelehnt an [JKXB01])

5.4.4 iterative Anfrageformulierung

Der Nutzer hat insgesamt sechs Möglichkeiten sich auf den Bildern der Datenbank zu bewegen. Er kann anhand einzelner Merkmale suchen, Beispielbilder einfügen, Schlüsselwörter einsetzen, eine zusammengesetzte Anfrage erstellen, auf der Zugriffsstruktur browsen und über Relevanz-Feedback bewertete Anfrageergebnisse als neue Anfrage abschicken.

Die Einstellungen in den einzelnen Bereichen können im „Expression“-Feld beliebig durch ein Auswahl-Menü mit AND und OR Verknüpfungen zu einer Anfrage verbunden werden. Wird z.B. beim Abschicken einer Anfrage folgender Ausdruck übergeben: „(shape or texture) and colour“, werden aus den entsprechenden Bereichen die momentanen Einstellungen übernommen und zu einer Anfrage zusammengestellt.

Das Browsen oder Navigieren auf den Clustern von Textur und Kontur erfolgt durch Auswahl des jeweils am geeignetsten erscheinenden Cluster-Bildes. Das dann als „Vater-Knoten“ mit den Bildern der nächst tieferen Cluster-Ebene erscheint.

Eine weitere Möglichkeit bietet die beliebige Bestimmung einzusetzender Merkmale. Steht ein Beispielbild in Form einer roten Blume zur Verfügung, wird aber eine Blume der gleichen Sorte in der Farbe Gelb gesucht, kann das Beispielbild den Bereichen Textur und Kontur zugewiesen werden und im Fenster-Bereich der Farbe der gewünschte Gelbwert

(oder auch ein Beispielfeld mit der Farbe) eingestellt werden.

Das Retrieval mit Relevanz-Feedback erfolgt in zwei Schritten. In den meisten Fällen startet der Nutzer einen iterativen Anfrageprozess mit einer Anfrage die nur partielle Information enthält, wie zum Beispiel Farbangaben zum Bild. Der erste Retrieval Schritt kann nur diese Information nutzen und somit nicht den gesamten Index des Systems. Die Ergebnismenge wird in einem Extrafenster präsentiert. Dem Nutzer steht frei einzelne Bilder als relevant zu markieren oder Ergebnisbilder zu editieren um einzelne Regionen anhand von Polygonen als relevant zu markieren. Der Nutzer kann eine beliebige Kombination von Textur, Kontur und Farbe wählen, die dann für die Suche aus den Region extrahiert werden und mit den entsprechenden Merkmalen der gewählten Bilder für das Relevanz-Feedback genutzt werden.

Um bei einer Anfrage mit Beispielfeldern oder auch während den iterativen Schritten im Relevanz-Feedback eine möglichst schnelle Suche zu erhalten, werden die Dimensionen des Merkmalsraums in einem ersten Schritt reduziert. Dazu werden Merkmale mit geringer Aussagekraft anhand ihrer Verteilung herausgefiltert.

5.5 VIQING

VIQING ist eine Entwicklung vom Department of Electrical Engineering and Computer Sciences an der University of California at Berkeley [OSAH98]. Durch die direkte Spezialisierung auf das Gebiet der Geographie ist VIQING eine Ausnahme in den vorgestellten Systemen. VIQING besitzt eine grafische Schnittstelle für Selektion, Projektion und Vereinigung auf visualisierten Daten. Das relationale Datenbanksystem heißt DataSplash und bietet eine Schnittstelle für die Datenbankinhalts-Visualisierung an. Die Schnittstelle benutzt ein einfaches Zeichenprogramm zum Erzeugen von 2-dimensionalen Gebieten, die mit Polygonen gezeichnet werden. Die Zeichnungen werden in Form von *Folien* abgelegt. *Folien* die mehrere Gebiete enthalten repräsentieren eine Relation mit weiteren *Folien*. Enthält eine *Folie* nur ein Gebiet entspricht sie einem Tupel der Relation, in dem der Polygonzug des Gebietes abgelegt ist.

Die Lage der Gebiete auf der Folie stellen die räumlichen Beziehungen zwischen den Gebieten dar. Visuelle Eigenschaften wie Form (Polygon) und Farbe der Gebiete werden durch Attribute festgelegt. Damit erfolgt eine Trennung von räumlichen und nicht räumlichen Merkmalen. Während der Suche können *Folien* „betreten“ werden. Dadurch erhält der Nutzer Zugriff auf die enthaltenen *Folien*. Die Zugriffsstruktur ähnelt einer Baumstruktur, muss aber nicht zwingend einen Wurzelknoten haben. Die Struktur kann in alle Richtungen beliebig durchlaufen werden. Damit erfolgt die Navigation wie sie auf Internetseiten durch Links bekannt ist.

5.6 CHABOT

CHABOT [OS95] ist eine weitere Entwicklung der University of California, Berkeley. Es baut auf dem erweiterten relationalen Datenbankmanagementsystem POSTGRES auf und unterscheidet sich dadurch wie VIQING von allen anderen aufgeführten Anfragesystemen. POSTGRES erlaubt die Definition von Klassen, Datentypen und Funktionen so wie das Anlegen von Zugriffsstrukturen mit R- und B-Bäumen. Attribute in CHABOT sind Datentypen wie Text, numerische Daten, Zeit und geografische Lokalisierung. Der Datentyp Text wird neben dem Einsatz für assoziierte Daten (Fotograph, Ort, Titel, Bildinhalt usw.) auch für die Beschreibung der Farbverteilung im Bild (kein Merkmalsvektor) eingesetzt.

