Soufyane El Allali Daniel Blank Martin Eisenhardt
Andreas Henrich Wolfgang Müller

Farb-, Textur-Features und Distanzmaße für zusammenfassungsbasiertes P2P CBIR

– Technischer Bericht–
Lehrstuhl für Medieninformatik
Universität Bamberg

[1]Diese Arbeit wurde teilweise von der Deutschen Forschungsgemeinschaft gefördert.

Zusammenfassung

Gegenstand des vorliegenden Berichts ist die feature-basierte Ähnlichkeitssuche in Bilddatenbanken, 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 eines auf CANs basierenden Ansatzes einer verteilten Indexstruktur. Schließlich evaluieren wir das System auch hinsichtlich Relevanzbeurteilungen, die von Testnutzern gegeben wurden. Die Ergebnisse zeigen deutlich die unterschiedliche Eignung verschiedener Featuresätze und Distanzmaße für P2P-Szenarien auf.

1 Einführung

Peer-to-Peer-(P2P)-Netzwerke entstehen durch den Zusammenschluß mehrerer autonomer, kooperierender Rechnerknoten, die ohne den Einsatz eines zentralisierten Servers interagieren. Solche Netze eignen sich besonders für die dezentrale Verwaltung großer Datenmengen bzw. für die gemeinsame Nutzung von Ressourcen. Im Gegensatz zu einer klassischen Client-Server-Architektur erhöhen sie die Ausfallsicherheit des Gesamtsystems, da sie nicht von einem *single point of failure* abhängig sind. Die inhaltsbasierte Ähnlichkeitssuche auf Medienobjekten stellt für viele Anwendungsbereiche in zentralen Systemen einen wichtigen Dienst dar, der daher auch für P2P-Netze wünschenswert ist.

Eine Vielzahl der im WWW eingesetzten Peer-to-Peer-Systeme nutzt bei der Suche lediglich inhaltsbeschreibende Annotationen (sog. Deskriptoren) bzw. Teile des Dateinamens, um Medienobjekte wie bspw. Bilder des Bamberger Doms oder Musikstücke eines bestimmten Musikstils zu finden[2]. Diese Herangehensweise greift zu kurz, da einerseits Informationen bewusst verfälscht werden können und andererseits Homonyme, Synonyme oder Sprachvarianten die Suche erheblich erschweren.

Suchverfahren, die den Inhalt verschiedener Medienobjekte analysieren, umgehen diese Problematik. Sie nutzen Informationen, die aus dem Inhalt eines Medienobjektes extrahiert werden können (bei Bildern z.B. Farb-, Textur- oder Formeigenschaften) und repräsentieren auf diese Weise ein Medienobjekt in Form eines hochdimensionalen Feature-Vektors. Während die inhaltsbasierte Suche nach Multimedia-Objekten in zentralen Datenbanken bereits seit langem Gegenstand der Forschung ist [14], stellt sie im Bereich P2P-Systeme ein junges Forschungsgebiet dar.

In diesem Bericht vergleichen wir unser auf Zusammenfassungen beruhendes P2P-System mit einem synthetischen, schwer zu erreichenden Benchmark für verteiltes Information Retrieval (IR), 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

[2] z.B. KaZaa, *http://www.kazaa.com*, letzter Abruf: 04.10.2006

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 mit einem auf CANs basierenden Ansatz einer verteilten Indexstruktur [10].

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.

Im folgenden Abschnitt 2 beschreiben wir zunächst Arbeiten, die relevant im Hinblick auf unseren Ansatz sind. In Abschnitt 3 wird das P2P-System skizziert, das Gegenstand unserer Messungen ist, bevor in Abschnitt 4.1 bzw. 4.2 kurz auf Farb- und Textur-Features eingegangen wird, die im Rahmen dieser Arbeit Anwendung finden. Um die verschiedenen Features in Abschnitt 5 miteinander vergleichen zu können, werden Distanzmaße eingesetzt, welche in Abschnitt 4.4 erläutert werden. Abschnitt 5 beschreibt schließlich die Experimente, die von uns durchgeführt wurden, und präsentiert relevante Ergebnisse. In Abschnitt 6 erfolgt eine Zusammenfassung und es wird ein Ausblick auf weitere Arbeiten gegeben.

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 [7, 30]. 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 [14, 9, 39]; 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.

Strukturierte P2P-Systeme, die auf Basis verteilter Hashtabellen (sog. DHTs: *distributed hash tables*) wie etwa Chord [40] oder CAN [33] arbeiten, speichern Indexdaten an einem anderen Ort als die eigentlichen Dokumente.

Inhaltsbasierte Suchdienste wie bspw. Minerva [4, 25] oder PRISM [35] lassen sich bei Chord [40] als zusätzliche Schicht oberhalb der DHT-Schicht implementieren. Minerva liefert für Textdokumente vielversprechende Ergebnisse [27]. Gao et al. [11] implementieren ebenfalls aufbauend auf der DHT-Schicht einen Dienst für das Audio Retrieval.

Verschiedene Erweiterungen von CANs [43, 10] bzw. verteilte Suchbäume [46, 24] erlauben die *k-nearest-neighbor*-Suche (*k*-NN-Suche) nach Bildern, wobei jedoch nur selten Anwendungen im CBIR betrachtet werden.

