Ähnlichkeitsanfragen unter Verwendung von Bildmerkmalen und Text

Michael Mlivoncic, Klemens Böhm, Roger Weber *

1 Einleitung

Informationen sind nicht nur in reiner Textform, sondern zunehmend auch in multimedialen Formaten verfügbar. So bestehen im Internet HTML-Dokumente oft aus Texten und Bildern. Um den Informationsbedarf der Benutzer Rechnung zu tragen, gewinnt die Bildsuche zunehmend an Bedeutung.

Im allgemeinen werden heute zwei Verfahren zur Bildsuche eingesetzt. Zum einen kann man die Bilder mit Schlagworten annotieren und unter Anwendung von Text-Retrieval-Verfahren suchen. In manchen Anwendungsfeldern (z.B. bei Bildagenturen) existieren solche Annotationen, die den Inhalt eines Bildes in Schlagworten relativ genau beschreiben. Falls die zu durchsuchenden Bilder Bestandteile von Dokumenten sind, können die Annotationen auch aus der Umgebung der Bilder gewonnen werden: Informationen, wie beispielsweise Überschriften, Bildunterschriften oder benachbarte Textpassagen geben Aufschluß über den Inhalt und Kontext des Bildes.

Beim zweiten Ansatz extrahiert man (visuelle) Merkmale, wie z.B. Farbe oder Textur, aus den Bildern und sucht aufgrund dieser Merkmale nach ähnlichen Bildern zu einer Vorlage. Bei dieser inhaltsbasierten Bildähnlichkeitssuche verwendet man zum Teil hochdimensionale Merkmalsvektoren mit mehreren hundert Dimensionen. Wenn man beispielsweise ein Farbhistogramm als Merkmal verwendet, sind zwei Bilder (farblich) ähnlich, falls sie ähnliche Vektoren aufweisen. Die Vektoren aller Bilder bezüglich eines Merkmales befinden sich im gleichen Vektorraum, den man kurz Merkmalsraum nennt. Eine k-Nächste-Nachbar-Suche in diesem Merkmalsraum liefert die k ähnlichsten Bilder zu einem oder mehreren gegebenen Bildern, den Referenzbildern.

Von der Kombination der inhaltsbasierten Suche mit der Textsuche versprechen wir uns bessere Suchresultate. Dieser Beitrag beschreibt ein Modell und Techniken zur Auswertung kombinierter Anfragen mit Bild- und Textbestandteil. Zur effizienten Auswertung kommen dabei drei Indexstrukturen parallel zum Einsatz: Invertierte Listen (via SQL) und Signaturen für die Textbestandteile und das VA-File für die Suche in den hochdimensionalen Merkmalsräumen.

Im folgenden stellen wir unser Modell solcher kombinierter Anfragen dar. Anschliessend beschreiben wir effiziente Verfahren zur Bildsuche und zur Textsuche. Diese Verfahren sind dann die Grundlage des kombinierten Algorithmus zur Auswertung solcher Anfragen, den wir danach darstellen werden.

2 Anfragemodell

Wir haben unser Anfragemodell so gewählt, daß es komplex genug ist, um Anfragen aus mehreren Referenzbildern und mit unterschiedlichen Merkmalen pro Referenzbild darzustellen. Unser Modell entspricht dem Modell aus [BMSW01], wurde aber um die Verwendung von Annotationen und Texten als zusätzliche Merkmale der Bilder erweitert.

Eine Anfrage hat eine baumartige Struktur und besteht aus einem oder mehreren Referenzbildern, die auf der mittleren Ebene dargestellt sind. Darunter befinden sich die Merkmale, die nicht unbedingt für alle Referenzbilder gleich gewählt werden müssen.

* Institut für Informationssysteme, ETH Zentrum, CH-8092 Zürich {mlivoncic, boehm, weber}@inf.ethz.ch