Bildinhalte werden mit Textattributen wie „comments“, „title“, „perspective“ oder „description“ beschrieben. Die Darstellung der Farben durch Attributwerte erfolgt anhand einem Farbhistogramm dessen Bins Farbnamen zugeordnet werden. Der RGB-Farbenraum wird dazu in 20 Farbbereiche eingeteilt. Anhand der Anzahl der Pixel, die in ein Bin fallen wird noch eine Häufigkeitsbeschreibung hinzugefügt: „mostly“ wenn 50% oder mehr, aller Pixel im Bild der Farbe entsprechen im anderen Fall wird „some“ angefügt z.B. mostlyRed oder someBlue.

Neben Bilddaten werden auch Video und Textdokumente in CHABOT auf ähnliche Weise abgelegt.

Zur Trennung der Originalbilder von den Metadaten und Thumbnails (für die Präsentation) erfolgt eine Speicherung auf zwei Ebenen. Bei CHABOT sind die Bilder auf Magnetbändern gespeichert und die Daten auf denen gesucht wird auf schnellen Festplatten.

Die Nutzerschnittstelle soll auf möglichst einfache Weise Anfragen an das System erlauben. Außerdem sollen alle Anfragearten (geografisch, zeitlich und inhaltlich) miteinander kombiniert werden können. Dazu sind auf der Nutzeroberfläche alle Attribute durch Textfelder oder pull-down-Menüs repräsentiert, je nach dem ob die Attributwerte frei wählbar sind einen vorgegebenen Wertebereich haben. Anfragen werden durch optionales auswählen aus den pull-down-Menüs und/oder Eintragungen in die Textfelder erzeugt und dann über einem „Submit“-Schalter abgeschickt. Die Ergebnisse werden in einem neuen Fenster neben der Anfrage dargestellt. In weiteren iterativen Schritten können entsprechend der Ergebnisvorstellung die Feldinhalte verändert oder ergänzt werden. Ist eine zufriedenstellende Ergebnismenge erreicht kann die Anfrage bei Bedarf als Konzept unter einem Namen abgespeichert werden. Das Konzept kann dann als eigenständige (Teil-)Anfrage in einem pull-down-Menü im Anfragefenster ausgewählt werden.

CHABOT erzielte die besten Ergebnisse, wenn die Anfragen mit einer Kombination aus Text und Farbmerkmal ausgeführt wurden. Lücken in der Bildbeschreibung werden dadurch teilweise mit den angegebenen Farbmerkmalen ausgeglichen. Fehlt Text bezüglich benennbarer Inhalte kann das Bild auch nicht durch Ähnlichkeitssuche bezüglich der Farbe gefunden werden. CHABOT eignet sich aufgrund der sehr einfachen Repräsentation von Farbmerkmalen besonders für die Suche auf assoziierten Daten, bei

denen die Ähnlichkeit nicht ausschlaggebend ist. In erster Linie wird nach benanntem Inhalt gesucht, Blickwinkel und Gestaltung sind zweitrangig. Eine Typische Anfrage wäre z.B. „See X aus der Vogelperspektive“ oder „Blumen mit einem Farbwert von „someYellow“.

5.7 VIBE

VIBE ist ein Video-Retrieval-System [CTDB98], benutzt aber zur Suche einzelne Bilder aus den Sequenzen /Szenen. Die Bilder werden aus den einzelnen Iterationsschritten in Form einer sortierten Liste ausgegeben. Aus der Liste kann der Nutzer einzelne Bilder entnehmen und in einem Editierfeld zu Clustern zusammenstellen, die jeweils eine vom Nutzer definierte Ähnlichkeit ausdrücken. Ähnlich wie in El Niño versucht das System die Distanzen in der Art neu zu berechnen, dass die erhaltenen Bilder die vorgegebenen Cluster bilden und gegebenenfalls durch neue aus der Datenbank ergänzt werden.

5.8 VIPER

VIPER arbeitet auf über 85 000 farblichen und räumlichen Merkmal-Ausprägungen. Die Merkmalsextraktion erfolgt lokal und global auf mehreren Skalierungsstufen des Bildes. VIPER ist, vergleichbar mit CHARIOT. Es ist ebenfalls ein Relevanz-Feedback-System das eine sortierte Liste als Ergebnis liefert.

Die Hauptunterschiede zwischen VIPER und CHARIOT liegen in der Bildung des Anfrage-Vektors. VIPER benutzt für die Reformulierung eine binäre Bewertung. Wird ein Bild als relevant oder nicht relevant markiert, wird es in die Anfragemenge von Beispielen mit entsprechendem Vorzeichen (+1 oder -1) aufgenommen. Die Bilder der Anfrage werden dem Nutzer immer angezeigt. Damit kann der Nutzer jeder Zeit die einmal gemachte Bewertung korrigieren. Als relevant markierte Bilder können nur einmal in die Anfragemenge aufgenommen werden. Nicht relevante Bilder hingegen können mehrmals in der Ergebnisliste und Anfragemenge erscheinen. Eine Besonderheit von VIPER ist die Integration des Benutzerverhaltens, wie sie in Abschnitt 4.3.1 beschrieben ist.

Kapitel 6

Auswertung

Im Ersten Abschnitt werden die Anfragesysteme miteinander verglichen. Im zweiten Abschnitt erfolgt eine Klassifikation der gefundenen Ansätze.