Als Alternative zu verteilten Indexstrukturen wurden *routing-basierte Ansätze* vorgeschlagen. DISCOVIR [21] unterstützt Ähnlichkeitsanfragen auf Bilddaten in einer Erweiterung eines Gnutella-Netzes[3], indem Anfragen gezielt an Cluster von Peers mit ähnlichen Zusammenfassungen weitergeleitet werden. Weitere Verbesserungen beruhen auf Replikation [36] oder Super-Peer-Architekturen[4] [45, 22, 37].

Verschiedene Methoden zur Ressourcenauswahl bei der Suche in verteilten Datenquellen schlagen Nottelmann und Fuhr vor [28]. Die Parameter des Modells müssen in Abhängigkeit von Daten und Relevanzurteilen gelernt werden. Beim Kontaktieren der Ressourcen steht stets die Minimierung der Gesamtkosten im Vordergrund. Zwei dieser Methoden unterstützen neben der Suche nach Textdokumenten auch die Suche nach Bildern. Hierdurch unterscheiden sich diese Methoden von traditionellen Algorithmen wie CORI [5] oder GlOSS [12].

PlanetP [6] 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. Leider skaliert dieser Ansatz nicht. Rumorama [26] erzeugt eine Hierarchie von derartigen Netzen und erreicht dadurch Skalierbarkeit. Blattnetze verhalten sich wie PlanetP-artige Netze. 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- und Textur-Features sehr breit ist (eine Übersicht geben [14, 9, 39]), fehlen Vergleiche verschiedener Bild-Features im Falle von P2P-Systemen. Deselaers et al. [7] 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. [42, 47, 38]) unentbehrlich um bei Farbfotografien gute Ergebnisse zu erzielen. Da sich unsere Datenkollektion (s. Abschnitt 5.1) 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. [30] vergleichen Distanzmaße für verschiedene Anwendungsszenarien. Einige Maße, die hierbei vielversprechende Resultate erzielen, sind in Abschnitt 4.4 erläutert und werden anschließend in den Experimenten in Abschnitt 5 verwendet. Zusätzlich untersuchen wir weitere Distanzmaße, die sich im Bereich des Bildretrievals als nützlich erwiesen haben [18, 31].

3 Die Peer-to-Peer-Umgebung

Unsere Untersuchungen basieren auf PlanetP [6]. Im Folgenden verwenden wir clusterzentroid-basierte Zusammenfassungen (s. Abschnitt 3.1), die durch einen zufälligen *Rumor Spreading*-Prozess im P2P-Netz verteilt werden. Der zugrunde liegende Mechanismus ist sehr gut in [6, 20, 26] beschrieben, so dass wir uns in

[3] Gnutella Protokollspezifikation v0.4, letzter Abruf: 09.10.2006
http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[4] z.B. Gnutella, *http://www.gnutella.com*, letzter Abruf: 09.10.2006

den folgenden Abschnitten auf eine knappe Darstellung beschränken können. Für unser Verfahren ist es hauptsächlich von Bedeutung, dass sich die Peers in periodischen Zeitabständen gegenseitig kontaktieren, um Zusammenfassungen auszutauschen und so das P2P-Netz aktuell zu halten.

3.1 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 gewissen 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 [8]. 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. Die Größe der Zusammenfassung beträgt $256 \cdot 4$ Byte. Dies ist wesentlich geringer als etwa das Datenvolumen, das für das Übermitteln von Bloom-Filtern bei Textdateien nötig ist [6].

3.2 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 [8], weshalb er im Folgenden Anwendung findet.

StableSortRanker trifft eine Entscheidung auf Grundlage von $L_{clusters}$, einer Liste, die die globalen Cluster-Zentroide, sortiert in aufsteigender Ordnung bezüglich ihrer Distanz zur Anfrage, enthält. Das erste Element dieser Liste entspricht daher immer dem Anfragecluster, d.h. dem Cluster in dem die Anfrage selbst liegt. Peers mit vielen Dokumenten im Anfragecluster werden höher gerankt als Peers mit wenigen Dokumenten im Anfragecluster. Sofern Peer α und Peer β die gleiche Anzahl an Dokumenten im momentan betrachteten Cluster haben, wählt *StableSortRanker* das nächste Element aus $L_{clusters}$ und vergleicht rekursiv Peer α und Peer β bezüglich der Anzahl ihrer Dokumente in diesem Cluster bis entweder eine Entscheidung getroffen werden kann oder das Ende von $L_{clusters}$ erreicht ist.

4 Feature-Extraktion und Ähnlichkeitsberechnung

Bei der inhaltsbasierten Suche werden Medienobjekte in der Regel durch hochdimensionale Feature-Vektoren $\vec{d} = (d_1, ..., d_k)$ repräsentiert, die bspw. den Inhalt eines Bildes beschreiben. 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 relevante 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 $\vec{q} = (q_1, ..., q_k)$ repräsentiert sein.

