

Grafische Bildsuche mit PictureFinder

Graphical Image Retrieval with PictureFinder

Thorsten Hermes, Andrea Miene und Andree Lüdtko
Digitale Medien – Bildverarbeitung, Technologie-Zentrum Informatik, Universität Bremen
Am Fallturm 1, 28359 Bremen
Tel.: +494212183316, Fax: +494212187196
E-mail: {hermes,miene,aluedtke,herzog}@tzi.de, Internet: <http://www.tzi.de>

Zusammenfassung:

Sowohl bei der Bildsuche im Internet (z.B. mit Yahoo) als auch in Datenbanken von Bildagenturen ist die textbasierte Suche nach bildlichen Inhalten immer noch gängige Praxis. Es wird nach Annotationen von Bildern gesucht, die in der Regel manuell erstellt wurden. Insbesondere im Internet fehlen diese meist. In Ermangelung einer Inhaltsbeschreibung wird oft lediglich der Dateiname herangezogen. Dieser Ansatz einer Volltextsuche führt häufig zu Fehlern.

In Analogie wird der Ansatz der „Vollbildsuche“ – also einer grafischen Suche (per Skizze oder Beispielbild) – aufgrund der Komplexität der Anfrageformulierung bisher selten bzw. gar nicht verwendet. In dieser Arbeit stellen wir unseren Ansatz zur grafischen Bildsuche und zur Bildähnlichkeitssuche vor. Durch Skizzen können Bilder anhand visueller Charakteristika gesucht werden. Die Kriterien sind dabei an den Ähnlichkeitsbegriff des Benutzers adaptierbar. Darüber hinaus besteht die Möglichkeit, nach ähnlichen Bildern zu suchen. Auch solche Anfragen werden dabei zunächst auf eine (generierte) grafische Anfrage zurückgeführt, die wiederum durch den Benutzer adaptiert werden kann.

Abstract:

On the Internet (e.g. Yahoo) as well as in large image databases the text search paradigm predominates when users are searching for image data. The usual way to search for images is to use manually formulated annotations. But especially on the internet such annotations are mostly unavailable. Typically a search based on filenames is performed instead which often leads to undesired or surprising results.

Searching image data by querying graphically, either with an image or sketch, currently seems not to be the preferred method due to the complexity in designing the query. In this paper we present our approach for graphically querying visual content and for image similarity search. Image repositories can be queried by sketches. Search criteria are flexibly adaptable to the user's understanding of similarity. Furthermore, the user can search for images similar to a given example image and these queries are also mapped to (generated) graphical queries which then can be refined in the same manner.

1. Einleitung

Betrachtet man die Tatsache, dass Bilder immer noch überwiegend mittels Text gesucht werden (insbesondere im Internet), so können die Suchresultate mitunter sehr überraschend sein, da die Annotationen der Bilder oft nur die jeweiligen Dateinamen oder Pfade sind. Die Gründe liegen auf der Hand: Es ist kaum möglich, alle Bilder im WWW per Hand zu annotieren.

Diese Beobachtung liefert entscheidende Argumente dafür, dass eine Erweiterung der herkömmlichen textuellen Bildsuche um grafische Suchkriterien wie Farbe, Textur und/oder Kontur wesentlich zur Verbesserung der Suchergebnisse beitragen wird. Es sollte nicht nur eine textbasierte, sondern auch eine grafische Suche nach Bildern möglich sein, bzw. eine Kombination aus beiden Anfragemethoden. PictureFinder ermöglicht genau diese Kombination von textbasierter und grafischer Suche. Dabei wird der für die grafische Suche benötigte Index vollautomatisch erzeugt, so dass kein zusätzlicher manueller Aufwand entsteht.

1.1. Stand der Forschung

Dieser Abschnitt gibt einen kurzen Überblick bezüglich der Entwicklungen auf dem Gebiet der Bildanalyse, die zur Unterstützung des Retrievals multimedialer Daten dienen.

