

12. GI-Fachtagung für Datenbanksysteme in  
Business, Technologie und Web (BTW 2007)  
5. bis 9. März 2007 - Aachen, Germany  
<http://www.btw2007.de/>

## Untersuchung des Einflusses verschiedener Bild-Features und Distanzmaße im inhaltsbasierten P2P Information Retrieval

Soufyane El Allali      Daniel Blank      Martin Eisenhardt      Andreas Henrich  
Wolfgang Müller\*

vorname.nachname@wiai.uni-bamberg.de

**Abstract:** Gegenstand der vorliegenden Arbeit ist die feature-basierte Ähnlichkeits-suche in Bilddatenbeständen, die in einem P2P-Netz verteilt sind. Wir untersuchen dabei die Eignung verschiedener Featuresätze und Distanzmaße für die Nutzung in diesem Szenario. Hierzu beziehen wir uns primär auf PlanetP-artige P2P-Netze und vergleichen die in Abhängigkeit von der Anzahl der kontaktierten Peers erreichten Ergebnisse zunächst mit einem zentralen System mit gleichem Featuresatz und Distanzmaß. Ferner vergleichen wir unser System mit einer Erweiterung, die Indexdaten im P2P-Netz transferiert, sowie mit der Implementierung einer auf CANs basierenden verteilten Indexstruktur. Schließlich evaluieren wir das System auch mittels Relevanz-beurteilungen, die von Testnutzern gegeben wurden. Die Ergebnisse zeigen deutlich die unterschiedliche Eignung verschiedener Featuresätze und Distanzmaße für unser P2P-Szenario auf.

### 1 Einführung

Peer-to-Peer-(P2P)-Netzwerke entstehen durch den Zusammenschluss mehrerer autonomer, kooperierender Rechnerknoten, die ohne den Einsatz eines zentralen Servers interagieren. Solche Netze eignen sich besonders für die dezentrale Verwaltung großer Datenmengen bzw. für die gemeinsame Nutzung von Ressourcen. Im Vergleich mit einer klassischen Client-Server-Architektur erhöhen sie die Ausfallsicherheit des Gesamtsystems, da sie nicht von einem *single point of failure* abhängig sind.

Eine Vielzahl eingesetzter P2P-Systeme nutzt bei der Suche lediglich inhaltsbeschreibende Annotationen (sog. Tags) bzw. Teile des Dateinamens, um Medienobjekte zu finden<sup>1</sup>. Auch Flickr<sup>2</sup> unterstützt ausschließlich eine tag-basierte Bildsuche. Diese Herangehensweise greift zu kurz, da einerseits Informationen in Tags bewusst verfälscht werden können und andererseits Homonyme, Synonyme oder Sprachvarianten die Suche erheblich erschweren. Ein Großteil der Nutzer macht von der Möglichkeit ihre Bilder mit Tags an-

---

\*Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Projekts „Skalierbares, inhaltsbasiertes Retrieval von Text und Multimedia-Dokumenten in Peer-to-Peer Netzwerken“ gefördert.

<sup>1</sup> z.B. KaZaa, <http://www.kazaa.com>, letzter Abruf: 04.10.2006

<sup>2</sup> Flickr, <http://www.flickr.com>, letzter Abruf: 30.11.2006

notieren zu können auch keinen Gebrauch. Eigene Experimente haben gezeigt, dass von 666.909 zufällig ausgewählten Bildern bei Flickr nur etwa 60% mit Tags annotiert sind. Dies bedeutet, etwa 40% der Bilder bleiben für Anfragende komplett verborgen. Ein annotiertes Bild ist hierbei im Durchschnitt mit 4,0 Tags versehen. Es darf zumindest bezweifelt werden, ob dies ausreicht, um den Inhalt eines Bildes umfassend zu beschreiben. Aus diesem Grund ist ergänzend zur Schlagwort-orientierten Suche in vielen Anwendungen eine Suche auf Basis von Farb- oder Textureigenschaften sinnvoll [Ha06].

Im Rahmen dieser Arbeit vergleichen wir unser auf Zusammenfassungen beruhendes P2P-System mit einem für verteiltes Information Retrieval schwer zu erreichenden Benchmark, nämlich einem zentralen System mit gleichem Featuresatz und Distanzmaß. Dabei erachten wir ein P2P-System dann als „ideal“, wenn es die Top-20-Ergebnisse des zentralen Falls exakt reproduziert. Wir messen also die Ergebnisqualität des P2P-Systems relativ zum zentralen Fall (bzw. den zentral ermittelten Top-20-Bildern) in Abhängigkeit von der Anzahl der kontaktierten Peers und der Zahl der betrachteten Dokumente. Hierbei vergleichen wir die Ergebnisse für verschiedene Featuresätze und Distanzmaße.

Um die dabei erzielten Ergebnisse besser einschätzen zu können, vergleichen wir ferner die zunächst betrachteten P2P-Systeme, bei denen lediglich Zusammenfassungen im Netz verteilt werden (die Indexdaten zu den einzelnen Dokumenten verbleiben auf den Peers, auf denen die Dokumente selbst liegen), mit einer Variante, bei der selektiv auch Datensätze bzw. Indexdaten zwischen Peers transferiert werden, sowie einer auf CANs basierenden verteilten Indexstruktur [GYGM04].

Schließlich verwenden wir in weiteren Experimenten als Benchmark nicht mehr den zentralen Fall, sondern von Testnutzern gegebene Relevanzurteile zu den Bildern. Dabei wird betrachtet, wie viele der als relevant erachteten Bilder in Abhängigkeit von der Anzahl der betrachteten Peers im Gesamtergebnis enthalten sind. Damit soll untersucht werden, ob Feature-Distanz-Kombinationen, die sich zuvor als günstig für unser P2P-Szenario erwiesen haben, auch für den Benutzer zufriedenstellende Resultate liefern.

## **2 Ein Überblick über verwandte Ansätze**

Bei der inhaltsbasierten Bildsuche wird in vielen Anwendungsbereichen nach Methoden gesucht, die eine effiziente und effektive Ähnlichkeitssuche ermöglichen. Abhängig vom jeweiligen Anwendungsszenario eignen sich verschiedene Features und Distanzmaße unterschiedlich gut. Daher kommt der Beurteilung der Anwendbarkeit und Leistungsfähigkeit der Features und Distanzmaße eine zentrale Bedeutung zu. Für das zentralisierte Content-Based Image Retrieval (CBIR) wurde eine Vielzahl an Features und Distanzen vorgeschlagen (vgl. z.B. [Ha06]); ein kurzer Überblick wird am Ende dieses Abschnitts gegeben. Da wir uns mit der Ähnlichkeitssuche in P2P-Netzen beschäftigen, werden zunächst existierende Arbeiten auf diesem Gebiet rekapituliert.

Inhaltsbasierte Suchdienste in strukturierten P2P-Netzen wie bspw. Minerva [Be05] oder PRISM [Sa05] lassen sich auf Basis verteilter Hashtabellen (sog. DHTs) implementieren. Verschiedene Erweiterungen von CANs [TXM02, GYGM04] erlauben ebenso die

*k-nearest-neighbor*-Suche (*k*-NN-Suche).

