Qualitätsverbesserung durch gewichtete Teilaspekte im Image Retrieval

Raoul Pascal Pein[1,2]

[1]Labor für Multimediale Systeme (MMLab),
Fakultät Technik und Informatik,
Hochschule für Angewandte Wissenschaften Hamburg,
Berliner Tor 7, 20099 Hamburg, Germany
pein_p@informatik.haw-hamburg.de
[2]Department of Informatics,
School of Computing and Engineering, University of Huddersfield,
Queensgate, Huddersfield HD1 3DH, United Kingdom

Abstract:
Diese Arbeit stellt eine Methode vor, content-based image retrieval Systeme im Allgemeinen zu verbessern. Die meisten derzeit verfügbaren Systeme basieren auf einer einzigen Technik, mit der Informationen aus Bildern gewonnen und verglichen werden. Jede einzelne Technik hat für sich genommen sowohl Vor- als auch Nachteile in Bezug auf die Ergebnisqualität. In der Regel decken sie wenige, bestimmte Bildaspekte ab, wie zum Beispiel Histogramme oder Formen. Um diese Einschränkungen zu umgehen, wird hier ein erweiterbares Framework vorgestellt, das mehrere verschiedene Aspekte auf einfache Weise in einem einzigen Image Retrieval System vereinigt. Ein Nutzer kann die vorhandenen Aspekte in der Suchanfrage seinen eigenen Bedürfnissen anpassen. So ist es möglich, query-by-sketch/-example mit Metainformationen wie Schlagworten oder Kategorien anzureichern.

1 Einleitung

Der weltweite Bestand an digitalen Bildern wächst immer stärker an. Dadurch gewinnt das *Content based Image Retrieval* (CBIR) immer mehr an Bedeutung, wie bereits 1999 von John P. Eakins und Margaret E. Graham [EG99] oder in [Sha86, RR00] beschrieben.

Viele der derzeit verfügbaren CBIR Programme bieten sowohl Nutzern als auch Anbietern nur wenige Konfigurationsmöglichkeiten. Sie konzentrieren sich entweder auf konventionelle, textbasierte Suche oder realisieren nur wenige fest implementierte Aspekte. Mit *Aspekten* sind hier alle Daten gemeint, die ein Bild beschreiben und für eine Suche verwendbar sind. Beispiele sind Histogramme, Formen, Texturen oder Schlagworte, die mit Sprachen wie MPEG-7 beschrieben werden können.

Weit verbreitete Web-Suchmaschinen bieten meist nur Schlagwortsuche oder Kategorien an. Beispiele für kommerzielles CBIR sind „Excalibur“ oder „IBM Image Miner“. In der Forschung sind Systeme wie „SIMBA“ [SSB01] der Universität Freiburg oder „PictureFinder“ [MHI03] der Universität Bremen zu finden.

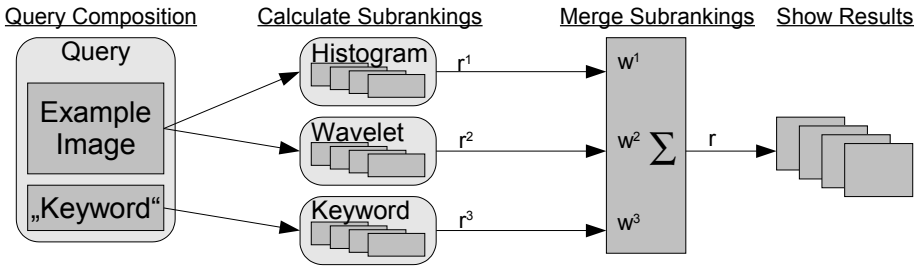

Abbildung 1: Kombiniertes Ranking

2 Anforderungen

Das Hauptziel der Arbeit liegt in der Erstellung eines einfach erweiterbaren und konfigurierbaren CBIR Systems. Es sollen viele verschiedene Suchaspekte in einem einzigen System vereint werden, wobei die Kontrolle über die Bedeutung der einzelnen Vergleichskriterien beim Benutzer liegt. Dazu muss die Gewichtung jedes Aspektes frei wählbar sein. Außerdem soll es möglich sein, die Ergebnisse im Voraus auf eine geforderte Mindestähnlichkeit zu beschränken, da im CBIR die Definition für einen Treffer unscharf ist.