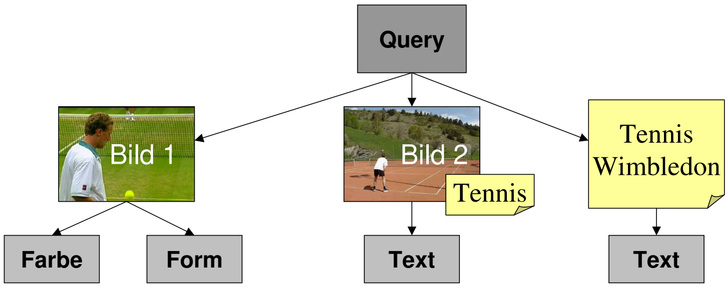


Abbildung 1: Beispielanfrage

Abbildung 1 gibt ein Beispiel für eine solche Anfrage: Wir suchen Bilder mit Tennisspielern auf dem Rasen. Wir haben ein solches Bild zur Verfügung und verwenden es daher als Referenzbild 1 bezüglich der Merkmale "Farbe" und "Form". Wir haben noch ein zweites Bild, das allerdings einen Spieler auf einem Sandplatz zeigt. Da dieses Bild einem Tennis-Dokument entnommen ist, können wir es als Referenz bezüglich des Textes verwenden. Zusätzlich kann man auch noch ein künstliches Referenzobjekt mit den Schlagworten "Tennis" und "Wimbledon" einführen.

Unser Modell berücksichtigt zusätzlich eine Gewichtung auf Merkmals- und Referenzbild-Ebene und ermöglicht auch die Verwendung von Relevanz-Feedback-Techniken. Auf diese Punkte werden wir aber in diesem Beitrag nicht weiter eingehen.

Sowohl eine Anfrage, wie sie durch die Wurzel des Baumes dargestellt wird, als auch eine Teilanfrage, die einem Teilbaum entspicht, liefert immer eine Distanz zu einem gegebenen Bild der Kollektion. Dabei bedeutet eine kleine Distanz eine große Ähnlichkeit und umgekehrt. Auf der Merkmalsebene ist dies die Distanz zwischen dem Vektor des entsprechenden Referenzbildes und dem Vektor des jeweils gegebenen Bildes. Auf den höheren Ebenen werden jeweils die Einzeldistanzen der Kind-Knoten zusammengefasst. Hierbei kann beispielsweise der (gewichtete) Durchschnitt zum Einsatz kommen. Auf der Wurzelebene erhalten wir die Gesamtdistanz der Anfrage zum gegebenen Bild.

3 Verwendete Verfahren

Im folgenden beschäftigen wir uns mit Verfahren, die wir bei der kombinierten Suche als "Bausteine" verwenden wollen. Es handelt sich um Techniken, zum Auswerten *atomarer* Anfragen. "Atomar" bedeutet in diesem Zusammenhang, daß man jeweils bezüglich eines einzelnen Referenzobjektes und Merkmales sucht, d.h. noch keine Kombination von Textsuche und Bildähnlichkeitssuche vorgesehen ist.

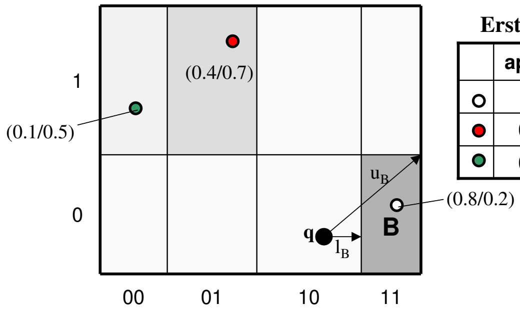
Das VA-File hat sich besonders bei hochdimensionalen Merkmalsdaten, wie sie bei der Bildersuche eingesetzt werden, als Datenstruktur bewährt, die eine effiziente Suche ermöglicht. Für das Textretrieval findet man in der Literatur eine Vielzahl von Methoden. Wir haben uns für zwei bewährte und effiziente Verfahren, Invertierte Listen und Signaturen, entschieden. Beide Verfahren lassen sich gut mit der Bildähnlichkeitssuche verbinden, wie wir in Abschnitt 4 erläutern werden.

3.1 Vector Approximation File (VA-File)

Das VA-File besteht aus zwei Dateien: die erste Datei enthält Vektoren, die zweite Quantisierungen dieser Originalvektoren. Das Erzeugen der Quantisierungen sowie das Suchverfahren sollen am zweidimensionalen Beispiel der Abbildung 2 erläutert werden.

Jede Dimension wird in mehrere Partitionen unterteilt. Jeder Punkt ist eindeutig einem Rechteck zugeordnet, das seiner Lage im Raum entspricht. Im Beispiel werden die Punkte durch eine Bitsequenz der Länge 3 approximiert.

Gesucht werden die k Bilder, deren Vektoren dem zu der Anfrage kleinsten Abstand haben. Im Folgenden beschreiben wir anhand des Beispieles, wie der zwei phasige Algorithmus, der auf der VA-File



Erste Phase

	approx	l_B	u_B	
○	11 0	0.1	0.4	★
●	01 1	0.35	0.85	★
●	00 1	0.50	0.9	

Zweite Phase

	data	dist
○	0.8 0.2	0.25

Abbildung 2: VA-File Beispiel

Datenstruktur beruht, den ähnlichsten Vektor ($k = 1$) ermittelt. Nähere Details findet man in [WSB98].