Als Alternative zu verteilten Indexstrukturen wurden *routing-basierte Ansätze* vorgeschlagen. DISCOVER [KNS04] unterstützt Ähnlichkeitsanfragen auf Bilddaten, indem Anfragen gezielt an Cluster von Peers mit ähnlichen Zusammenfassungen weitergeleitet werden. Weitere Verbesserungen beruhen auf Replikation [SBR04] oder Super-Peer-Architekturen (z.B. [SSY04]).

Verschiedene Methoden zur Ressourcenauswahl bei der Suche in verteilten Datenquellen schlagen Nottelmann und Fuhr vor [NF03]. Die Parameter des Modells müssen in Abhängigkeit von Daten und Relevanzurteilen gelernt werden. Zwei dieser Methoden unterstützen neben der Suche nach Textdokumenten auch die Suche nach Bildern. Hierdurch unterscheiden sich diese Methoden von traditionellen Algorithmen wie CORI [CLC95] oder GLOSS [GGM95].

PlanetP [CAN02] adaptiert GLOSS für mittelgroße P2P-Netze. Jeder Peer kennt Zusammenfassungen zu den Dokumenten aller anderen Peers. Auf Basis der Zusammenfassungen erstellt der Anfragende ein Ranking der Peers, das festlegt, in welcher Reihenfolge die Peers während der Anfragebearbeitung kontaktiert werden. Dieser Ansatz vereinfacht die Bestimmung von Collection Wide Information, weil alle Peers alle Zusammenfassungen kennen; leider skaliert er nicht. Rumorama [MEH05] erzeugt eine Hierarchie von derartigen Netzen und erreicht dadurch Skalierbarkeit. Auf diese Weise lassen sich die in dieser Arbeit untersuchten Auswirkungen bei der Auswahl geeigneter Bild-Features und passender Distanzmaße von mittelgroßen PlanetP-Netzen auf große P2P-Netze übertragen.

Obwohl das Spektrum der im zentralisierten CBIR verwendeten Farb-Features sehr breit ist (eine Übersicht geben [Ha06]), fehlen Vergleiche verschiedener Bild-Features im Falle von P2P-Systemen. Deselaers et al. [DKN04] vergleichen verschiedene Features für den zentralen Fall. Sie stellen fest, dass die Wahl des jeweiligen Features sehr stark domänenabhängig ist. So sind Farbhistogramme (z.B. [SB91, ZLZ99, SC95]) unentbehrlich um bei Farbfotografien gute Ergebnisse zu erzielen. Da sich unsere Datenkollektion (s. Abschnitt 5) aus Fotografien verschiedenster Nutzer mit unterschiedlicher Herkunft, Hobbys, etc. zusammensetzt, verwenden wir in unseren Experimenten (s. Abschnitt 5) u.a. einige Varianten dieser Farbhistogramme.

Puzicha et al. [Pu99] vergleichen Distanzmaße für verschiedene Anwendungsszenarien. Einige Maße, die hierbei vielversprechende Resultate erzielen, sind in Abschnitt 4.3 erläutert und werden anschließend verwendet. Zusätzlich untersuchen wir weitere Distanzmaße, die sich im Bereich des Bildretrievals als nützlich erwiesen haben [HR05, Qu04].

### 3 Die Peer-to-Peer-Umgebung

Unsere Untersuchungen basieren auf PlanetP [CAN02]. Im Folgenden verwenden wir clusterzentroid-basierte Zusammenfassungen, die durch einen *Rumor Spreading*-Prozess im P2P-Netz verteilt werden. Der zugrunde liegende Mechanismus ist sehr gut in [CAN02] bzw. [MEH05] beschrieben, so dass wir uns in den folgenden Abschnitten auf eine knappe Darstellung beschränken können. Für unser Verfahren ist es von Bedeutung, dass sich die

Peers in periodischen Zeitabständen gegenseitig kontaktieren, um Zusammenfassungen auszutauschen und so das P2P-Netz aktuell zu halten.

#### **Zusammenfassungen für eine effiziente Ressourcenauswahl**

Die Daten eines Peers, d.h. die Bilder, die er bereit ist mit anderen zu teilen, werden durch sog. Cluster-Histogramme zusammengefasst. Das Cluster-Histogramm eines Peers ist ein Vektor, jede Komponente des Vektors repräsentiert einen bestimmten Cluster. Der Wert jeder Komponente des Histogramms gibt an, wie viele Dokumente eines Peers in einem gegebenen Cluster liegen.

Zur Gewinnung der Cluster könnte man etwa den  $k$ -Means-Algorithmus einsetzen. Frühere Arbeiten haben aber gezeigt, dass das Ranking der Peers nur unter gewissen Umständen von einem verteilten  $k$ -Means-Clustering der Dokumente profitiert [Ei06]. Daher werden im Rahmen dieser Arbeit zufällig 256 Dokumente aus der Dokumentenkollektion als Cluster-Zentroide ausgewählt, so dass diese die Verteilung der Datenpunkte widerspiegeln. Die Cluster-Zentroide determinieren, in welche Cluster die Dokumente eines Nutzers fallen. Diese Art der Zusammenfassung lässt sich leicht generieren. Beim Eintritt eines neuen Peers in das Netzwerk erhält dieser die 256 Cluster-Zentroide. Ein Peer berechnet auf Basis einer Distanzfunktion und der globalen Cluster-Zentroide die Zugehörigkeit seiner Dokumente zu den 256 Clustern. Danach versendet er die Zusammenfassung seiner Dokumente an alle anderen Teilnehmer im PlanetP-Netz.

#### **Ranking der Peers**

Der Mechanismus, der die Peers bzgl. der Anfrage rankt und determiniert, in welcher Reihenfolge die Peers kontaktiert werden, nutzt die Zusammenfassungen der einzelnen Peers sowie die zufällig ausgewählten Cluster-Zentroide. Unter drei verschiedenen Rankingmechanismen hat sich *StableSortRanker* als der vielversprechendste erwiesen [Ei06]. Dieser Mechanismus trifft eine Entscheidung auf Grundlage von  $L_{cl}$ , einer Liste, die die globalen Cluster-Zentroide, sortiert in aufsteigender Ordnung bezüglich ihrer Distanz zur Anfrage, enthält. Das erste Element dieser Liste entspricht immer dem Zentroid des Anfrageclusters, d.h. dem Zentroid des Clusters in dem die Anfrage selbst liegt. Peers mit vielen Dokumenten im Anfragecluster werden höher gerankt als Peers mit wenigen Dokumenten im Anfragecluster. Sofern Peer  $\alpha$  und Peer  $\beta$  die gleiche Anzahl an Dokumenten im momentan betrachteten Cluster haben, wählt *StableSortRanker* das nächste Element aus  $L_{cl}$  und vergleicht rekursiv Peer  $\alpha$  und Peer  $\beta$  bezüglich der Anzahl ihrer Dokumente in diesem Cluster, bis entweder eine Entscheidung getroffen werden kann oder das Ende von  $L_{cl}$  erreicht ist.