In Abschnitt 5 werden Messungen basierend auf verschiedenen Feature-Distanz-Kombinationen vorgestellt. Im Folgenden werden zunächst die verwendeten Farb-Features, dann die Textur-Features vorgestellt. Ferner untersuchen wir in unseren Experimenten die Retrieval-Leistung bei Reduzierung der Dimensionalität der Feature-Vektoren, weshalb in Abschnitt 4.3 kurz auf die Hauptkomponentenanalyse eingegangen wird. Abschnitt 4.4 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 [42] verwenden Farbhistogramme um Dokumente in einer Bilddatenbank zu indexieren, wobei die Länge k der Histogramme $\vec{d} = (d_1, ..., d_k)$ 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: Lei et al. [47] 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 [38] 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 (vgl. u. a. [23]) 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 (*color coherence vectors:* CCVs) [29] 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 [41] 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.

4.2 Textur-Features

Neben Farb-Features bilden Textur-Features eine weitere wichtige Feature-Klasse, die im CBIR Verwendung finden. Hierzu werden die Farbbilder in der Regel in Grauwertbilder transformiert. In der Literatur existieren eine Vielzahl vorgeschlagener Textur-Features, eine Übersicht geben [14, 9].

Im Rahmen dieser Arbeit werden drei verschiedene Textur-Features aus [39] verwendet – es sind dies *Autocorrelation*, *Cooccurrence Matrizen* sowie *Edgefrequency*. Sie werden im Folgenden erläutert und bei den Experimenten in Abschnitt 5 verwendet.

AUTOCORR: Die Methode der Autokorrelation [39] misst die Grobkörnigkeit eines Bildes durch Auffinden der räumlichen Beziehungen zwischen einzelnen Textur-Primitiven[5]. Sind diese Flächen groß, deutet dies auf eine grobe Struktur hin (z.B. felsige Oberflächen), sind sie klein, so beschreiben die Primitive eher feine Strukturen (z.B. seidene Stoffe). Im Rahmen dieser Arbeit wird der Autokorrelations-Vektor aus 25 Koeffizienten gebildet, die das Verhältnis zwischen Pixelpaaren (x, y) und $(x + p, y + q)$ abschätzen (p und q reichen jeweils in Zweierschritten von null bis acht).

COOCCUR: Grauwertübergangsmatrizen (Cooccurrence Matrizen) [13] beschreiben die Grauwertübergänge in der Umgebung eines betrachteten Pixels, indem sie die Verbundwahrscheinlichkeiten zweier Pixel x und y erfassen, die in

[5] Grundtexturflächen

einer bestimmten räumlichen Beziehung stehen, welche bzgl. des aktuellen Pixels durch eine gewisse Distanz und einen bestimmten Winkel definiert wird. In der momentanen Implementierung werden Distanzen $dist = \{1, 4, 9, 16\}$ verwendet, woraus sich vier Grauwertübergangsmatrizen ergeben. Da mittels dieser jeweils fünf statistische Kennzahlen (maximale Wahrscheinlichkeit, Differenzmoment, inverses Differenzmoment, Entropie und Homogenität) berechnet werden (vgl. [39]), resultiert ein 20-dimensionaler Feature-Vektor.

EDGEFREQ: Das Textur-Feature der Edgefrequency [39] berechnet Differenzen von Gradienten zwischen einem Pixel (x, y) und benachbarten Pixeln in einer Entfernung d. Für einen bestimmten Distanzwert d werden die Differenzen der Gradienten über das ganze Bild aufsummiert. Bei der Wahl von $d = \{0, 2, ..., 48, 50\}$ ergibt sich hieraus für jedes Bild ein 25-dimensionaler Feature-Vektor.

4.3 Hauptkomponentenanalyse

Hochdimensionale Feature-Vektoren stellen eine besondere Herausforderung für Clustering-Algorithmen [3] und Distanzmaße [1] dar. Eine Möglichkeit die Dimensionalität der Feature-Vektoren zu reduzieren bietet die Hauptkomponentenanalyse (auch PCA: *Principal Component Analysis*). 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 Peer-to-Peer-Systemen ist eine verteilte Variante der PCA anzuwenden, wie sie etwa in [2, 32] vorgestellt wird. Die Auswirkungen der Anwendung der PCA auf die Retrieval-Leistung wird in Abschnitt 5 betrachtet.

4.4 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. Sie liefern für jedes Dokument-Anfrage-Paar einen Wert aus $\mathbb{R}_0^+$. Je geringer die Distanz zweier Feature-Vektoren, desto höher ist ihre Ähnlichkeit. Daher können die Distanzwerte durch Verwendung einer Umkehrfunktion, z.B. $2^{-dist(\vec{q},\vec{d})}$, in Ähnlichkeitswerte aus dem Intervall $[0, 1]$ überführt werden. 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. [34]. Die Autoren unterscheiden Bin-By-Bin-Distanzmaße, die bei der Distanzberechnung nur Vektorkomponenten der beiden Vektoren mit dem gleichen Index miteinander in Beziehung setzen, von sog. Cross-Bin-Distanzmaßen. Letztere berücksichtigen Informationen, die über verschiedene Indizes hinweg berechnet werden.