Die von der Bedienung her einfachste Form der grafischen Bildsuche ist die Suche nach ähnlichen Bildern, indem der Benutzer ein Beispielbild vorgibt (query by example). Hierzu existieren bereits verschiedene Ansätze zur Ähnlichkeitssuche ([2], [3], [4], [7], [12], [15],[16], [17]).

Neben der Suche nach ähnlichen Bildern gibt es auch Ansätze zur grafischen Bildsuche, die spezielle Benutzungsschnittstellen zur Formulierung von einfachen grafischen Anfragen bieten. Als eine frühe Entwicklung auf diesem Gebiet ist das QBIC System von IBM zu nennen [9]. Trotzdem ist immer noch zu beobachten, dass auch eine wohl formulierte Anfrage nicht unbedingt direkt zum gewünschten Ergebnis führt. Zum einen ist dies in der Tatsache begründet, dass die Schnittstellen nicht entsprechend ausgereift sind, um Anfragen hinreichend präzise zu formulieren. Zum anderen kann es auch daran liegen, dass die relevanten Attribute der multimedialen Objekte schwer oder auch gar nicht extrahiert werden können. Ebenso mag es für den Anwender nicht offensichtlich sein, welche Attribute vom System unterstützt werden, bzw. kann ein Anwender andere Attribute als wichtig erachten als die vom System bereitgestellten. Ein ausführlicher Überblick über CBIR-Systeme wird in [19] gegeben.

Aus diesen Gründen ist es wichtig, Methoden zu entwickeln, die den Nutzer beim Suchprozess sinnvoll unterstützen. Insbesondere sollten Anfragen an die Nutzerinteressen adaptierbar sein.

2. Überblick über das PictureFinder System

PictureFinder ist ein System zur automatischen Annotation von großen Bilddatenbanken. Gegenwärtig besteht PictureFinder aus einem Analysemodul sowie einem Suchmodul.

Die PictureFinder Suche erlaubt es dem Nutzer, seine Anfrage grafisch zu formulieren. Das bedeutet, er kann nach bestimmten Farb- und/oder Textureigenschaften hinsichtlich einer oder mehrerer Regionen suchen. Unter der URL <http://www-agki.tzi.de/bv/pfdemo> ist eine Onlinedemo des PictureFinder Systems verfügbar.

Die Farbanalyse ist in zwei Teilschritte unterteilt. Zunächst wird das jeweilige Bild durch ein zentroides Bereichswachstumsverfahren [18] farbsegmentiert. Anschließend werden die resultierenden Regionen indiziert. Die Farbe einer Region wird durch den mittleren Farbwert im HLS Raum [6] beschrieben. Dieser Farbwert, in Verbindung mit der Position im Bild, bildet den Farbindex. Zur Positionsbestimmung wird ein Raster verwendet. Wenn eine Rasterzelle zu mindestens 5% von einer Farbregion überlagert wird, werden die Farbe und die prozentuale Füllung der Rasterzelle im Index abgelegt. Ein Beispiel für eine Farbsegmentierung ist in Abbildung 1 (Mitte) dargestellt.

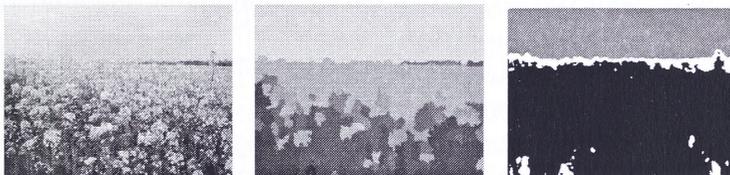


Abbildung 1: Original (links), Ergebnis der Farb- (Mitte) bzw. Textursegmentierung (rechts)