6.1 Bewertung

Die Anfragesprachen SEMCOG, CHARIOT, CBIR-VU und El Niño sind zum Teil grundverschiedene Anfragesysteme. SEMCOG unterstützt ausschließlich die manuelle Reformulierung der Anfrage, während CHARIOT nur das Relevanz-Feedback im Merkmal-Raum unterstützt. CBIR-VU bietet eine Kombination aus manueller Anfrageformulierung, Relevanz-Feedback und Browsen auf einer Baumstruktur an, dabei werden alle Inhaltsbeschreibungen anhand von Merkmalen bezüglich extrahierter Regionen gemacht. Das Anfragesystem El Niño bietet ein umfassendes Kontext-Feedback an mit Ähnlichkeitsvorgaben und Gruppieren von Bildern. Während in CHARIOT und CBIR-VU ausschließlich eine sortierte Liste der Ergebnismenge angezeigt wird, werden in SEMCOG und El Niño komplexere Datenzusammenhänge angezeigt. SEMCOG ergänzt die Ergebnismenge durch Alternativangaben und El Niño veranschaulicht Zusammenhänge der Ähnlichkeit und Inhalte der Datenbank durch eine 2-dimensionale Präsentation der Ergebnismenge. Die fehlende Umgebungsinformationen wird in El Niño teilweise dadurch ausgeglichen, dass der Nutzer auf den angelegten Clustern browsen kann. Die Gemeinsamkeit der vier Haupt-Systeme ist eine koordinierende Komponente. Auf diese Weise wird jeweils das Problem gelöst unterschiedliche Anfragemechanismen mit unterschiedlichen Bildbeschreibungen zu verbinden. Die Anfragen werden generell in Teilanfragen zerlegt deren Ausführung von zuständigen Teilkomponenten übernommen werden. Die Teilergebnismengen werden in der koordinierenden Komponente zu einem Gesamtergebnis zusammengeführt.

Während in SEMCOG relationale und objektorientierte Speicherung mit der Suche im Merkmal-Raum verknüpft wird, kombiniert CBIR-VU die Suche auf Clustern und im Merkmal-Raum. El Niño arbeitet ebenfalls im Merkmal-Raum und bietet als Erweiterung dazu die Suche in einem Katalog an der, vom Nutzer gruppierte Bilder aufnimmt.

CHARIOT arbeitet ausschließlich auf dem Merkmal-Raum die Zerlegung der Anfrage in der Komponente für die Koordinierung erfolgt ausschließlich für die unterschiedlich zu berechnenden Merkmale.

Die Außenseiter bei den Anfragesystemen sind CHABOT und VIQING. Sie arbeiten vollständig auf relationalen Systemen. Eine der Gründe für die Entwicklung solcher Systeme ist ein schon vorhandener Bestand an Bilddaten und Metadaten die relational abgespeichert vorliegen und für die nur ein Retrieval-System entwickelt werden soll.

Eine Eigenschaft die keine der Systeme unterstützen, ist das rückgängig machen einer Anfrage. Da es aber zu erwarten ist, dass auch Fehlentscheidungen während der Suche gemacht werden, sollte diese Option für die Anfragen zur Verfügung stehen.

Die Reduzierung der Dimensionen und die Auswahl der geeigneten Merkmale wird in allen Anfragesystemen größtenteils vom System übernommen. In CHARIOT und CBIR-VU ist dies nur im geringen Maße für den Nutzer möglich darauf Einfluss zu nehmen. In CHARIOT kann zu Beginn eines iterativen Anfrageprozesses einmalig eine Auswahl bezüglich der Merkmal-Repräsentationen getroffen werden. Eine etwas erweiterte Funktionalität bietet CBIR-VU mit der flexiblen Zusammenstellung von Merkmalen aus Beispielbildern oder ausgewählten Clustern. Ein Verfahren das aus dem Text-Retrieval bekannt ist, wird aber bei keinem der Systeme angeboten. Hierbei werden Änderungen an der Anfrageformulierung vor der Präsentation einer Ergebnismenge dem Nutzer präsentiert, so kann der Nutzer Terme aussortieren die er mit Sicherheit nicht in der Anfrage haben will [CMRB01]. Auch wenn sich dieses Verfahren nur begrenzt auf die Suche in Bildern übertragen lässt, weil der Mensch einige der ergänzten Merkmale sowieso nicht interpretieren kann, könnten bei einer geeigneten Präsentation einzelner ausgewählter Merkmale und anderer Inhaltsbeschreibungen ein Überblick über die „Richtung“ der Reformulierung und der zu erwartende Ergebnismenge geschaffen werden. Mit Hilfe des Überblicks können dann Korrekturen vorgenommen werden.