Bin-By-Bin-Distanzmaße

- *Minkowski-Distanz:*

$$dist_{L_m}(\vec{q},\vec{d}) = \left(\sum_i |q_i - d_i|^m\right)^{1/m} \tag{1}$$

Im Bereich CBIR häufig verwendete Distanzmaße sind drei Ausprägungen der Minkowski-Distanz [42, 41, 34]; es sind dies die Manhattan-Distanz $dist_{L_1}$, die Euklidische Distanz $dist_{L_2}$, sowie die L_{max}-Distanz $dist_{L_{max}}$. Letztere resultiert aus obiger Formel für $\lim_{m\to\infty} dist_{L_m}(\vec{q},\vec{d})$ und entspricht dem Betrag der maximalen Differenz zwischen zwei Vektorkomponenten, die den gleichen Index besitzen.

- *Fraktionale Distanz:*

Während die klassischen Minkowski-Distanzmaße nur für $m \geq 1$ definiert sind, erweitern [1] 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 [18] bestätigen in ihren Untersuchungen, dass ein Wert von $m = \frac{1}{2}$ meist bessere Retrieval-Ergebnisse als etwa $dist_{L_1}$ oder $dist_{L_2}$ liefert. Daher verwenden wir $m = \frac{1}{2}$ in unseren Experimenten.

- *Kullback-Leibler Divergenz:*

$$dist_{KL}(\vec{q},\vec{d}) = \sum_i q_i \cdot \log\frac{q_i}{d_i} \tag{2}$$

Bei der Kullback-Leibler Divergenz (auch relative Entropie) 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 $dist(\vec{d'},\vec{d''}) = dist(\vec{d''},\vec{d'})$, wird im Folgenden eine symmetrische Variante der Kullback-Leibler Distanz verwendet [44]:

$$dist_{SKL}(\vec{q},\vec{d}) = \frac{1}{2}\sum_i (q_i - d_i)\log\frac{q_i}{d_i} \tag{3}$$

- *Kosinusmaß:*

$$dist_{cos}(\vec{q},\vec{d}) = \frac{\sum_i q_i \cdot d_i}{\sqrt{\sum_i q_i^2}\cdot\sqrt{\sum_i d_i^2}} \tag{4}$$

Vielfach wird im Bereich des Information Retrievals bei k-NN-Anfragen, speziell auch im CBIR [31], der Kosinus des Winkels zweier Vektoren als Maß für die Unähnlichkeit zweier Dokumente eingesetzt. Je geriner dieser Winkel desto größer ist die Ähnlichkeit der durch die Vektoren repräsentierten Dokumente.

Cross-Bin-Distanzmaße

Bin-By-Bin-Distanzmaße 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 jedoch Zusammenhänge zwischen den Bins erfasst werden.

- *Match-Distanz:*

$$dist_{match}(\vec{q}, \vec{d}) = \sum_i |Q_i - D_i| \tag{5}$$

  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 $(D_1, ..., D_k)$, 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:*

$$dist_{KS}(\vec{q}, \vec{d}) = \max_i |Q_i - D_i| \tag{6}$$

  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 5.2 basieren auf 50 Simulationsläufen mit jeweils 100 Anfragen. Insgesamt fließen demnach pro Feature-Distanz-Kombination 5000 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). Diese werden auf Basis der globalen Dokumentenkollektion a priori berechnet. Sie können als Retrieval-Ergebnis des zentralen Falles betrachtet werden, gegen das wir unser verteiltes P2P-Retrieval-System evaluieren.

5.1 Datenselektion

Unsere Experimente basieren auf Bildern, die aus einem Crawl von Flickr[6] 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 50.000 Bilder von 2.623 zufällig ausgewählten Flickr-Nutzern. Von jedem dieser Bilder werden die in Abschnitt 4 beschriebenen Farb- und Textur-Features extrahiert. Die Bilder werden so auf die Peers verteilt, dass jeder Peer einen Flickr-Nutzer repräsentiert.

[6] Flickr.com, *http://www.flickr.com*, letzter Abruf: 04.10.2006

5.2 Vergleich des verteilten mit dem zentralisierten Retrieval-Ergebnis

Im Folgenden werden zunächst die in Abschnitt 4 vorgestellten Farb- und Textur-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 1 zeigt 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. Tendenziell scheinen sich niedrigdimensionale Feature-Vektoren besser für eine P2P-Umgebung zu eignen, gerade auch vor dem Hintergrund, dass lokale Farbhistogramme mit 576 Dimensionen die schlechteste Performance aller betrachteten Farb-Features liefern.

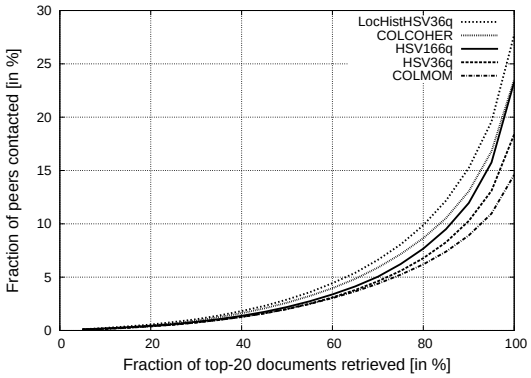

Abbildung 1: Einfluss verschiedener Farb-Features auf die Anzahl der zu kontaktierenden Peers

Abbildung 2 stellt im Gegensatz zu Abbildung 1, in der die Anzahl der kontaktierten Peers betrachtet wurde, 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 50.000 Dokumenten auf 2.623 Peers nicht den Transfer von über 22.500 Feature-Vektoren im P2P-Netz, sondern lediglich die Übertragung von

maximal $15\% \cdot 2.623 \cdot 20 = 7.869$ Feature-Vektoren.