Ein weiteres Ziel ist die spätere Erweiterbarkeit um zusätzliche Dateitypen, wie beispielsweise Texte. Hier können im Grunde die selben Mechanismen verwendet werden. Aus Performanzgründen ist allerdings ein alternatives Vorgehen ratsam, da die Ergebnismengen anders als beim CBIR in der Regel scharf abzugrenzen sind.

3 Systemdesign

Das grundlegende Design des hier vorgestellten Systems wurde aus [VT02] übernommen. Weiterhin können mehrere Aspekte zu einem gemeinsamen Ranking vereinigt werden.

Das System besteht aus lose gekoppelten Komponenten. Diese sollen bei Bedarf austauschbar sein. Als Grundlage dient eine einfach anzusteuernde Datenhaltung. Der erste Prototyp basiert auf einer Datenbank und wird später durch ein Dateisystem-Modul ergänzt. Darauf baut das Kernsystem auf, das die eigentliche Suche mit diversen Aspektmodulen durchführt. Die oberste Schicht wird von einem Client gebildet, der Suchanfragen an den Kern sendet und für die Darstellung der Ergebnisse zuständig ist. Der Prototyp besitzt 2 Varianten, einen Swing-Client und einen Servlet-Client. Für die Verwaltung der Daten und das Hinzufügen neuer Bilder wird ein Administrationswerkzeug benötigt.

Um eine hohe Erweiterbarkeit zu erzielen, muss jeder zu implementierende Aspekt auf einen eindimensionalen Ähnlichkeitswert (0.0 - 1.0) zwischen zwei Bildern zurückführbar sein. Auf diese Weise können die meisten bekannten Techniken zur Indexerstellung relativ schnell in das System integriert werden. Im Idealfall müssen lediglich der Extraktionsalgorihmus und eine Vergleichsoperation implementiert werden. Diese werden in einer Unterklasse der abstrakten FeatureVector-Klasse implementiert und im Betrieb über das Factory-Pattern instanziiert.

Für jeden implementierten Aspekt kann das System ein Ranking erstellen, indem sie den jeweiligen Anfragevektor mit den gespeicherten Vektoren vergleichen. Dadurch erhält man jeweils eine sortierbare Liste mit Ähnlichkeiten. Für das Endergebnis werden diese Listen zusammengeführt. Dabei kann eine einfache Gewichtung jedes einzelnen Aspektes in das Ergebnis einfließen (Abb. 1).

Die Ähnlichkeit r_x zwischen zwei Bildern - Anfrage und Bild x - wird dabei als gewichtete Summe der Teilrankings berechnet. Das Suchverhalten kann auf die Bedürfnisse des Nutzers angepasst werden, indem Gewichte und Toleranzgrenzen in der Anfrageformulierung mitgegeben werden, im Prototyp zwischen 0 und 1.

Die Anforderungen an die Datenhaltung sind für das Retrieval relativ speziell. Es fallen große Mengen gleichartiger Datensätze/Objekte an, die möglichst schnell durchsucht werden müssen, ohne den Arbeitsspeicher zu überlasten. Die Indexdaten werden daher so abgelegt, dass beim Retrieval jeder Aspekt sequentiell durchsucht werden kann und nur ein kleiner Teil der Gesamtdatenmenge im Speicher sein muss. Dieser sequentielle Zugriff wird über das Iterator-Pattern realisiert. Als Erweiterung ist ein gefilterter Iterator vorgesehen, der im Voraus irrelevante Daten aussortiert.

Ein bisher nur teilweise gelöstes Problem ist die Unterstützung variabler Eingabemasken. Da jeder implementierte Aspekt selbst definierte Daten erlaubt, kann es kein universelles Formular zur Queryerstellung geben. Insbesondere, wenn die Daten nicht automatisch aus einem Anfragebild gewonnen werden können, muss eine Möglichkeit zur freien Gestaltung des Formulars gegeben sein. Derzeit wird mit dynamisch eingebetteten JFrames experimentiert, die optional als Zusatz implementiert werden können.

Als alternative Schnittstelle werden Servlets erstellt, die die wichtigen Funktionen des Systems steuern können. Auch hier stellt sie die Frage, wie die Bedienoberfläche dynamisch an die Einzelaspekte angepasst werden kann.