Nachteile von SEMCOG sind, das die eingesetzte Ontologie und die vorgegebenen Prädikate die Bildbeschreibung einschränkt. Auch wenn über die Suche im Merkmal-Raum Beispielbilder in die Anfrage eingefügt werden können sind durch den stark eingeschränkten Einfluss des Nutzers auf die Suche im Merkmal-Raum keine Situationsorientierten Ergebnisse zu erwarten. Es ist bei CBIR-VU nicht zu erwarten, dass die Bedienung sehr nutzerfreundlich ist, mit dem Überangebot an Kombinationsmöglichkeiten und Retrievalverfahren ist die Desorientierung des Nutzers nicht unwahrscheinlich. Durch die ausschließliche Suche auf automatisch extrahierten Regionen können in CBIR-VU keine Gebiete gefunden werden die nicht oder falsch extrahiert wurden. Bis auf SEMCOG kann in keinem der Anfragesystemen eine direkte Suche nach benannten Inhaltsobjekten mit räumlichen Beziehungen in einem Bild gemacht werden. Eine weitere allgemeine Schwierigkeit der iterativen und unscharfen Anfragesysteme ist das Ende der Suche. Es ist in der Regel schwer abzuschätzen ob noch weitere relevante Bilder in der Datenbank vorhanden sind oder nicht, da der Nutzer nicht wissen kann ob intern der Anfrage-Raum so stark spezifiziert ist, dass weitere Iterationsschritte keine große Veränderung mehr bewirken können und der Nutzer sich somit in einer „Sackgasse“ befindet oder ob das

gesuchte Bild nicht in der Datenbank ist. Damit wird in der Regel die Suche dann beendet sein, wenn der Nutzer genügend relevante Bilder erhalten hat oder der Eindruck besteht, dass keine Verbesserung mehr zu erwarten ist.

Da nicht alle Systeme zugreifbar für Tests sind, konnten keine allgemeinen Aussagen über die Geschwindigkeit oder Effizienz gemacht werden.

6.2 Klassifikation

Eine Klassifikation nach den Anfragesystemen ist aus folgenden Gründen ungeeignet: Die vorgestellten Anfragesysteme bestehen in vielen Fällen aus einer Kombination mehrerer unterschiedlicher Ansätze. Einige der Ansätze sind in mehreren Anfragesystemen verwirklicht. Dadurch überlappen sich die Funktionalitäten der Anfragesysteme und verhindern so eine deutliche Trennung in einer Klassifikation. Ein weiterer Punkt ist die in Kapitel 5 erwähnte kontinuierliche Weiterentwicklung der Anfragesysteme. Dadurch wäre eine Klassifikation nur sehr kurzzeitig aktuell.

Für die Klassifikation der iterativen Suche (siehe Abbildung 6.1) werden die verwirklichten Ansätze definiert und zueinander in Beziehung gesetzt. Bei der vorliegenden Klassifikation wird die initialisierende Anfrage nicht mit betrachtet. Zwei Gründe dafür sind, dass die initiale Anfrage meist unabhängig von der iterativen Veränderung der Anfrage ist und es oft mehrere, vollkommen verschiedene, Formulierungsmöglichkeiten für die initiale Anfrage gibt.

Die Ansätze der Such-Strategien lassen sich in einer Klassifikation in drei Hauptebenen zusammenfassen. Die erste Ebene unterscheidet die Ansätze anhand von vier Suchstrategien. In der zweiten und dritten Ebene werden die Suchstrategien anhand der vom Nutzer zu machenden Angaben unterschieden und der damit verbundenen Variablen.

Mögliche Suchstrategien sind Kontext-Feedback, Relevanz-Feedback, manuelle Anfrageformulierung und Navigation bzw. Browsen.

Browsen

Browsen trifft auf Anfragereformulierungen zu die keine direkten Eingaben des Nutzers bezüglich der Anfrage benötigen. Statt dessen bewegt sich der Nutzer auf grafisch dargestellten Datenbankinhalten ähnlich wie im World-Wide-Web. Anhand der Darstellung schätzt der Nutzer seine momentane Position innerhalb der Datenbank bezüglich der gewünschten Ergebnismenge ab und verändert die Position durch einfache Auswahlmethoden. Es erfolgt keine direkte Reformulierung von Anfragen und damit auch keine Interne Veränderung der Distanzmessung oder anderer Ähnlichkeitsberechnungen. Das Browsen kann je nach interner Repräsentation oder entsprechend der Darstellung auf der Benutzeroberfläche im Raum einer mehr-dimensionalen Präsentation, auf Clustern oder in einer Liste erfolgen.

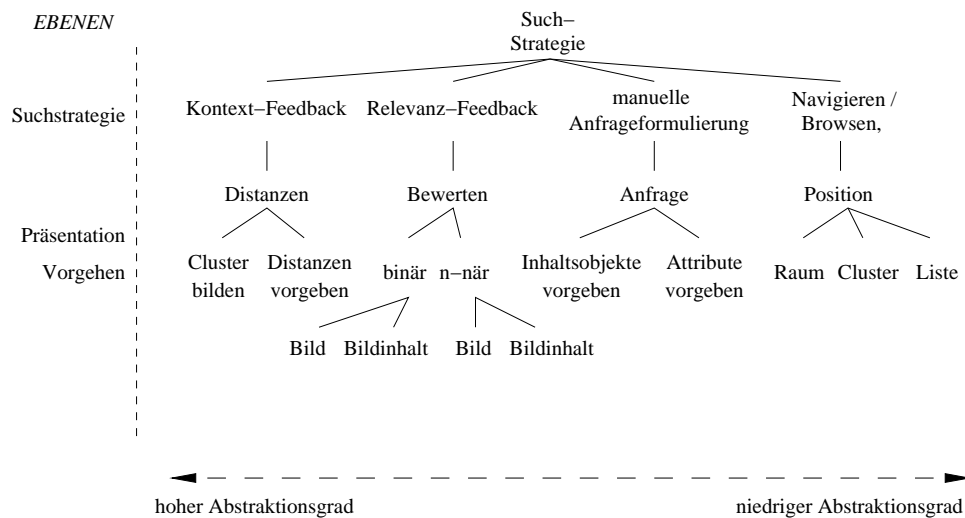


Abbildung 6.1: Klassifikation der iterativen Anfrageformulierung in grafischen Anfragesprachen