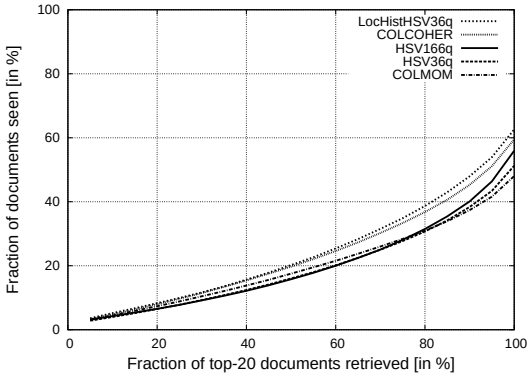

Abbildung 2: Einfluss verschiedener Farb-Features auf die Anzahl der betrachteten Dokumente

Abbildung 3 visualisiert die Anzahl der kontaktierten Peers bei der Suche nach den 20 ähnlichsten Dokumenten für die verschiedenen Textur-Features aus Abschnitt 4.2. Die beiden Textur-Features *Autocorrelation* und *Edgefrequency* verhalten sich hinsichtlich der Anzahl zu kontaktierender Peers trotz gleicher Dimensionalität der Feature-Vektoren unterschiedlich. Neben der Dimensionalität spielen demnach weitere Faktoren eine Rolle. Die Ausprägung der Feature-Werte determiniert etwa zusammen mit dem gewählten Distanzmaß die Qualität der Summaries und damit des gesamten Ranking-Prozesses.

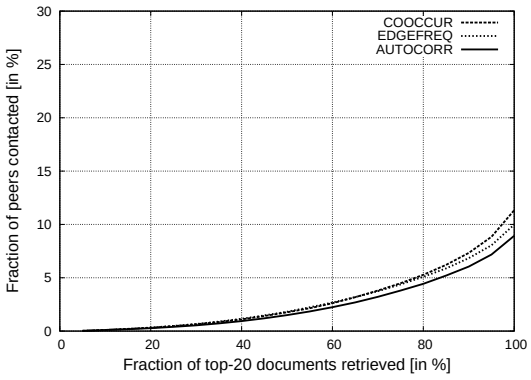

Abbildung 3: Einfluss verschiedener Textur-Features auf die Anzahl der zu kontaktierenden Peers

In Abschnitt 4.4 wurden verschiedene Distanzmaße betrachtet, auf deren Basis die Ähnlichkeit zwischen Dokumenten berechnet werden kann. Abbildung 4 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. Abbildung 1), nicht alle Distanz-Kombinationen sinnvoll sind. Abbildung 4 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.

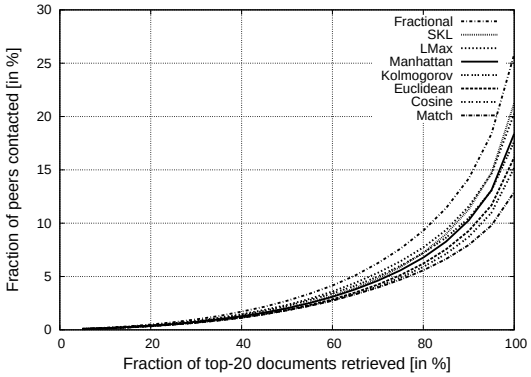

Abbildung 4: Einfluss der Distanzmaße auf die Anzahl der kontaktierten Peers (Feature: HSV36q)

Der Einfluss der Hauptkomponentenanalyse (Feature: HSV36q, Distanz: Manhattan) auf das Retrieval ist in Abbildung 5 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 im *Curse of Dimensionality*, der eine Ähnlichkeitssuche im hochdimensionalen Raum aufwändig macht [16]. Gibt der Nutzer sich mit 60% der Top-20 zufrieden, so müssen jedoch bei allen Varianten der PCA gleich viele Peers (etwa 3%) kontaktiert werden.

Abbildung 6 fasst die von uns durchgeführten Messungen zusammen. Wir stellen jeweils dar, wieviele Peers bei der Suche nach den jeweiligen Top-20-Bildern kontaktiert werden müssen. Die Match-Distanz zeigt hierbei bei den meisten Features – mit einer Ausnahme – ein günstigeres Verhalten als die anderen Distanzmaße.

5.3 Retrieval-Experimente mit Content-Addressable Networks

Content-Addressable Networks [33] waren die ersten mehrdimensionalen Indexstrukturen für P2P-Netze. Sie erlauben es Schlüssel/Wert-Paare abzulegen. Jeder Schlüssel ist hierbei ein n-dimensionaler Vektor $\vec{v}$ aus dem n-dimensionalen Einheitshyperwürfel $\vec{v} \in [0;1]^n$.