## **4 Feature-Extraktion und Ähnlichkeitsberechnung**

Bei der inhaltsbasierten Suche werden Medienobjekte in der Regel durch hochdimensionale Feature-Vektoren  $\vec{d} = (d_1, \dots, d_\delta)$  repräsentiert. Häufig verwendete Feature-Klassen sind hierbei Farbe, Textur und die Form von Objekten, die auf dem Bild zu sehen sind, sowie deren räumliche Lage. Anfragen werden häufig in Form einer *query by example* gestellt, bei der der Anfragende ein oder mehrere Anfragebilder auswählt, zu denen rele-

vante Bilder aus der Dokumentenkollektion gefunden werden sollen. Um nun die zu einer gegebenen Anfrage relevanten Dokumente finden zu können, muss auch das Anfragebild in Form eines Feature-Vektors  $\vec{q} = (q_1, \dots, q_\delta)$  repräsentiert sein.

In Abschnitt 5 werden Messungen basierend auf verschiedenen Feature-Distanz-Kombinationen vorgestellt. Im Folgenden werden zunächst die verwendeten Farb-Features vorgestellt. Ferner untersuchen wir in unseren Experimenten die Retrieval-Leistung bei Reduzierung der Dimensionalität der Feature-Vektoren, weshalb in Abschnitt 4.2 kurz auf die Hauptkomponentenanalyse eingegangen wird. Abschnitt 4.3 beschreibt die von uns untersuchten Distanzmaße, die als Basis für die Ähnlichkeitsberechnung zwischen Dokumenten und Anfrage dienen.

#### 4.1 Farb-Features

Die Analyse von Farbverteilungen bietet den Vorteil, dass sie größtenteils unabhängig vom Blickwinkel des Fotografen und der gewählten Auflösung ist. Als Farbmodelle werden oftmals der HSV-, der RGB-, sowie der CIE-Farbraum verwendet. Swain und Ballard [SB91] verwenden Farbhistogramme, um Dokumente in einer Bilddatenbank zu indexieren, wobei die Länge  $\delta$  der Histogramme  $\vec{d} = (d_1, \dots, d_\delta)$  durch die Farben des Farbmodells determiniert wird. Die Werte  $d_i$  entsprechen hierbei den relativen Vorkommenshäufigkeiten eines Farbwertes  $i$  im Bild. Um die Repräsentationen kompakt zu halten, bietet sich die Möglichkeit der Quantisierung. In dieser Arbeit werden zwei Arten globaler Farbhistogramme basierend auf dem HSV-Farbraum verwendet, die jeweils ein Farbhistogramm für das gesamte Bild berechnen, ohne es in Regionen aufzuteilen.

**HSV36q:** Zhang et al. [ZLZ99] schlagen eine Quantisierung in 36 Farben (sog. *bins*) vor, wobei die Quantisierung nicht gleichförmig erfolgt. Vielmehr wird die Hue-Komponente des HSV-Farbmodells in sieben Farben unterteilt, so dass diese den Farben, die in der chinesischen Sprache bekannt sind, entsprechen. Die Saturation/Value-Ebene des Farbraums wird in sechs Regionen unterteilt, wobei für  $V \leq 0,2$  unabhängig von S- und H-Wert ein Bin vorgesehen ist. Daraus resultieren  $7 \cdot 5 + 1 = 36$  Dimensionen. Acht der 36 Farben sind Grautöne, weshalb sich dieses Quantisierungsschema sowohl für Farb- als auch für Schwarzweißbilder eignet.

**HSV166q:** Ein anderes im CBIR häufig verwendetes Quantisierungsschema wurde von Smith und Chang [SC95] vorgeschlagen und quantisiert den HSV-Farbraum gleichförmig in 166 Bins; 18 Intervalle in der Hue-Dimension, drei in der Saturation-Dimension und drei in der Value-Dimension. Vier weitere Bins repräsentieren Grauwerte.

**LocHistHSV36q:** Lokale Farbhistogramme erfassen die Farbverteilungen bestimmter Regionen eines Bildes. So ist es bspw. möglich das Bild in eine bestimmte Anzahl  $n$  kleiner Bilder zu unterteilen und für diese  $n$  Farbhistogramme zu berechnen. In dieser Arbeit wird ein Bild in 16 rechteckige Regionen unterteilt und für diese wird jeweils ein HSV-Farbhistogramm berechnet, das den Farbraum wie beschrieben in 36 Bins quantisiert. Daraus resultiert ein 576-dimensionaler Feature-Vektor.

**COLCOHER:** Farbkohärenzvektoren (CCVs) [PZM96] klassifizieren ein Pixel eines Bins

als kohärent, wenn es Teil einer großen Region mit ähnlichen Farben ist. Ist dies nicht der Fall, fällt es in die Klasse inkohärent. Hierzu werden zwei Histogramme mit je 64 Dimensionen berechnet. Insgesamt ergeben sich so Feature-Vektoren mit 128 Dimensionen. CCVs vermeiden einen Vergleich von kohärenten Pixeln eines Bildes mit inkohärenten eines anderen Bildes und umgekehrt. Zunächst wird ein Pixelwert geglättet und durch den Durchschnittswert der acht benachbarten Pixel repräsentiert. Im Anschluss werden die Pixel anhand des RGB-Farbraumes gleichförmig in 64 Bins quantisiert (drei Farbkanäle mit je vier Farben) und abschließend klassifiziert. Maßgebend für die Einordnung eines Pixels als kohärent ist die Menge der gleichfarbigen, benachbarten Pixel, die größer als ein festgelegter Schwellenwert (5% der Pixelanzahl eines Bildes) sein muss.

**COLMOM:** Farbmomente [SO95] stellen Maßzahlen dar, welche verschiedene Farbverteilungen durch deren statistische Momente arithmetisches Mittel, Varianz und Schiefe beschreiben. Bei Verwendung des RGB-Modells und somit drei Farbkanälen resultiert daher pro Bild ein neun-dimensionaler Feature-Vektor, welcher ein Bild auf eine sehr kompakte Art und Weise repräsentiert und sehr einfach und schnell berechnet werden kann.

Neben Farb-Features haben wir ebenso verschiedene Textur-Features betrachtet. Da die Erkenntnisse hieraus sich größtenteils mit den bei der Analyse der Farb-Features gewonnenen decken, verweisen wir auf [AI06].

## 4.2 Hauptkomponentenanalyse

Hochdimensionale Feature-Vektoren stellen eine besondere Herausforderung für Clustering-Algorithmen und Distanzmaße dar. Eine Möglichkeit die Dimensionalität der Feature-Vektoren zu reduzieren bietet die Hauptkomponentenanalyse (PCA). Sie führt eine Hauptachsentransformation durch und versucht hierbei die für eine bestimmte Eigenschaft charakteristischen Merkmale zu extrahieren, um auf Basis dessen die Dimensionalität der Feature-Vektoren reduzieren zu können. Im Kontext von P2P-Systemen ist eine verteilte Variante der PCA anzuwenden, wie sie etwa in [BCL05] vorgestellt wird. Die Auswirkungen der Anwendung der PCA auf die Retrieval-Leistung wird in Abschnitt 5.1 betrachtet.