manuelle Anfrageformulierung

Die manuelle Reformulierung der Anfrage erfordert Wissen über die Syntax der Anfragesprache. In vielen Fällen ist das Vokabular für die Inhaltsbeschreibung bzw. sind die einsetzbaren grafischen Elemente vordefiniert. Unterstützung durch das System erfolgt nur noch über die Ergebnispräsentation. Anhand der Ergebnismenge wird die nächste Anfrage manuell formuliert oder die initiale Anfrage direkt geändert. Im Fall der manuellen Reformulierung, ist das Vorgehen bei der Formulierung der initialen Anfrage und der folgende Anfrageanpassungen identisch.

Relevanz-Feedback

Beim Relevanz-Feedback erfolgt eine Bewertung der Ergebnismenge durch den Nutzer. Diese geht direkt in eine Reformulierung der Anfragen durch das System ein. Der Einfluss des Nutzers auf die Anfragerreformulierung ist durch die als relevant bewertbaren Elemente bestimmt bzw. auch begrenzt. Für eine gezielte Bewertung sollte der Nutzer wissen, wie die Variablen die hinter der Bewertung liegen die Anfragerreformulierung beeinflussen. Mit dem Wissen kann er die geeigneten Bewertungen für die Veränderung machen. Dazu gehören binäre und n-näre Bewertungen, die jeweils auf einzelne Bildinhalte bezogen sein können oder das ganze Bild betreffen.

Kontext-Feedback

Beim Kontext-Feedback erfolgt eine „künstliches“ erzeugen einer Ähnlichkeit durch den Nutzer. Der Nutzer gibt Distanzen vor die vom System über Optimierungsprozesse nach-

gebildete werden. Die Distanzen können vorgegeben werden in dem direkt die Distanz von einzelnen Bildern zueinander vorgegeben werden oder durch zusammenfügen von Bildgruppen zu Cluster die das System dann als Einheit bewertet. Innerhalb der Cluster bestehen aber keine konkreten Vorgaben der Distanzen zwischen den einzelnen Bilder.

In Abbildung 6.1 ist die Klassifikation graphisch dargestellt.

Die Klassifikation ist so aufgebaut, das von rechts nach links in allen Ebenen eine steigende Abstraktion der Such-Strategie zu finden ist.

Einordnung der Anfragesysteme in die Klassifikation

Ordnet man die Sprachen aus dem Kapitel 5 in die Klassifikation ein, ergeben sich folgende Zusammenstellungen der Ansätze.

SEMCOG verwirklicht den Ansatz der Inhaltsobjektorientierten manuellen Anfragevisualisierung mit ausführlichen Systeminformationen für die Ergebnisauswertung. El Niño verbindet beide Kontext-Feedback-Verfahren (Cluster- und Distanzbestimmung) während VIBE sich auf die Cluster-Methode beschränkt. CHABOT ist ein manuelles Anfragesystem bei dem die Attribute verändert werden. Die Strategie von VIQING lässt sich zum Browsen über Cluster zuordnen. VIQING unterstützt als einziges System keine Ähnlichkeitssuche. Die Suche erfolgt auf festgelegten Strukturen die durch die durch Gruppierungen der Polygone auf Folien definiert. Demnach lassen sich auch Systeme in die Klassifikation einordnen die keine Ähnlichkeitsberechnung unterstützen. CBIR-VU kombiniert Browsen in Cluster und die binäre Relevanz-Feedback-Strategie mit Bewertung von Bildern und Inhalten. CHARIOT und VIPER arbeiten mit Relevanz-Feedback auf Bildebene wobei CHARIOT eine n-näre Bewertung ermöglicht und VIPER einer binäre.

6.3 Diskussion

Werden die Kombinationsmöglichkeiten die sich aus der Vielzahl an Ansätzen und Variablen einer iterativen Suche ergeben betrachtet, ist für eine organisierte Suche nach einer situationsgebundenen, idealen Kombination von Suchstrategien, Inhaltsbeschreibungen, Anfrageformulierungen, Reformulierungsmethoden und Ergebnisvisualisierung eine Umgebung notwendig wie sie für das Text-Retrieval mit Umgebungen wie OKAPI oder TREC (siehe Abschnitt 2.6) schon existieren. Zu einem Bedienungsfreundlichen Retrieval-System gehört die Ausgewogenheit von Eingabeaufwand und Zeitaufwand für die Suche. Eine allgemeine Bedingungen an Benutzeroberflächen, dass der Nutzer sich über seine momentane Position im ausgeführten Prozess klar sein sollte (Abschnitt 3.2) könnten in Form einer History-Liste realisiert werden. Mit einer History-Liste könnte auch ein Backtracking im Retrieval ermöglicht werden. Eine der menschlichen Eigenschaften ist, dass Textur, Farbe und Kontur jeweils separat betrachtet werden [JGK97]. In vielen Anfragesprachen wird dies durch Anbieten der getrennten Auswahl dieser Merkmale unterstützt, die Ergebnismengen werden jedoch als „over-all-Liste“ ausgegeben. Ei-

ne mögliche Erweiterung für eine iterative Such-Strategie wäre demnach eine getrennte Ausgabe von Ergebnislisten bezüglich der Merkmale, erst damit wäre es dem Nutzer möglich den Einfluss eines einzelnen Merkmals wirklich abzuschätzen und entsprechend zu verändern. Zur flexibleren Bestimmung der Merkmale gehört auch die Wahl feinerer oder gröberer Merkmale, wie z.B. die Möglichkeit während der Iteration manuell von einem 16-Farbenhistogramm zu einem 24-Farbenhistogramm zu wechseln wenn der Nutzer die präsentierten Farben in der Ergebnismenge für geeignet hält. Die, in der Bewertung schon erwähnte, Korrektur vor der Anfrageausführung ist nur eine weitere Möglichkeit einen direkteren Einfluss auf die „Richtung“ der Iterationsschritte zu nehmen.