In einem CAN ist jeder Peer für eine achsenparallele, quaderförmige Region des Einheitswürfels 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 d-dimensionalen CAN mit N Knoten müssen $\mathcal{O}(\sqrt[d]{N})$

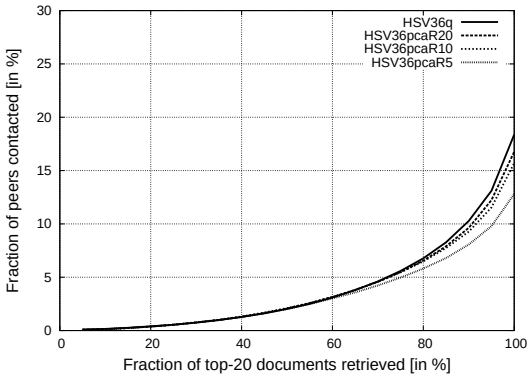

Abbildung 5: Einfluss der PCA auf die Anzahl der kontaktierten Peers (Feature: HSV36q)

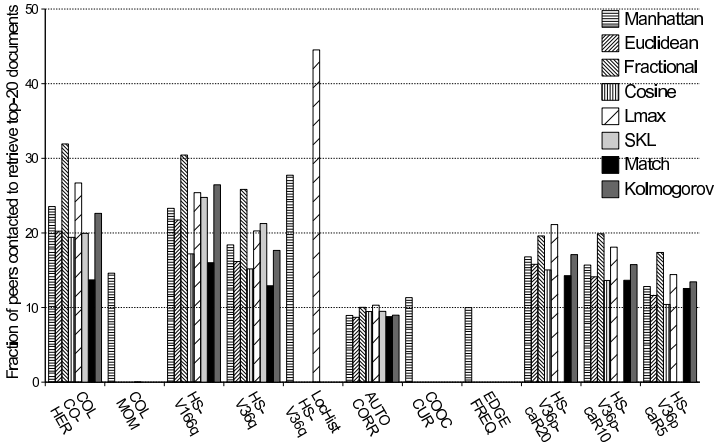

Abbildung 6: Anteil zu kontaktierender Peers, um alle Top-20-Bilder zu finden.

Routingschritte durchgeführt werden, um eine Membership-Anfrage zu beantworten[7].

Ähnlich wie [43, 10] 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

[7]Small-world-CAN-Varianten, die logarithmische Komplexität bieten, existieren, wie z.B. [10]. Für die von uns betrachteten hochdimensionalen Räume ist die hier zu erwartende Ersparnis allerdings nicht relevant.

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 [10] 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 [15]. Als Splitdimension d_s wird die Dimension mit der höchsten Varianz gewählt und die Kollektion entlang dieser Dimension so aufgeteilt, dass diejenigen Punkte $\vec{v}$ deren d_s-te Komponente v_{d_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 [17]. 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.

Vergleich zwischen einer CAN-Implementierung und *StableSortRanker*

Abbildung 7 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 7, rechte Seite). *StableSortRanker* kontaktiert dabei weniger Peers, um die gleiche Anzahl an Top-20-Dokumenten aufzufinden. In dieser Abbildung ist zusätzlich die in [8] 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

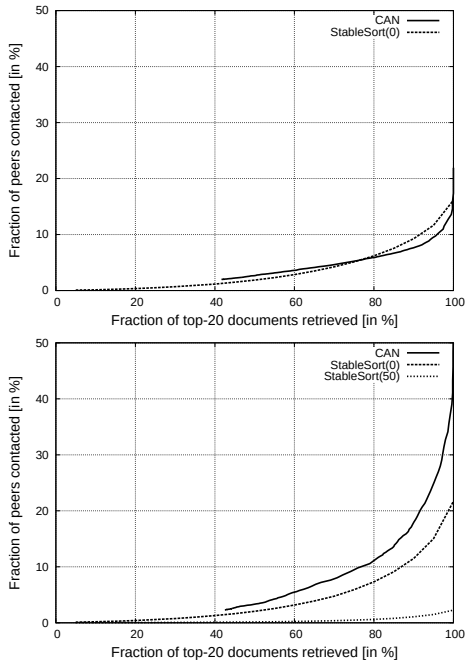

Abbildung 7: CAN-Implementierung vs. *StableSortRanker* (links: HSV36q rechts: HSV166q)

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.

5.4 Evaluierung der Retrieval-Qualität von *StableSortRanker*

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. [19]) 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 verwenden 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 theore-