## 4.3 Distanzmaße

Typische  $k$ -NN-Anfragen suchen im Datenbestand nach den  $k$  Feature-Vektoren, die den geringsten Abstand zum Anfragevektor  $\vec{q}$  aufweisen. Der Abstand zweier Vektoren  $\vec{q}$  und  $\vec{d}$  wird hierbei mittels sog. Distanzmaße  $dist(\vec{q}, \vec{d})$  ermittelt. Da wir in Abschnitt 5 die Leistungsfähigkeit verschiedener Feature-Distanz-Kombinationen untersuchen, werden nun die von uns eingesetzten Distanzmaße vorgestellt. Das Spektrum der im CBIR verwendeten Distanzmaße ist breit, einen kurzen Überblick geben u.a. [RTG00].

**Minkowski-Distanz:**  $dist_{L_m}(\vec{q}, \vec{d}) = (\sum_i |q_i - d_i|^m)^{1/m}$

Im Bereich CBIR häufig verwendete Distanzmaße sind drei Ausprägungen der Minkowski-Distanz [SB91, SO95, RTG00]; es sind dies die Manhattan-Distanz  $dist_{L_1}$ , die Eukli-

dische Distanz  $dist_{L_2}$ , sowie die  $L_{max}$ -Distanz  $dist_{L_{max}}$ . Letztere resultiert aus obiger Formel für  $\lim_{m \rightarrow \infty} dist_{L_m}(\vec{q}, \vec{d})$  und entspricht dem Betrag der maximalen Differenz zwischen zwei Vektorkomponenten, die den gleichen Index besitzen.

**Fraktionale Distanz:**

Während klassische Minkowski-Distanzmaße für  $m \geq 1$  definiert sind, erweitern [AHK01] diese Definition auch für Werte  $0 < m < 1$  mit dem Ziel, ein gegenüber  $L_1$  oder  $L_2$  günstigeres Verhalten des Distanzmaßes zu erzielen. Howarth und Rüger [HR05] bestätigen in ihren Untersuchungen, dass ein Wert von  $m = 1/2$  meist bessere Retrieval-Ergebnisse als etwa  $dist_{L_1}$  oder  $dist_{L_2}$  liefert. Daher verwenden wir  $m = 1/2$  in unseren Experimenten.

**Symmetrische Kullback-Leibler Divergenz:**  $dist_{SKL}(\vec{q}, \vec{d}) = \frac{1}{2} \sum_i (q_i - d_i) \log \frac{q_i}{d_i}$   
 Bei der Kullback-Leibler Divergenz handelt es sich um ein Maß, das seinen Ursprung in der Informationstheorie hat. Es misst die minimale durchschnittliche Anzahl von verschwendeten Bits, wenn man einen Prozess mit Verteilung  $\vec{q}$  auf der Basis von  $\vec{d}$  kodiert. Da im Bereich des CBIR für zwei Feature-Vektoren  $\vec{d}'$  und  $\vec{d}''$  gelten soll  $dist(\vec{d}', \vec{d}'') = dist(\vec{d}'', \vec{d}')$ , verwenden wir eine symmetrische Variante der Kullback-Leibler Distanz.

**Kosinusmaß:**  $dist_{cos}(\vec{q}, \vec{d}) = (\sum_i q_i \cdot d_i) / (\sqrt{\sum_i q_i^2} \cdot \sqrt{\sum_i d_i^2})$   
 Vielfach wird im Bereich des Information Retrievals bei  $k$ -NN-Anfragen, speziell auch im CBIR [Qu04], der Kosinus des Winkels zweier Vektoren als Maß für die Unähnlichkeit zweier Dokumente eingesetzt. Je geringer dieser Winkel desto größer ist die Ähnlichkeit der durch die Vektoren repräsentierten Dokumente.

Bin-By-Bin-Distanzmaße, wie sie zuvor vorgestellt wurden, vergleichen die Feature-Vektoren komponentenweise. Dem liegt die Annahme zugrunde, dass die Komponenten von ihrer semantischen Bedeutung her orthogonal sind. Dies ist jedoch gerade bei Farbhistogrammen nicht gegeben, beispielsweise ist die Farbe hellrosa der Farbe rosa ähnlicher als der Farbe hellblau. Bei Bin-By-Bin-Distanzmaßen werden jedoch rosa und hellblau beide als gleich unähnlich zu hellrosa betrachtet. Mit Cross-Bin-Distanzmaßen können demgegenüber Zusammenhänge zwischen den Bins erfasst werden:

**Match-Distanz:**  $dist_{match}(\vec{q}, \vec{d}) = \sum_i |Q_i - D_i|$   
 Sowohl die Match-Distanz als auch die im Folgenden vorgestellte Kolmogorov-Smirnov-Distanz arbeiten mit kumulierten Histogrammen. Das kumulierte Histogramm  $\vec{D}$  eines Vektors  $\vec{d}$  ist definiert als  $(D_1, \dots, D_\delta)$ , wobei  $D_i = \sum_{j \leq i} d_j$ . Die Match-Distanz zweier eindimensionaler Vektoren ist demnach definiert als die Manhattan-Distanz ihrer kumulierten Histogramme.

**Kolmogorov-Smirnov-Distanz:**  $dist_{KS}(\vec{q}, \vec{d}) = \max_i |Q_i - D_i|$   
 Bei der Kolmogorov-Smirnov-Distanz handelt es sich um eine Maßzahl aus der Statistik, die definiert ist als  $L_{max}$ -Distanz zweier kumulierter Verteilungen.

## 5 Experimente

Die Experimente in Abschnitt 5.1 basieren auf 50 Simulationsläufen mit jeweils 100 Anfragen, wobei jeweils ein zufällig ausgewähltes Dokument aus der Kollektion als Anfra-

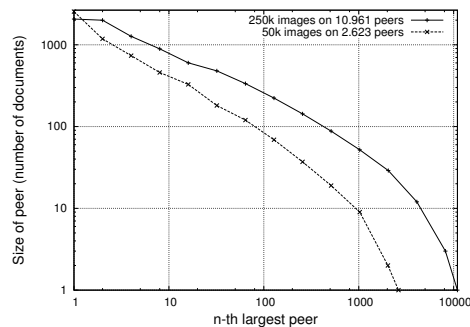


Abbildung 1: Verteilung der Peer-Größe bei 250k- bzw. 50k-Kollektion

ge verwendet wird. Insgesamt fließen demnach pro Feature-Distanz-Kombination 5.000 Anfragen ins Ergebnis ein. Falls nicht anders beschrieben, suchen wir in unseren Experimenten immer nach den, bezogen auf eine bestimmte Feature-Distanz-Kombination, 20 ähnlichsten Bildern (Top-20). Sie können als Retrieval-Ergebnis des zentralen Falles betrachtet werden, gegen das wir unser verteiltes P2P-Retrieval-System evaluieren.

Unsere Experimente basieren auf Bildern, die aus einem Crawl von Flickr.com resultieren. Flickr ist eine internetbasierte Foto-Community mit mehr als drei Millionen Nutzern. Diese können ihre Bilder einstellen, sie annotieren und auf diese Weise etwa ihre privaten Bilder mit anderen Benutzern teilen bzw. nach Bildern anderer Benutzer suchen. Wir nutzen in Abschnitt 5.1 bzw. 5.3 250.000 Bilder von 10.961 zufällig ausgewählten Flickr-Nutzern. Von jedem dieser Bilder werden die in Abschnitt 4.1 beschriebenen Farb-Features extrahiert. Da jeder Peer einen Flickr-Nutzer repräsentiert, simulieren wir hiermit insgesamt 10.961 Peers. In den Experimenten in Abschnitt 5.2 wird eine Kollektion aus 50.000 Bildern verwendet, die auf 2.623 Peers verteilt sind. Die Größenverteilung der Peers ist in Abb. 1 dargestellt.