Einfache visuelle Feedback-Verfahren sind aufgrund der menschlichen Eigenschaft visuelle Information schnell aufzunehmen aber Information nur langsam abgeben zu können (siehe Kapitel 3.2), sind für eine schnelle iterative Suche offensichtlich die geeignetsten. Es sollte aber immer die Option offen gehalten werden, auch partiell detaillierte Angaben machen zu können um immer vorhandenes Detailwissen direkt in die Anfrage einbringen zu können. Bei der Präsentationsauswahl der Ergebnismenge ist die rechenaufwändige mehrdimensionale Visualisierung der Datenbankinhalte die Geeignenste, da sie die Zusammenhänge am besten darstellt.

Eine weitere Option die Retrieval-Systeme erfüllen sollten ist die Erweiterbarkeit und Austauschbarkeit von einzelnen Komponenten des Anfragesystems, so können Neuentwicklungen schnell ergänzt werden und Verbesserte bzw. für eine spezielle Umgebung geeignetere oder angepasste Retrieval-Einheit ausgetauscht werden.

Eine Grundaussage zu den Ansätzen ist, dass es keinen idealen und allgemeingültigen Ansatz für das Retrieval auf Bildern gibt. Vielmehr ist eine geeignete Kombination aus mehreren Ansätzen und Bildbeschreibungen der vielversprechendste Weg zu einem effizienten Retrieval-System zu gelangen.

Kapitel 7

Zusammenfassung

Ziel der Arbeit war es, einen Überblick über Ansätze im Bereich der iterativen Anfragesprachen in Multimedia-Datenbanken zu erstellen. In den Grundlagen wurde dazu ein kurzer Einblick in die MMDBS gegeben. Besonders wurde dabei auf die Darstellung von Bildern im System eingegangen und die möglichen Repräsentationsformen der Bildinhalte vorgestellt. Um die Suche auf Grund der Ähnlichkeit von Bildern zu verwirklichen wurden zu den Repräsentationsformen der Bildinhalte mögliche Verfahren zur Bestimmung der Ähnlichkeit aufgezeigt. Für das merkmalsorientierte Content-Based-Image-Retrieval wurde ein formalisiertes Modell vorgestellt.

Aufgrund der Verschiedenheit der vorgestellten Beschreibungsmechanismen für Bildinhalte werden in der Literatur unterschiedliche Ansätze für die Beschreibung durch anfragen in MMDB verfolgt. In der Arbeit wurden dazu drei Ansätze für Attribute, Freitext, Inhaltsobjekte und Merkmale der Bildinhaltsbeschreibung verfolgt. Zum einen Daten- und Text-Retrieval, die für die Suche auf Attributen und Freitext eingesetzt werden. Zum Anderen wird ein spezielles Bild-Retrieval eingesetzt, das mit Inhaltsobjekten arbeitet oder angelehnt an Text-Retrieval den Merkmal-Raum zur Beschreibung und Suche von Bildern nutzt.

Die Visualisierung der Anfragesprache und eine geeignete Unterstützung der erwarteten Benutzeraktionen wurden als Teilbereich im Kapitel der Anfrageformulierung behandelt. Zum einen werden wegen unvollständigen Bildbeschreibungen im System und zum anderen wegen nur vagen Vorstellungen des Nutzers mehrere Iterationsschritte zu einem Anfrage-Prozess kombiniert, die ein Annähern an das Ergebnis realisieren. In der Arbeit werden dazu mehrere Ansätze vorgestellt, die einzelne Such-Strategien zugeordnet werden. Mit Browsen, manueller Anfrageformulierung, Relevanz- und Kontext-Feedback stehen vier Suchstrategien zur Verfügung, die unterschiedliche Vorgehensweisen bei der Suche nach Bildinhalten verwirklichen.

Im Kapitel 5 wurden einige Anfragesysteme vorgestellt, welche die erwähnten Ansätze für Bildbeschreibung und iterative Anfragen verwirklicht haben. In der Auswertung im darauffolgenden Kapitel wurden die Systeme miteinander verglichen und bezüglich der Ansätze betrachtet. Anhand der, in der Arbeit aufgezeigten Möglichkeiten für die Itera-

tion von Anfragen und unter einbeziehen der betrachteten Anfragesysteme, wurde eine Klassifikation erarbeitet die als Orientierung und Überblick für die gefundenen Ansätze angesehen werden kann. Mit einer abschließenden Diskussion wurden die Erkenntnisse bezüglich der iterativen Such-Strategien zusammengefasst und Anregungen für mögliche Weiterentwicklungen gegeben.