Phase 1: Die Quantisierungen werden sequentiell gelesen. Dabei wird für jede Quantisierung eine obere und eine untere Schranke für die Distanz zum Anfragevektor q berechnet (u_B und l_B in der Abbildung). Weil die untere Schranke des zweiten Punktes kleiner als die obere Schranke des ersten ist, könnte jeder dieser Punkte der Nächste Nachbar von q sein. Daher nehmen wir die ersten beiden Punkte in die Kandidatenmenge auf, d.h. sie werden in die zweite Phase übernommen. Der dritte Punkt wird hingegen verworfen, weil seine untere Schranke (0.50) weiter von q entfernt ist, als die bislang beste obere Schranke (0.4).

Phase 2: Die Kandidaten werden mit aufsteigender unterer Schranke betrachtet. Die exakte Distanz für den weißen Punkt ist 0.25. Weil die untere Schranken des nachfolgenden Punktes bereits größer ist, ist der weiße Punkt der gesuchte nächste Nachbar von q und der Suchvorgang kann beendet werden.

Da man in der ersten Phase für lediglich die Quantisierungen der Vektoren sequentiell liest, ergibt sich aufgrund des geringeren Datenvolumens auch bei hochdimensionalen Merkmalen noch eine gute Performanz.

3.2 Invertierte Listen

Invertierte Listen sind wie ein Stichwortverzeichnis eines Buches aufgebaut. Ein invertierter Index ist eine Liste von Termen, die in den indexierten Texten vorkommen. Zu jedem Term sind alle Dokumente[1] angegeben, in denen er vorkommt. Dies lässt sich effizient in einem relationalen DBMS umsetzen. Bei einer Suchanfrage nach mehreren Wörtern werden aus dem invertierten Index alle Dokumente entnommen, die eines oder mehrere dieser Wörter enthalten. Danach kann man noch für diese Dokumente den RSV-Wert berechnen, um die k relevantesten zu bestimmen. Für Boolsches Retrieval ist dies jedoch nicht erforderlich.

3.3 Signaturen

Signaturen sind Bitstrings, die die Merkmale (z.B. n-Gramme bei Text) eines Dokumentes repräsentieren. Eine Hashfunktion bildet die Merkmale dabei auf die einzelnen Bits der Signatur ab.

Für das Wort “Bild” ergäbe sich daher folgende Vorgehensweise: Zunächst werden die Merkmale “bi”, “bl”, “il”, “id” und “ld” extrahiert. Für diese Merkmale wird der Funktionswert für die angegebene Hashfunktion berechnet. Für “bi” erhält man beispielsweise: $hash(bi) = (Pos(b) \cdot 26 + Pos(i)) MOD\ 17 = (1 \cdot 26 + 8) MOD\ 17 = 0$. Die Signatur bekommt den Wert 1 an denjenigen Positionen, die dem Hashfunktionswert eines Merkmals entsprechen. Im Beispiel werden daher die Bits 0,3,15 und 7 gesetzt. Mit diesem Prinzip kann es passieren, daß mehrere Wörter auf dieselbe Signatur abgebildet werden.

Ein Suchvorgang liefert diejenigen Dokumente als Resultat, deren Signaturen am ähnlichsten zur Signatur der Anfrage sind. Dabei müssen alle Signaturen betrachtet werden. Für die Distanzberechnung zwischen Signaturen verwenden wir die Hamming Distanz und die Cover Distanz.

[1] Dokumente sind in diesem Zusammenhang die Textinformationen bzw. die Bilder, denen die Texte zugeordnet wurden

Die Hamming Distanz ist definiert als die Anzahl Bitpositionen, in denen sich zwei Bitstrings unterscheiden. Sie eignet sich vor allem für Anfragen, bei die Query viele Merkmale umfaßt, z.B. wenn ein Referenzdokument als Query vorgegeben wird. Spezifiziert die Query nur einzelne Merkmale, z.B. einzelne Worte, so ist die Cover Distanz vorzuziehen. Die Cover Distanz stellt fest, wieviele Merkmale der Anfrage im Dokument nicht vorhanden sind. Die weiteren Merkmale das Dokumentes werden nicht berücksichtigt.