### 5.1 Vergleich des verteilten mit dem zentralisierten Retrieval-Ergebnis

Im Folgenden werden zunächst die in Abschnitt 4.1 vorgestellten Farb-Features anhand der Manhattan-Distanz evaluiert. Die hier vorgestellten Messungen stellen dabei eine ausgewählte Teilmenge aller möglichen und von uns gemessenen Feature-Distanz-Kombinationen dar. Wir verwenden zunächst die Manhattan-Distanz, weil nicht alle Feature-Distanz-Kombinationen sinnvoll sind. So ist es etwa nicht sinnvoll, Vektoren, die statistische Kennzahlen enthalten (z.B. Farbmomente), mittels Match-Distanz zu vergleichen. Abbildung 2 zeigt auf der linken Seite den Einfluss verschiedener Farb-Features auf die Anzahl der Peers, die kontaktiert werden müssen, um einen möglichst großen Anteil der globalen Top-20-Dokumente aufzufinden. Farbmomente eignen sich demzufolge besonders gut für das Peer-Ranking. Um die 20 besten Dokumente zu finden, müssen weniger als 15 Prozent der Peers betrachtet werden. Auch bei einer Quantisierung des

HSV-Farbmodells in 36 Bins reicht es aus, weniger als 20 Prozent der Peers zu kontaktieren, um alle Top-20-Dokumente zu finden. (Zum Vergleich: Ein Orakel müsste 20 Peers betrachten, ein rein zufälliges Vorgehen fast alle Peers.) Tendenziell scheinen sich niedrig-dimensionale Feature-Vektoren besser für unsere P2P-Umgebung zu eignen, gerade auch vor dem Hintergrund, dass lokale Farbhistogramme mit 576 Dimensionen die schlechteste Performance aller betrachteten Farb-Features liefern.

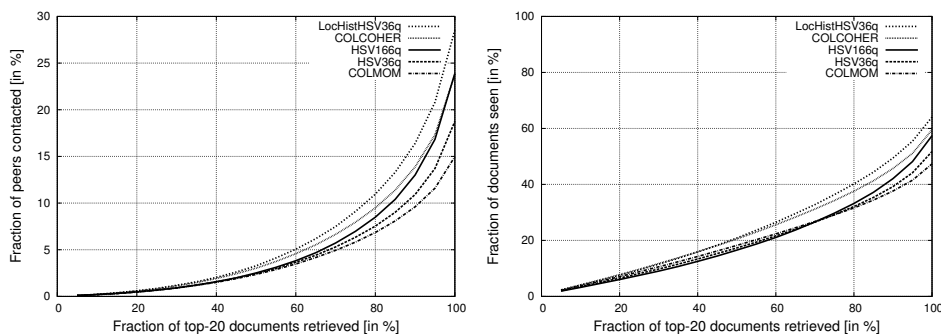


Abbildung 2: Farb-Features bzgl. zu kontaktierender Peers (li.) bzw. gesehener Dokumente (re.)

Abbildung 2 stellt auf der rechten Seite die Anzahl der betrachteten Dokumente bei Verwendung der Manhattan-Distanz dar. Die Messungen zeigen, dass die Betrachtung von weniger als 15% der Peers zur Bestimmung der „korrekten“ Top-20-Bilder im Falle der Farbmomente einer Betrachtung von über 45% der Dokumente entspricht. Dies zeigt, dass bei der Anfragebearbeitung tendenziell zunächst Peers kontaktiert werden, die viele Bilder bereitstellen. Man beachte aber, dass zur Ermittlung der Top-20-Bilder von jedem kontaktierten Peer nur seine lokalen Top-20-Bilder – bzw. deren Feature-Vektoren und IDs – übertragen werden müssen. Eine Betrachtung von über 45% der Dokumente bedeutet daher im Falle von 250.000 Dokumenten auf 10.961 Peers nicht den Transfer von über 112.500 Feature-Vektoren im P2P-Netz, sondern lediglich die Übertragung von maximal  $15\% \cdot 10.961 \cdot 20 = 32.883$  Feature-Vektoren.

In Abschnitt 4.3 wurden verschiedene Distanzmaße betrachtet, auf deren Basis die Ähnlichkeit zwischen Dokumenten berechnet werden kann. Abb. 3 (li.) zeigt die Anzahl der zu kontaktierenden Peers in Abhängigkeit von der Wahl eines bestimmten Distanzmaßes unter Verwendung von HSV36q. Dieses Feature wird hier verwendet, da für Farbmomente, obwohl diese bei Anwendung der Manhattan-Distanz die vielversprechendste Performance zeigen (vgl. Abb. 2), nicht alle Distanz-Kombinationen sinnvoll sind. Abb. 3 (li.) zeigt, dass sich die Match-Distanz und das Kosinusmaß ausgesprochen vielversprechend verhalten. Die Euklidische Distanz weist gegenüber der Manhattan-Distanz ein besseres Verhalten auf. Die schlechtesten Resultate liefert der Einsatz fraktionaler Distanzen.

Der Einfluss der Hauptkomponentenanalyse (Feature: HSV36q, Distanz: Manhattan) auf das Retrieval ist in Abb. 3 (re.) dargestellt. Erwartungsgemäß vergrößert sich der Aufwand zur Annäherung des zentralen Ergebnisses mit zunehmender Dimensionsanzahl. Je geringer die Anzahl der Dimensionen der Feature-Vektoren, desto weniger Peers müssen kontaktiert werden, um die Top-20-Dokumente aufzufinden. Die Ursache hierfür liegt wohl

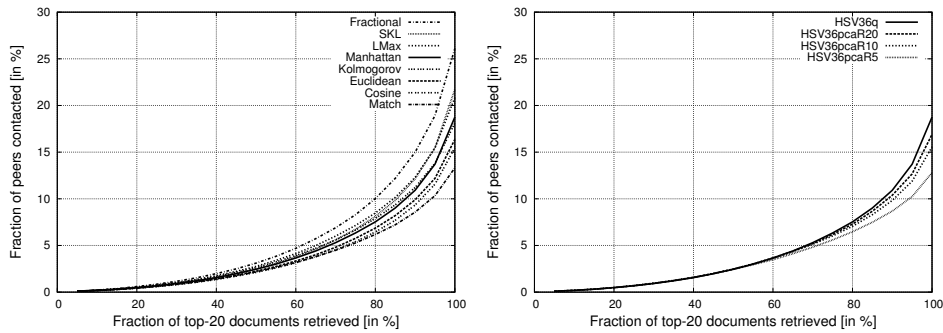


Abbildung 3: Einfluss der Distanzmaße (li.) bzw. PCA (re.) auf die Anzahl zu kontaktierender Peers (Feature: HSV36q)