Literaturverzeichnis

- [ACL97] Andreasen, T.; Christiansen, H.; Larsen, H. L. (Hrsg.): *Flexible query answering systems*. Kluwer Academic Publishers, 1997.
- [Bim99] Bimbo, A. D.: *Visual information retrieval*. Kaufmann, 1999.
- [BMSW01] Bhm, K.; Mlivonic, M.; Schek, H.-J.; Weber, R.: Fast evaluation techniques for complex similarity queries. In *In Proceedings of the International Conference on Very Large Databases (VLDB '2001)*. Rome, Italy, September 2001.
- [Bos00] Bosshard, J.: An open and powerful relevance feedback engine for content-based image retrieval. Master's thesis, Eidgenössische Technische Hochschule Zürich, 2000.
- [BYRN99] Bazea-Yates, R.; Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison Wesley, Harlow [u.a.], 1999.
- [CMRB01] Carpineto, C.; Mori, R.; Romano, G.; Bigi, B.: An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, Band 19, Nr. 1, S. 1–27, 2001.
- [CTDB98] Chen, J.; Taskiran, C.; Delp, E.; Bouman, C.: Vibe: A new paradigm for video database browsing and search. In *Content-based Access to Image and Video Libraries*, S. 96–100. Santa Barbara, California, 1998.
- [FS95] Fowler, S. L.; Stanwick, V. R.: *The GUI style guide*. AP Professional, Boston [u.a.], 1995.
- [Gap95] Gappa, U.: *Grafische Wissensakquisitionssysteme und ihre Generierung*. Dissertation, Universität Karlsruhe, Sankt Augustin, 1995.
- [Guo99] Guojun, L.: *Multimedia database management systems*. Artech House, Boston, 1999.
- [HS95] Heuer, A.; Saake, G.: *Datenbanken: Konzepte und Sprachen*. International Thomson Publishing, Bonn, erste. Auflage, 1995.

- [HS99] Heuer, A.; Saake, G.: *Datenbanken: Implementierungstechniken*. MITP-Verlag, Bonn, erste. Auflage, 1999.
- [Jan93] Jansen, U.: *Werkzeuge und Methoden intelligenter Benutzeroberflächen*. Dissertation, Bergischen Universität-GH, 1993.
- [JGK97] Jin, J. S.; Greenfield, H.; Kurniawati, R.: Visual data processing in multimedia systems. In Leung, C. (Hrsg.): *Visual information systems*, S. 27–39. Springer, 1997.
- [JKX96] Jin, J. S.; Kurniawati, R.; Xu, G.: A scheme for intelligent image retrieval in multimedia databases. *Journal of Visual Communication & Image Representation*, Band 7, S. 369–377, 1996.
- [JKXB01] Jin, J. S.; Kurniawati, R.; Xu, G.; Bai, X.: Using browsing to improve content-based image retrieval. *Journal of Visual Communication & Image Representation*, Band 12, Nr. 2, S. 123–135, 2001.
- [Lan94] Langmann, R.: *Graphische Benutzerschnittstellen: Einführung und Praxis der Mensch-Prozess-Kommunikation*. VDI-Verlag, Düsseldorf, 1994.
- [LCHH97a] Li, W.-S.; Candan, K. S.; Hirata, K.; Hara, Y.: Ifq: a visual query interface and query generator for object-based media retrieval. In *International Conference on Multimedia Computing and Systems (ICMCS '97)*, 1997.
- [LCHH97b] Li, W.-S.; Candan, K. S.; Hirata, K.; Hara, Y.: Facilitating multimedia database exploration through visual interfaces and perpetual query reformulations. In Jarke, M.; Carey, M. J.; Dittrich, K. R.; Lochovsky, F. H.; Loucopoulos, P.; Jausfeld, M. A. (Hrsg.): *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, S. 538–547. Morgan Kaufmann, Aug 1997.
- [LE00] Lienhart, R.; Effelsberg, W.: Automatic text segmentation and text recognition for video indexing. *Multimedia Systems*, Band 8, Nr. 1, S. 69–81, 2000.
- [MMS⁺00a] Mller, H.; Mller, W.; Squire, D. M.; Marchand-Maillet, S.; Pun, T.: Learning features weights from user behavior in content-based image retrieval. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Workshop on Multimedia Data Mining MDM/KDD2000)*. Boston, MA, USA, aug 20-23 2000.
- [MMS⁺00b] Mller, H.; Mller, W.; Squire, D. M.; Marchand-Maillet, S.; Pun, T.: Lernen von merkmalsgewichtungen beim inhaltsbasierten suchen in grossen bilddatenbanken (content-based image retrieval). In *Proceedings of the Fachwissenschaftlicher Informatik-Kongress*. Bad Schussenried, Germany, oct 2000.