4 Integrierter Suchalgorithmus

Im folgenden wollen wir uns mit der Frage befassen, wie ein kombiniertes Suchverfahren unter der Verwendung von Signaturen und Invertierten Listen zur Textsuche und dem VA-File zur Bildähnlichkeitssuche realisiert werden kann. Dabei soll das in Abschnitt 2 dargestellte Query-Modell zur Anwendung kommen. Grundsätzlich kommen bei der Auswertung zwei Varianten in Betracht:

Verteilter Ansatz: Bei diesem Ansatz geht man davon aus, daß das Suchsystem aus mehreren *unabhängigen* Suchkomponenten besteht, die sich an verschiedenen Orten befinden können ("Middleware-Annahme"). Eine Koordinatorkomponente organisiert den gesamten Suchablauf, schickt Teilanfragen an die Komponenten und kombiniert die Antworten zum Gesamtresultat. Jede Komponente liefert dazu einen Stream mit Resultaten bezüglich einer Teilanfrage in der Reihenfolge zunehmender Distanz. Beispielsweise würde man eine Komponente für die inhaltsbasierte Bildähnlichkeitssuche und eine Komponente für die textbasierte Suche mit Signaturen verwenden. Diese Vorgehensweise entspricht dem von Fagin vorgeschlagenen A_0-Algorithmus [Fag96].

Zentraler Ansatz: Die Daten, die für die Suche benötigt werden, sind dem Suchsystem direkt zugänglich. Die verschiedenen Suchdaten werden in je einer Datei gespeichert. Um beim sequentiellen Lesen auf alle Merkmalsdaten eines Objektes parallel zugreifen zu können, ist es erforderlich, daß sich die Informationen zu einem Bild in allen Dateien an derselben Position befinden und die Daten innerhalb den Dateien linear organisiert sind. Dies trifft sowohl für das VA-File, als auch für Signaturen zu. Im Gegensatz zum verteilten Ansatz betrachtet man von vornherein die Gesamtdistanzen der Objekte und muß deshalb am Ende keine Resultatlisten von Teilanfragen zusammenführen.

Es hat sich gezeigt, daß das Kombinieren der Teilergebnisse zu einem hohen Mehraufwand führt, insbesondere wenn sich die Rangfolge der Bilder in den unterschiedlichen Streams der Teilanfragen stark unterscheidet (vgl. [BMSW01]). Wir haben daher den zentralen Ansatz gewählt, der sich bereits für komplexe Ähnlichkeitsanfragen bewährt hat. Wir können die Anordnung unserer Daten frei wählen und die Middleware-Annahme trifft in unserem Fall ebenfalls nicht zu.

Signaturen als Merkmale: In [BMSW01] wurde die Kombination mehrerer inhaltsbasierter Bildmerkmale innerhalb einer Anfrage betrachtet. Die Merkmale wurden hierzu im VA-File gespeichert. Anders als bei Vektoren, sind Quantisierungen von Signaturen wenig sinnvoll. Daher existiert lediglich eine Datei mit den exakten Signaturen der Dokumente. Ein integriertes (zentrales) Suchverfahren soll wie die VA-File-Suche in zwei Phasen ablaufen, weil der filternde Effekt der ersten Phase maßgeblich für die Effizenz ist. Bei Signaturen verwenden wir statt der benötigten Schranken die *exakten* Distanzen bereits in der ersten Phase. Zudem verbessern gute Schranken die Filterwirkung der ersten Phase.

Hinzunahme Invertierter Listen: Zusätzlich kann es von Interesse sein, nur Bilder in der Resultatmenge zuzulassen, die mit bestimmten Schlagworten, z.B. "Gruppenfoto" versehen sind. Solche Schlagworte werden dann in einer SQL-Datenbank gespeichert sein. Mittels Boolschem Retrieval kann man feststellen, welche Bilder sich für die Resultatmenge qualifizieren. Es wäre sehr ineffizient, wenn man für jedes Objekt einzeln bei der Datenbank anfragt, ob das Objekt ein Stichwort enthält. Andererseits haben wir beim zentralen Ansatz die Bedingung, daß wir *nacheinander* die Objekte betrachten und dabei jeweils die Gesamtdistanz bestimmen. Aus Gründen der Effizienz sucht man zunächst mit einem SELECT Statement alle Objekte, die die (Boolsche) Anfrage erfüllen. Für diese Objekte vermerkt man in einer Tabelle bzw. einem Bitvektor, daß sie sich für die Resultatmenge qualifiziert haben. Wir bezeichnen dies auch als "prädikatbasierte Suche", da wir für jedes Objekt bestimmen, ob es als Resultat in Frage kommt oder nicht. Das "Prädikat" muß sich dabei nicht zwangsläufig auf das Vorhandensein von Schlagworten beziehen. Beispielsweise könnte man auch ein Prädikat definieren das angibt, ob ein Bild "Copyright-frei" ist. Die prädikatbasierte Suche wird allerdings nicht als zusätzliche Distanz in die Gesamtdistanz integriert. Stattdessen wird bei der Distanzberechnung für ein bestimmtes Objekt zuerst überprüft, ob dieses Objekt das angegebene Prädikat erfüllt. Ansonsten wird dieses Objekt nicht weiter betrachtet.