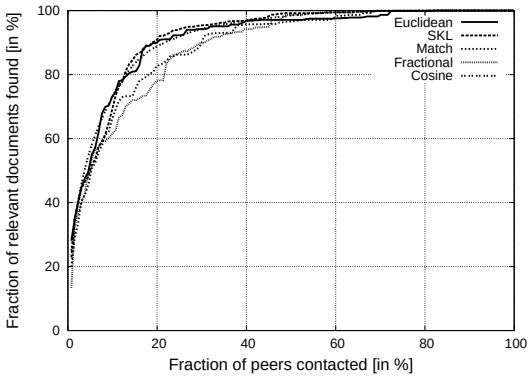

Abbildung 8: Kontaktierte Peers vs. relevante Bilder gefunden (Feature: HSV36q)

tische Poolgröße von 1.350 Bildern. Aus Gründen der Überlappung reduziert sich die Poolgröße im Durchschnitt auf 619 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 50-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 im Folgenden die signifikantesten Ergebnisse. Abbildung 8 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 1 besser als etwa HSV166q. Es weist darüber hinaus in Abbildung 4 in Kombination mit $dist_{cos}$ und $dist_{match}$ ein ähnlich gutes bzw. besseres Verhalten als die Kombination von Farbmomenten und Manhattan-Distanz in Abbildung 1 auf. Diese Kombination aus $dist_{L_1}$ und Farbmomenten, die bei der Messung in Abbildung 1 sehr gut abschneidet, fällt jedoch bei der Betrachtung des relativen Recalls klar hinter die besten Ergebnisse, wie sie in Abbildung 8 dargestellt sind, zurück.

Insbesondere das Kosinusmaß sowie die symmetrische Variante der Kullback-Leibler Divergenz zeigen in Abbildung 8 ein besseres Verhalten als etwa die Match-Distanz oder die fraktionale Distanz, die sich am schlechtesten verhält und wohl schon aufgrund ihres schlechten Verhaltens in Abbildung 4 ausscheidet. Um etwa 80% aller relevanten Dokumente zu finden, werden bei Verwendung von $dist_{cos}$, $dist_{L_2}$ bzw. $dist_{SKL}$ lediglich etwa 13% der Peers kontaktiert. Demgegenüber werden bei Verwendung von $dist_{match}$ etwa 18% der Peers besucht.

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 symmetrische Variante der Kullback-Leibler Divergenz findet ebenso nach einer geringen Anzahl kontaktierter Peers einen Großteil der relevanten Dokumente, wenngleich sie bei

den Top-20-Experimenten in Abschnitt 5.2 $dist_{cos}$ deutlich unterlegen ist. Die Verwendung der Euklidischen Distanz scheint bei Betrachtung von Abbildung 4 und Abbildung 8 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

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. *LNCS*, 1973:420–434, 2001.

[2] J.Z. Bai, R. Chan, and F. Luk. Principal component analysis for distributed data sets with updating. In *6th International Workshop on Advanced Parallel Processing Technologies, LNCS, Vol. 3756, Hong Kong, China*, pages 471–483, 2005.

[3] Christian Baumgartner et al. Subspace selection for clustering high-dimensional data. *Fourth IEEE International Conference on Data Mining*, pages 11–18, 2004.

[4] Matthias Bender et al. The minerva project: Database selection in the context of p2p search. In *11. GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW'05)*, 2005.

[5] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. *18th Int. ACM SIGIR Conference on Research and Development in Information Retrieval, New York*, 1995.

[6] F. M. Cuenca-Acuna and T.D. Nguyen. Text-based content search and retrieval in ad hoc p2p communities. Technical Report DCS-TR-483, Department for Computer Science, Rutgers University, 2002.

[7] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval – a quantitative comparison. In *Pattern Recognition, 26th DAGM Symposium*, number 3175 in LNCS, pages 228–236, Tübingen, Germany, 2004.

[8] Martin Eisenhardt et al. Clustering-based source selection for efficient multimedia retrieval in peer-to-peer networks. *wird erscheinen in: Proc. of the 2nd IEEE Int. Workshop on Multimedia Information Processing and Retrieval, San Diego, California*, 2006.

[9] David Feng, W C Siu, and Hong Jiang Zhang. *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*, chapter Fundamentals of content-based image retrieval, pages 1–26. Springer, 2003.

[10] Prasanna Ganesan, Beverly Yang, and Hector Garcia-Molina. One torus to rule them all: multi-dimensional queries in p2p systems. In *Proc. of the 7th Int. Workshop on the Web and Databases*, pages 19–24, New York, NY, USA, 2004.

[11] Jun Gao, George Tzanetakis, and Peter Steenkiste. Content-based retrieval of music in scalable peer-to-peer networks. *IEEE Int. Conf. on Multimedia & Expo, ICME'03, Baltimore, MD*, I:309–312, 2003.

[12] L. Gravano and H. Garcia-Molina. Generalizing gioss to vector-space databases and broker hierarchies. *VLDB'95, Los Altos, California,*, 1995.

[13] J. F. Haddon and J. F. Boyce. Co-occurence matrices for image analysis. In *IEEE Electronics and Communications Engineering Journal 5(2)*, pages 71–83, 1993.

[14] Alaa Halawani et al. Fundamentals and applications of image retrieval: An overview. *Datenbank Spektrum*, 18:14–23, August 2006.

[15] Andreas Henrich, Hans-Werner Six, and Peter Widmayer. The LSD tree: Spatial Access to Multidimensional Point and Nonpoint Objects. In Peter M. G. Apers and Gio Wiederhold, editors, *VLDB*, pages 45–53. Morgan Kaufmann, 1989.

[16] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *The VLDB Journal*, pages 506–515, 2000.

[17] Gísli R. Hjaltason and Hanan Samet. Distance browsing in spatial databases. *ACM Trans. Database Syst.*, 24(2):265–318, 1999.