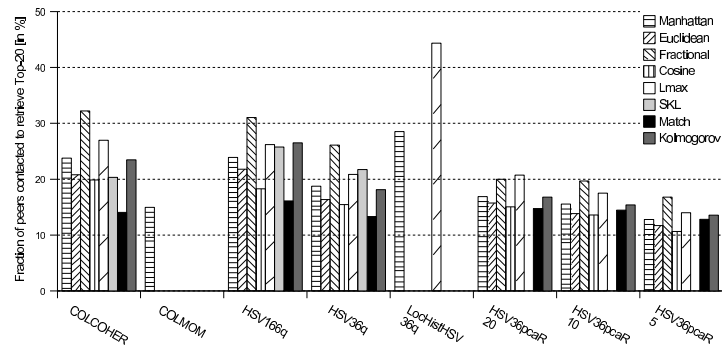


Abbildung 4: Anteil zu kontaktierender Peers, um alle Top-20-Bilder zu finden

im *Curse of Dimensionality*, der eine Ähnlichkeitssuche im hochdimensionalen Raum aufwändig macht [HAK00]. Gibt der Nutzer sich mit 60% der Top-20 zufrieden, so müssen jedoch bei allen Dimensionalitäten gleich viele Peers (etwa 3,5%) kontaktiert werden.

Abbildung 4 fasst die von uns durchgeführten Messungen zusammen. Wir stellen jeweils dar, wie viele Peers bei der Suche nach den jeweiligen Top-20-Bildern kontaktiert werden müssen. Die Match-Distanz zeigt hierbei bei den Features aus Abschnitt 4.1, für die alle Distanzen dargestellt sind, ein günstigeres Verhalten als die anderen Distanzmaße.

## 5.2 Retrieval-Experimente mit Content-Addressable Networks

Content-Addressable Networks [Ra01] waren die ersten mehrdimensionalen Indexstrukturen für P2P-Netze. Sie erlauben es Schlüssel/Wert-Paare abzulegen. Jeder Schlüssel ist hierbei ein  $\delta$ -dimensionaler Vektor  $\vec{v}$  aus dem  $\delta$ -dimensionalen Einheitshyperwürfel  $\vec{v} \in [0; 1]^\delta$ .

In einem CAN ist jeder Peer für eine achsenparallele, quaderförmige Region des Ein-

heitswürfels zuständig, d.h. alle Schlüssel/Wert-Paare  $(\vec{v}, x)$ , die im CAN indexiert sind, werden im für  $\vec{v}$  zuständigen Knoten gespeichert. Jeder Peer hält Verbindung zu denjenigen Peers, die für angrenzende Regionen zuständig sind. In ihrer ursprünglichen Form ermöglichen CANs effiziente (exakte) Membership-Anfragen. In einem  $\delta$ -dimensionalen CAN mit  $N$  Knoten müssen  $\mathcal{O}(\sqrt[\delta]{N})$  Routingschritte durchgeführt werden, um eine Membership-Anfrage zu beantworten<sup>3</sup>.

Ähnlich wie [TXM02, GYGM04] nutzen wir hier Erweiterungen von CANs für Ähnlichkeitsanfragen. Hierzu sind Designentscheidungen zu fällen, die im Wesentlichen die sogenannte *Splitstrategie* und die *Anfragebearbeitung* an sich betreffen.

Neue Peers gliedern sich in Standard-CANs sofort in das Netzwerk ein. In unseren Experimenten gehen wir zur besseren Vergleichbarkeit davon aus, dass sich neue Peers zunächst in einer Warteschlange einreihen. Beim Einfügen eines Schlüssel/Wert-Paares wird jeweils der für den neuen Vektor  $\vec{v}_n$  zuständige Peer  $p_{v_n}$  bestimmt. Enthält er mehr als  $n_{split}$  Schlüssel/Wert-Paare, so wird der  $p_{v_n}$  zugeordnete Teilraum aufgeteilt. Dazu wird ein neuer Peer  $p'$  aus der Warteschlange entfernt und in das CAN eingegliedert. Die Performance des CANs hängt nun entscheidend von der *Splitstrategie* ab, d.h. der Auswahl der Dimension, entlang derer die Zuständigkeitsregion von  $p_{v_n}$  in zwei Zuständigkeitsregionen aufgeteilt wird.

Bei einer exakten Anfrage ist sicher, dass sich der gesuchte Schlüssel  $\vec{q}$  in *exakt einem* Peer befindet, nämlich dem Peer, der für  $\vec{q}$  zuständig ist. Bei Ähnlichkeitsanfragen hingegen ist jedoch nicht einmal sicher, dass der zum Anfragevektor  $\vec{q}$  ähnlichste Vektor in dem Peer  $p_q$  zu finden ist, der für  $\vec{q}$  zuständig ist. Es sind also von  $p_q$  ausgehend Peers zu suchen, die die  $k$  nächsten Nachbarn enthalten. Wir haben sowohl die Splitstrategie als auch die Methode der Anfragebearbeitung so gewählt, dass CANs bezüglich der von uns gemessenen Eigenschaften möglichst gut abschneiden.

**Splitstrategie:** Anders als [GYGM04] verwenden wir eine datenabhängige Splitstrategie. In dem zu splittenden Knoten  $p_s$  werden entlang jeder Dimension Mittelwert und Varianz der in  $p_s$  enthaltenen Daten berechnet [HSW89]. Als Splitdimension  $i_s$  wird die Dimension mit der höchsten Varianz gewählt und die Kollektion entlang dieser Dimension so aufgeteilt, dass diejenigen Punkte  $\vec{v}$ , deren  $i_s$ -te Komponente  $v_{i_s}$  kleiner dem Mittelwert ist, in  $p_s$  verbleiben. Die anderen Schlüssel/Wert-Paare werden in den neuen Peer migriert.

**Anfragebearbeitung:** Die Anfragebearbeitung besteht aus zwei Schritten. Zunächst muss der für den Anfragevektor zuständige Peer  $p_q$  gefunden werden. Dann müssen von  $p_q$  ausgehend eventuell Nachbarn (Kandidaten-Peers  $p_c$ ) kontaktiert werden, die einen oder mehrere der  $k$ -NN von  $\vec{q}$  enthalten könnten. Hierzu verwaltet  $p_q$  eine Prioritätswarteschlange, die Punkte und Regionen, geordnet nach ihrer Entfernung zu  $\vec{q}$ , enthält [HS99]. Wird ein Punkt aus der Warteschlange gezogen, so ist er ein  $k$ -NN von  $\vec{q}$ . Wird eine Region gezogen, so könnte sie einen  $k$ -NN enthalten. Der zuständige Peer wird kontaktiert. Er sendet eine Liste von Punkten und Regionen an  $p_q$ . Dieser sortiert sie anschließend in die Prioritätswarteschlange ein. Das Verfahren wird solange fortgesetzt, bis die  $k$ -NN gefunden sind, oder die Warteschlange leer ist.

<sup>3</sup> Small-world-CAN-Varianten, die logarithmische Komplexität bieten, existieren, wie z.B. [GYGM04]. Für die von uns betrachteten hochdimensionalen Räume ist die hier zu erwartende Ersparnis allerdings nicht relevant.