- [OS95] Ogle, V. E.; Stonebraker, M.: Chabot: Retrieval from a relational database of images. *IEEE Computer*, Band 28, Nr. 9, S. 40–48, 1995.
- [OSAH98] Olston, C.; Stonebraker, M.; Aiken, A.; Hellerstein, J. M.: Vqing: Visual interactive querying. In *Proceedings/1998 IEEE Symposium on Visual Languages, September 1-4, 1998, [Halifax], Nova Scotia, Canada/sponsored by IEEE Computer Society Technical Committee on Multimedia Computing*, S. 162–169. Los Alamitos, Calif. [u.a.]: IEEE Computer Society, sep 1998.
- [OXO98] Oria, V.; Xu, B.; Ozsu, M. T.: Visualmoql: A visual query language for image databases. In *VDB*, S. 186–191, 1998.
- [Pas99] Paskamp, M.: Vergleichende analyse von anfragesprachen. Master's thesis, Universität Magdeburg, 1999.
- [RGT97] Rubner, Y.; Guibas, L. J.; Tomasi, C.: The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA Image Understanding Workshop*, S. 661–668. New Orleans, LA, Mai 1997.
- [RHM97] Rui, Y.; Huang, T.; Mehrotra, S.: Content-based image retrieval with relevance feedback in mars, 1997.
- [RHM98] Rui, Y.; Huang, T. S.; Mehrotra, S.: Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, S. 25–36, 1998.
- [Rij79] Rijsbergen, C. J. V.: *Information retrieval*. Butterworth, London, 2.. Auflage, 1979.
- [Rob97] Robertson, S. E.: Overview of the okapi projects. *The journal of documentation*, Band 53, Nr. 1, S. 3–7, jan 1997.
- [RT01] Rubner, Y.; Tomasi, C.: *Perceptual metrics for image database navigation*. Kluwer Acad. Publ., 2001.
- [San99] Santini, S.: Exploratory interfaces for visual information systems. In *Proceedings of Vision Interface*. Trois Rivieres, Quebec, CA, May 1999. (in press).
- [SB90] Salton, G.; Buckley, C.: Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, Band 41, S. 288–297, 1990.
- [Sch97] Schäuble, P.: *Multimedia information retrieval: content-based information retrieval from large text and audio databases*. Kluwer Acad. Publ, Boston, 1997.

-
-
- [Sch00] Schneider, J.: Entwicklung einer bild-retrieval-komponente für ein multimediales ir-system. Master's thesis, Universität Dortmund, März 2000.
- [SGJ01] Santini, S.; Gupta, A.; Jain, R.: Emergent semantics through interaction in image databases. *Knowledge and Data Engineering*, Band 13, Nr. 3, S. 337–351, 2001.
- [SJ99a] Santini, S.; Jain, R.: User interfaces for emergent semantics in image databases. In *In Proceedings of the 8th IFIP Working Conference on Database Semantics (DS-8)*, 1999.
- [SJ99b] Santini, S.; Jain, R.: The 'El Niño' image database system. In *Proceedings of the IEEE International Conference on Multimedia Systems and Computing*. Florence, Italy, June 1999.
- [SJ00] Santini, S.; Jain, R.: Integrated browsing and querying of image databases. *IEEE Multimedia*, Band 7, Nr. 3, S. 26–39, 2000.
- [SMMP00] Squire, D. M.; Müller, W.; Müller, H.; Pun, T.: Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters (Selected Papers from The 11th Scandinavian Conference on Image Analysis SCIA '99)*, Band 21, Nr. 13-14, S. 1193–1198, 2000.
- [SST97] Saake, G.; Schmitt, I.; Türker, C.: *Objektdatenbanken - Konzepte, Sprachen, Architekturen*. International Thomson Publishing, Bonn, 1997.
- [Sub98] Subrahmanian, V. S.: *Principles of multimedia database systems*. The Morgan Kaufmann series in data management systems. Kaufmann, 1998.
- [Thi94] Thies, M. A.: *Planbasierte Hilfeverfahren für direkt-manipulative Systeme : Erkennung, Vervollständigung und Visualisierung von Interaktionsplänen*. Dissertation, Universität Stuttgart, 1994.
- [vC93] Campenhausen, C. v.: *Die Sinne des Menschen: Einführung in die Psychophysik der Wahrnehmung*. Thieme, 1993.
- [WI98] Weiss, S. M.; Indurkha, N.: *Predictive data mining: a practical guide*. Morgan Kaufmann, 1998.
- [Zwa99] Zwahr, A. (Hrsg.): *Meyers grosses taschenlexikon*. Meyers Lexikonredaktion, 1999.

Selbständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Magdeburg, den 28. März 2002

Anke Schneidewind

Thesen

1. Die Subjektivität der menschlichen Wahrnehmung kann nicht modelliert werden. Es ist deshalb kein effizientes und allgemeingültiges Retrieval-System ohne Iteration möglich.
2. Mit Lernverfahren lassen sich in kurzer Zeit allgemeine und benutzerspezifische Verbesserungen in Retrieval-Systemen erreichen.
3. Die Vielfalt der möglichen Beweggründe einer Suche nach Bildern erzwingt ein flexibles Retrieval-System, das an die entsprechende Problemstellung angepasst werden kann.
4. Eine einzelne Suchstrategie ist in der Regel nicht ausreichend, erst durch eine geeignete Kombination aus mehreren Suchstrategien lassen sich effiziente Systeme entwickeln.
5. Je besser dem Bediener des Retrieval-Systems Einblick in die Zusammenhänge des Datenbank-Systems und der Datenbankinhalte gewährt wird, desto sicherer kann ein befriedigendes Ergebnis erreicht werden oder die Gewissheit erlangt werden, dass das Gesuchte Bild nicht in der Datenbank ist.
6. Die Lücke zwischen der Menschlichen Wahrnehmung und Repräsentation im Rechner kann für allgemeine, unspezialisierte Retrieval-Systeme am einfachsten mit der iterativen Ausführung von Anfragen geschlossen werden.
7. Da bei der Suche nach Bildern oft nur eine vage Vorstellung vom Inhalt existiert ist es unabhängig vom System dem Nutzer gar nicht möglich eine Anfrage so zu formulieren, dass in einem ersten Schritt das geeignete Bild gefunden werden kann und somit Iteration z.B. in Form von Navigations-Strategien notwendig ist.