4 Methodik

Die Entwicklung des Prototypen geschieht evolutionär. In den ersten Schritten wurde die Persistenzschicht zusammen mit dem Administrationstool entwickelt. Der Kern wurde vorerst rudimentär mit den wichtigsten Schnittstellen implementiert. Nach und nach wurde die Funktionalität ausgebaut.

Als erster Suchaspekt wurde eine einfache Schlagwortsuche auf Basis des Dateinamens realisiert. Später folgten 2 histogrammbasierte Varianten, die echtes CBIR erlauben. Dabei wurde die Persistenzschicht schrittweise um die zusätzlich benötigten Funktionen erweitert. Zum Abschluß des ersten Prototypen (Bachelorarbeit) wurde eine einfache Swing-Oberfläche implementiert.

In der 2. Phase (Projekt im Masterstudium) wird das System um neue Aspekte erweitert. Im Zuge der Implementierung einer Servlet-Schnittstelle werden Teile des alten System noch einmal überarbeitet.

In der anstehenden Masterarbeit sollen eine Evaluation der Benutzbarkeit und der Ergebnisqualität folgen.

5 Zusammenfassung

Diese Arbeit stellt einen einfachen Weg dar, beliebig viele Aspekte beim Image Retrieval zu verknüpfen. Das System ist nicht auf reines CBIR beschränkt. Weitere Aspekte wie Schlagworte oder Kategorien sind einfach integrierbar, wodurch die Indexerstellung allerdings nicht mehr vollständig automatisiert ablaufen kann.

Die Skalierbarkeit des Prototypen hängt im Wesentlichen von einer geeigneten Indexstruktur über den Daten ab, da sonst sequentiell gesucht werden muss. Strukturen wie Bäume grenzen den Suchraum zwar ein, aber können bei unscharfen Randbedingungen auch Treffer ignorieren.

6 Danksagung

Als Inspiration diente die Vorlesung „Multimediale Systeme“ von Prof. Dr. Renz an der HAW Hamburg, die unter anderem Multimediaarchive vorstellt. Der erste Prototyp wurde 2005 an der University of Huddersfield als Bachelorarbeit realisiert, die von Dr. Joan Lu und Prof. Dr. Renz betreut wurde. Für diesen Teil standen neben den regulären Vorlesungen je ein Semester für die theoretische und praktische Arbeit zur Verfügung.

Derzeit wird die Software in ein größeres, bisher rein akademisches Projekt unter der Leitung von Prof. Dr. Kai von Luck integriert, welches als Grundlage für die geplante Masterarbeit dient. Das Projekt findet inenrhalb eines Vorlesungssemesters des Masterstudiengangs statt.

Alle drei Professoren haben mich in verschiedenen Phasen des bisherigen Projektes betreut und in Gesprächen wichtige Anregungen gegeben.

Literatur

[EG99] J.P. Eakins und M.E. Graham. Content-based Image Retrieval. A Report to the JISC Technology Applications Programme. Bericht, University of Northumbria at Newcastle, Januar 1999.

[MHI03] A. Miene, Th. Hermes und G.T. Ioannidis. Graphical Image Retrieval with PictureFinder. In *DELOS Workshop on Multimedia Contents in Digital Libraries*, 2003.

[RR00] Monika Renz und Wolfgang Renz. Neue Verfahren im Bildretrieval. Perspektiven für die Anwendung. In R. Schmidt, Hrsg., *Proceedings der 22. Online-Tagung der DGI*, Seiten 102–128, Mai 2000.

[Sha86] S. Shatford. Analyzing the Subject of a Picture: A Theoretical Approach. *Cataloging and Classification Quarterly*, 6:39–62, 1986.

[SSB01] Sven Siggelkow, Marc Schael und Hans Burkhardt. SIMBA - Search IMages By Appearance. In *Pattern Recognition: 23rd DAGM Symposium, Munich, Germany, September 12-14, 2001. Proceedings*, Jgg. 2191/2001, Seite 9, 2001.

[VT02] Remco C. Veltkamp und Mirela Tanase. Content-Based Image Retrieval Systems: A Survey. Bericht UU-CS-2000-34, Department of Computing Science, Utrecht University, Oktober 2002.