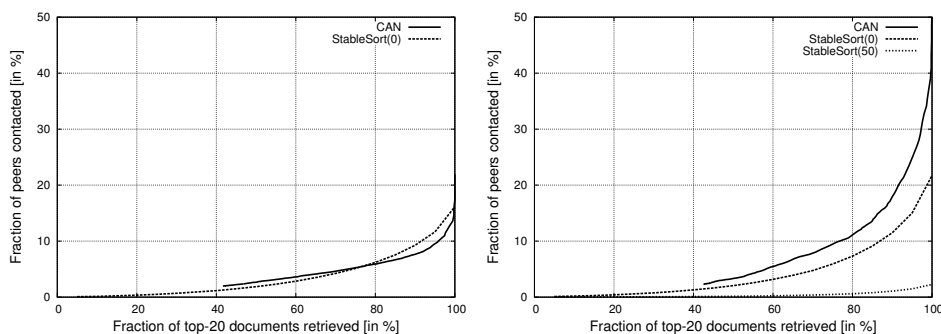


Abbildung 5: CAN-Implementierung vs. *StableSortRanker* (links: HSV36q, rechts: HSV166q)

Abbildung 5 zeigt auf der linken Seite einen Vergleich zwischen dem von uns bisher betrachteten *StableSortRanker* für PlanetP-artige Netze und unserer CAN-Implementierung für HSV36q unter Verwendung der Euklidischen Distanz. Sollen mehr als 80% der Top-20-Dokumente gefunden werden, verhält sich die CAN-Implementierung besser als *StableSort(0)*.

Bei Verwendung von HSV166q verhält sich *StableSort(0)* besser als CAN (Abbildung 5, rechte Seite). *StableSortRanker* kontaktiert dabei weniger Peers, um die gleiche Anzahl an Top-20-Dokumenten aufzufinden. In dieser Abbildung ist zusätzlich die in [Ei06] beschriebene Strategie des *Index Swappings* dargestellt. Bei Anwendung dieser Strategie, die es erlaubt, gezielt in begrenztem Umfang Indexdaten zwischen den einzelnen Peers auszutauschen, um auf diese Weise homogenere Datenverteilungen und damit prägnantere Zusammenfassungen zu ermöglichen, lässt sich die Leistungsfähigkeit von *StableSortRanker* noch einmal deutlich steigern. *StableSort(50)* visualisiert das Ergebnis, wobei jeder Peer 50-mal die Möglichkeit hatte, einen nicht zu seiner lokalen Datenkollektion passenden Indexdatensatz zu einem anderen, geeigneteren Peer zu transferieren. Dementgegen verbleiben bei *StableSort(0)* Indexdaten und zugehörige Dokumente auf dem gleichen Peer. Die Performance-Vorteile für unsere Methode ergeben sich hierbei daraus, dass sie die Verteilung der Daten (s. Abb. 1) über die Peers gezielt nutzt.

### 5.3 Experimente mit menschlicher Relevanzbeurteilung

Bisher haben wir Messungen durchgeführt, bei denen die im zentralen Fall erzielten Ergebnisse als Benchmark verwendet wurden. Um diese Betrachtungsweise zu ergänzen werden wir nun Experimente vorstellen, bei denen Relevanzurteile von Experten als Benchmark genutzt werden. Bei der Relevanzbeurteilung wird eine *Pooling*-Strategie (vgl. [JvR75]) eingesetzt. Der Pool setzt sich je Anfrage aus den Top- $N$ -Retrieval-Ergebnissen, die mit  $n$  verschiedenen Retrieval-Systemen ermittelt wurden, zusammen. Alle Dokumente dieses Pools werden von Experten nach Relevanz bzgl. der Anfrage beurteilt. Dokumente, die nicht im Pool enthalten sind, werden als irrelevant angesehen. Als Anfragen verwen-

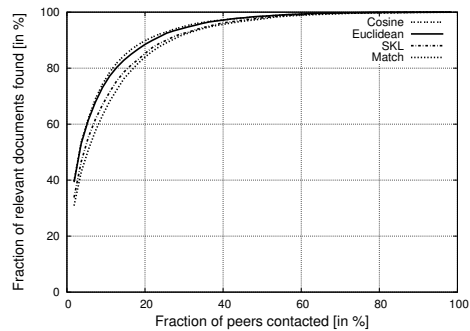


Abbildung 6: Kontaktierte Peers vs. relevante Bilder gefunden (Feature: HSV36q)

den wir 17 Anfragebilder, die nicht in der Kollektion enthalten sind. Diese besitzen eine semantische Aussagekraft wie etwa ein Feld voller Sonnenblumen oder ein rotes Automobil. Wir bilden 17 Pools aus Top-50-Anfragen mit Hilfe von je 27 verschiedenen Feature-Distanz-Kombinationen. Hieraus resultiert eine maximale Poolgröße von 1.350 Bildern pro Anfrage. Aus Gründen der Überlappung reduziert sich die Poolgröße im Durchschnitt auf 690 Bilder pro Anfrage. Die Bilder der Pools wurden von zwei Personen manuell evaluiert, wobei die Schnittmenge aus beiden Relevanzbeurteilungen als die Menge relevanter Dokumente angesehen wurde, mittels derer im Folgenden die Performance von *StableSortRanker* unter verschiedenen Feature-Distanz-Kombinationen evaluiert wird.

Die 17 Anfragen wurden jeweils 30-mal für verschieden gewählte Cluster-Zentroide zur Histogrammerstellung ausgeführt, da die zufällige Wahl der Cluster-Zentroide das Ergebnis beeinflusst. Aus der Menge der möglichen Feature-Distanz-Kombinationen wählen wir die signifikantesten Ergebnisse. Abbildung 6 zeigt den relativen Recall unseres P2P-Systems bei Verwendung von HSV36q. Dieses Feature zeigt bei der Evaluierung zusammen mit vielen Distanzen ein besseres Verhalten als andere Features. Außerdem erweist es sich in Abbildung 2 besser als etwa HSV166q. Es weist darüber hinaus in Abb. 3 (li.) in Kombination mit  $dist_{cos}$  und  $dist_{match}$  ein ähnlich gutes bzw. besseres Verhalten als die Kombination von Farbmomenten und Manhattan-Distanz in Abb. 2 (li.) auf. Diese Kombination aus  $dist_{L_1}$  und Farbmomenten, die bei der Messung in Abb. 2 (li.) sehr gut abschneidet, fällt jedoch bei der Betrachtung des relativen Recalls klar hinter die besten Ergebnisse, wie sie in Abbildung 6 dargestellt sind, zurück.

Insbesondere das Kosinusmaß sowie die Euklidische Distanz zeigen in Abb. 6 ein besseres Verhalten als die anderen Distanzen. Um etwa 80% aller relevanten Dokumente zu finden, werden bei Verwendung von  $dist_{cos}$  bzw.  $dist_{L_2}$  lediglich etwa 12% der Peers kontaktiert.

Insgesamt scheint die Kombination HSV36q im Zusammenspiel mit dem Kosinusmaß geeignet zu sein, da sie sowohl bei den Experimenten auf Basis der globalen Top-20-Dokumente als auch im Rahmen der Evaluierung mit Relevanzbeurteilungen mit die besten Ergebnisse liefert. Die Verwendung der Euklidischen Distanz scheint bei Betrachtung von Abbildung 3 (li.) und Abbildung 6 ebenfalls berechtigt.