Der integrierte Suchalgorithmus ist eine Verallgemeinerung der VA-File-Suche und unterstützt die Auswertung von Anfragen, die unserem Modell aus Abschnitt 2 entsprechen. Dabei werden visuelle Bildmerkmale und Textinformationen, die auf Signaturen abgebildet wurden, in die Berechnung der Gesamtdistanz einbezogen, während die Invertierten Listen zur Boolschen Textsuche durch das Konzept der Prädikate integriert wird. Zusammenfassend ergibt sich der folgende Algorithmus:

Phase 1 (Geordneter Speicherzugriff): Lies aus allen verwendeten Featuredateien parallel und führe für jedes Objekt folgende Schritte aus:
Prüfe, ob dieses Objekt das definierte Prädikat erfüllt. Wenn nein, gehe zum nächsten Objekt. Berechne für jedes Referenzobjekt und jedes Feature die Schranken der Distanz zum aktuellen Objekt. Kombiniere diese Schranken pro Referenzobjekt und anschliessend für alle Referenzobjekte zu Gesamtschranken für die Anfrage. Entscheide anhand der Gesamtschranken, ob das betrachtete Objekt in die Kandidatenmenge aufgenommen wird (analog zum bisherigen Algorithmus).

Phase 2 (Zufälliger Speicherzugriff): Lies die Daten für jeweils ein Objekt der Kandidatenmenge sortiert nach aufsteigenden (unteren) Schranken. Solange die untere Schranke des nächsten Vektors kleiner als die k-kleinste bisher berechnete Distanz ist, wird für das nächste Element der Kandidatenmenge die (exakte) Gesamtdistanz zur Anfrage berechnet. Von den betrachteten Objekte werden die mit den k kleinsten Distanzen am Ende als Resultatmenge zuruckgeliefert.

Die Auswertung der Prädikate kann dabei asynchron erfolgen, d.h. während man in der ersten Phase alle Objekte sequentiell betrachtet, kann die Datenbank parallel dazu die Liste der Objekte zusammenstellen, die das Prädikat erfüllen. Es muss dabei gewährleistet sein, dass die Prädikatauswertung in der gleichen Reihenfolge erfolgt, in der die Objekte in den Featuredateien vorkommen. Der Suchalgorithmus muß ggf. warten, bis das Prädikat für das aktuelle Objekt ausgewertet ist.

5 Ausblick

Wir haben in diesem Beitrag ein Anfragemodell für kombinierte Bildähnlichkeitsanfragen mit Bildmerkmalen und Textbestandteilen beschrieben. Aufbauend auf diesem Modell haben wir einen integrierten Algorithmus zur Auswertung solcher Anfragen präsentiert. Der Algorithmus wurden – bis auf die asynchrone Auswertung der Prädikate – bereits implementiert. Die Leistungsfähigkeit muss jedoch noch experimentell ermittelt werden. Hierzu ist es zunächst erforderlich, eine Kollektion zu erstellen, die neben den Bildern auch Text bzw. Annotationen der Bilder enthält. Es wurden bereits Bausteine entwickelt, die die entsprechenden Informationen aus HTML-Seiten extrahieren können. Der bestehende Prototyp zur Bildähnlichkeitssuche muß weiterhin noch für die Verwaltung der Signaturdaten und der invertierten Listen vorbereitet werden.

Literatur

[BMSW01] K. Böhm, M. Mlivoncic, H.-J. Schek, and R. Weber. Fast Evaluation Techniques for Complex Similarity Queries. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, Rome, September 2001.

[Fag96] R. Fagin. Combining Fuzzy Information from Multiple Systems. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, volume PODS, pages 216–226, Montreal, Canada, June 1996.

[WSB98] R. Weber, H.-J. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, volume 24, New York, USA, August 1998.