[18] Peter Howarth and Stefan M. Rüger. Fractional distance measures for content-based image retrieval. In David E. Losada and Juan M. Fernández-Luna, editors, *27th European Conf. on IR Research*, volume 3408 of *Lecture Notes in Computer Science*, pages 447–456. Springer, 2005.

[19] K. Sparck Jones and C. J. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[20] R. Karp et al. Randomized rumor spreading. In *FOCS '00: Proc. of the 41st Annual Symposium on Foundations of Computer Science*, page 565, Washington, DC, USA, 2000. IEEE Computer Society.

[21] Irwin King, Cheuk Hang Ng, and Ka Cheung Sia. Distributed content-based visual information retrieval system on peer-to-peer networks. volume 22, pages 477–501, New York, NY, USA, 2004. ACM Press.

[22] Jie Lu and Jamie Callan. Content-based retrieval in hybrid peer-to-peer networks. In *12th Int. Conf. on Information and Knowledge Management*, New York, NY, 2003.

[23] Jamal Malki et al. Region queries without segmentation for image retrieval by content. In *Visual Information and Information Systems*, pages 115–122, 1999.

[24] Batko Michal, Gennaro Claudio, and Zezula Pavel. A scalable nearest neighbor search in p2p systems. *2nd Int. VLDB Workshop on Databases, Information Systems and Peer-to-Peer Computing, Toronto.*, 2005.

[25] Sebastian Michel et al. IQN Routing: Integrating quality and novelty for web search. In *EDBT*, 2006.

[26] Wolfgang Müller, Martin Eisenhardt, and Andreas Henrich. Scalable summary based retrieval in p2p networks. In *CIKM '05: Proc. of the 14th ACM Intl. Conf. on Information and knowledge management*, pages 586–593, New York, NY, 2005.

[27] Thomas Neumann et al. A reproducible benchmark for p2p retrieval. In *Proc. of the 1st Int. Workshop on Performance and Evaluation of Data Management Systems*, Chicago, Illinois, USA, 2006.

[28] H. Nottelmann and N. Fuhr. Decision-theoretic resource selection for different data types in mind. *ACM SIGIR Workshop on Distributed Information Retrieval*, 2003.

[29] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, New York, NY, USA, 1996. ACM Press.

[30] Jan Puzicha et al. Empirical evaluation of dissimilarity measures for color and texture. In *Int. Conf. on Computer Vision, Kerkyra, Griechenland*, 1999.

[31] Gang Qian et al. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1232–1237, New York, NY, USA, 2004. ACM Press.

[32] Y. Qu et al. Principal component analysis for dimension reduction in massive distributed data sets. In *2nd SIAM International Conference on Data Mining: Workshop on High Performance Data Mining*, pages 4–9, 2002.

[33] Sylvia Ratnasamy et al. A scalable content-addressable network. In *Proc. 2001 Conf. on applications, technologies, architectures, and protocols for computer communications*, San Diego, CA, United States, 2001.

[34] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. volume 40, pages 99–121, Hingham, MA, USA, 2000.

[35] O. D. Sahin et al. PRISM: indexing multi-dimensional data in P2P networks using reference vectors. In *Proc. of the 13th annual ACM Intl. Conf. on Multimedia*, pages 946–955, New York, NY, USA, 2005.

[36] Nima Sarshar, P. Oscar Boykin, and Vwani P. Roychowdhury. Percolation search in power law networks: making unstructured peer-to-peer networks scalable. In *Proc. of 4th Int. Conf. on Peer-to-Peer Computing*, pages 2–9. IEEE, August 2004.

[37] H. Shen, Y. Shu, and B. Yu. Efficient semantic-based content search in p2p network. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):813–826, 2004.

[38] J. Smith and S. Chang. Single color extraction and image query. *Proc. IEEE Int. Conf. on Image Proc.*, pages 528–531, 1995.

[39] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson Learning Vocational, 2 edition, 1998.

[40] Ion Stoica et al. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proc. ACM SIGCOMM Conf.*, San Diego, CA, USA, 2001.

[41] Markus A. Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[42] M. J. Swain and D. H. Ballard. Color indexing. In *International Journal of Computer Vision, vol. 7, no. 1*, pages 11–32, 1991.

[43] C. Tang, Z. Xu, and M. Mahalingam. pSearch: Information retrieval in structured overlays. In *1st Workshop on Hot Topics in Networks*, Princeton, NJ, 2002.

[44] Raymond Veldhuis. The centroid of the symmetrical kullback-leibler distance. *IEEE Signal Processing Letters*, Vol. 9 No. 3:96–99, 2002.

[45] Beverly Yang and Hector Garcia-Molina. Designing a super-peer network. In *IEEE Intl. Conf. on Data Engineering*, 2003.

[46] Zhang, Arvind, and Chi. Brushwood: Distributed trees in peer-to-peer systems. *4th Int. Workshop, IPTPS 2005, Ithaca NY, USA*, 2005.

[47] L. Zhang, F. Lin, and B. Zhang. A cbir method based on color-spatial feature. *IEEE Region 10 Annual International Conference 1999*, pages 166–169, 1999.