## 6 Zusammenfassung und Ausblick

In dieser Arbeit haben wir das Verhalten verschiedener Bild-Features und Distanzmaße in PlanetP-artigen Netzen untersucht. Wir haben Vergleiche mit einem zentralen System, mit einer Erweiterung, die Indexdaten im P2P-Netz transferiert, sowie mit einem CAN-artigen Ansatz einer verteilten Indexstruktur angestellt. Abschließend haben wir das System hinsichtlich Relevanzbeurteilungen, die von Testnutzern gegeben wurden, evaluiert. Bezogen auf die Ergebnisse eines zentralen Systems als auch auf die von Nutzern gegebenen Relevanzbeurteilungen eignen sich bestimmte Feature-Distanz-Kombinationen hierbei besser als andere. Beim Vergleich mit einem CAN-artigen Ansatz schneidet unser System für 166-dimensionale Feature-Vektoren besser ab als für niedrigdimensionale Vektoren mit 36 Dimensionen. Auch hier erweist es sich aber als konkurrenzfähig.

In Zukunft werden wir andere Medientypen untersuchen (z.B. Text und Audio). Ebenso möchten wir die Lastverteilung im Netz optimieren. Momentan werden aufgrund des Rankings tendenziell eher die großen Peers besucht. Ein Ausgangspunkt weiterer Forschungsarbeiten ist die Suche nach einem geeigneten Abbruchkriterium, das festlegt, ab wann es sich nicht mehr lohnt weitere Peers zu kontaktieren.

### Literatur

- [AI06] S. El Allali et al. Farb-, Textur-Features und Distanzmaße für zusammenfassungsbasiertes P2P CBIR. Bericht, Lehrstuhl für Medieninformatik, Universität Bamberg, 2006.
- [AHK01] Charu C. Aggarwal, Alexander Hinneburg und Daniel A. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space. *LNCS*, 1973:420–434, 2001.
- [Be05] Matthias Bender et al. The Minerva Project: Database Selection in the Context of P2P Search. In *BTW'05*, 2005.
- [BCL05] J.Z. Bai, R. Chan und F. Luk. Principal Component Analysis for Distributed Data Sets with Updating. In *Workshop on Advanced Parallel Processing Technologies*, 2005.
- [CAN02] F. M. Cuenca-Acuna und T.D. Nguyen. Text-Based Content Search and Retrieval in ad hoc P2P Communities. Bericht DCS-TR-483, Dept. for CS, Rutgers University, 2002.
- [CLC95] J. Callan, Z. Lu und W. Croft. Searching distributed collections with inference networks. *SIGIR'05*, 1995.
- [DKN04] T. Deselaers, D. Keysers und H. Ney. Features for Image Retrieval – A Quantitative Comparison. In *Pattern Recognition, 26th DAGM Symposium*, 2004.
- [Ei06] M. Eisenhardt et al. Clustering-based Source Selection for Efficient Multimedia Retrieval in Peer-to-Peer Networks. *IEEE MIPR'06, San Diego, CA*, 2006.
- [GGM95] L. Gravano und H. Garcia-Molina. Generalizing GIOSS to vector-space databases and broker hierarchies. *VLDB'95, Los Altos, California*, 1995.
- [GYGM04] Prasanna Ganesan, Beverly Yang und Hector Garcia-Molina. One torus to rule them all: multi-dimensional queries in P2P systems. In *WebDB'04*, 2004.
- [Ha06] Alaa Halawani et al. Fundamentals and Applications of Image Retrieval: An Overview. *Datenbank Spektrum*, 18:14–23, August 2006.
- [HAK00] Alexander Hinneburg, Charu C. Aggarwal und Daniel A. Keim. What Is the Nearest Neighbor in High Dimensional Spaces? In *The VLDB Journal*, Seiten 506–515, 2000.

- [HR05] P. Howarth und S. M. Rüger. Fractional Distance Measures for Content-Based Image Retrieval. In *Europ. Conf. on IR Research*, 2005.
- [HS99] Gísli R. Hjaltason und Hanan Samet. Distance Browsing in Spatial Databases. *ACM Trans. Database Syst.*, 24(2):265–318, 1999.
- [HSW89] A. Henrich, H.-W. Six und P. Widmayer. The LSD tree: Spatial Access to Multidimensional Point and Nonpoint Objects. In *VLDB*, 1989.
- [JvR75] K. Sparck Jones und C. J. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Bericht, British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.
- [KNS04] I. King, C. Hang Ng und K. Cheung Sia. Distributed content-based visual information retrieval system on peer-to-peer networks. *ACM Trans. Inf. Syst.*, 22(3), 2004.
- [MEH05] W. Müller, M. Eisenhardt und A. Henrich. Scalable summary based retrieval in P2P networks. In *CIKM 2005*, 2005.
- [NF03] H. Nottelmann und N. Fuhr. Decision-theoretic resource selection for different data types in MIND. *ACM SIGIR Workshop on Distributed Information Retrieval*, 2003.
- [Pu99] Jan Puzicha et al. Empirical Evaluation of Dissimilarity Measures for Color and Texture. In *Int. Conf. on Computer Vision, Kerkyra, Griechenland*, 1999.
- [PZM96] G. Pass, R. Zabih und J. Miller. Comparing images using color coherence vectors. In *Int. Conference on Multimedia*, 1996.
- [Qu04] G. Qian et al. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In *ACM Symposium on Applied Computing*, 2004.
- [Ra01] Sylvia Ratnasamy et al. A scalable content-addressable network. In *Conf. on applications, technologies, architectures, and protocols for computer communications*, 2001.
- [RTG00] Y. Rubner, C. Tomasi und L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *Int. Journal of Computer Vision*, 40(2):99–121, 2000.
- [Sa05] O. D. Sahin et al. PRISM: indexing multi-dimensional data in P2P networks using reference vectors. In *13th Intl. Conf. on Multimedia*, 2005.
- [SB91] M. J. Swain und D. H. Ballard. Color indexing. *Int. Journal of Computer Vision*, 7(1):11–32, 1991.
- [SBR04] N. Sarshar, P. O. Boykin und V. P. Roychowdhury. Percolation search in power law networks: making unstructured peer-to-peer networks scalable. In *P2P 2004*, 2004.
- [SC95] J. Smith und S. Chang. Single color extraction and image query. *Proc. IEEE Int. Conf. on Image Proc.*, Seiten 528–531, 1995.
- [SO95] Markus A. Stricker und Markus Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, Seiten 381–392, 1995.
- [SSY04] H. Shen, Y. Shu und B. Yu. Efficient Semantic-Based Content Search in P2P Network. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):813–826, 2004.
- [TXM02] C. Tang, Z. Xu und M. Mahalingam. pSearch: Information Retrieval in Structured Overlays. In *1st Workshop on Hot Topics in Networks*, Princeton, NJ, 2002.
- [ZLZ99] L. Zhang, F. Lin und B. Zhang. A CBIR method based on color-spatial feature. *IEEE Region 10 Annual International Conference 1999*, Seiten 166–169, 1999.