In der Texturanalyse wird das Bild in einem ersten Schritt in Regionen mit einheitlicher Textur segmentiert. Die Segmentierung verwendet eine Bank von Gaborfiltern mit n Frequenzen und m Orientierungen. Das Eingabebild wird in $n \times m$ Bänder unterteilt, wobei jedes Band für eine lokale Intensität einer korrespondierenden Frequenz steht [13]. In einem nachfolgenden Schritt werden punktwise durch eine nichtlineare Transformation alle Werte positiv und der Kontrast verstärkt. Die Mittelwerte der resultierenden „texture-blob“-Bilder werden bestimmt und über ein Clusteringverfahren (Mean Shift) [5] zusammengefasst. Hierbei werden als zusätzliche Merkmale die 2D Koordinaten eines jeden Punktes genutzt. Dadurch wird die räumliche Kohärenz der resultierenden Texturregionen verbessert [11]. Ein Beispiel ist in Abbildung 1 (rechts) zu sehen. Um die Texturen zu indexieren, wird auch hier ein Raster über das Bild gelegt. Dabei werden nur diejenigen Rasterzellen berücksichtigt, die vollständig innerhalb genau einer Texturregion liegen. Für diese Rasterzellen werden statistische Merkmale berechnet, aufgrund derer der Ausprägungsgrad von fünf visuellen Eigenschaften von Texturen (*homogen, gesprenkelt, mehrflächig, gestrichelt* und *wellig*) bestimmt wird. Mit Hilfe von Schwellwerten kann bestimmt werden, ob eine bestimmte visuelle

Eigenschaft vorliegt oder nicht. Im Texturindex wird für die relevanten Rasterzellen gespeichert, welche der fünf visuellen Eigenschaften vorliegen.

3. Formulierung einer Anfrage

Wir unterscheiden zwei Kategorien grafischer Anfragen: Erstens eine durch den Benutzer skizzierte Anfrage (query by sketch) und zweitens eine Anfrage, die durch ein Beispielbild (query by example) formuliert wird. Im ersten Fall kann der Nutzer mit verschiedenen grafischen Malwerkzeugen Farb- und/oder Texturregionen auf einer Zeichenfläche festlegen. Die einzelnen Flächen können unterschiedlich gewichtet werden. Außerdem ist es möglich, das Gewicht einzelner Attribute – wie Farbwert, Sättigung, Helligkeit, Textur – für jede Fläche festzulegen. Ebenso lässt sich über einen Parameter „Dichte“ einstellen, in welchem Maße die skizzierte Region von passenden Regionen im gesuchten Bild überdeckt werden muss. Eine Beispielanfrage ist in der Abbildung 2 zu sehen.

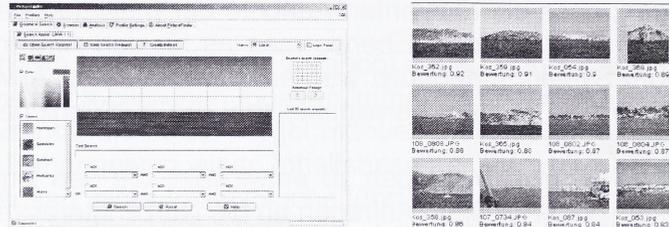


Abbildung 2: Beispiel für eine grafische Anfrage und Ergebnisse

Eine Ähnlichkeitssuchanfrage wird ebenfalls auf das Prinzip der regionenbasierten grafischen Suche zurückgeführt. Hierzu wird automatisch eine Anfrage generiert, die aus Farbregionen besteht, die den Farbflächen in dem Beispielbild entsprechen. Dadurch ist es auch hier möglich, die Anfrage zu später modifizieren, um sie an spezifische Nutzerintentionen zu anzupassen.

4. Ähnlichkeitsmaß

Die Abarbeitung einer Suchanfrage unterteilt sich in zwei Schritte. Zunächst werden alle Bilder, die innerhalb des geforderten Toleranzbereichs liegen, ermittelt. Anschließend werden die Bilder gemäß einer Rangfolge sortiert. Der Prozess des Vergleichens findet auf drei Ebenen statt:

1. Bestimmung der Ähnlichkeit für eine einzelne Region, die aus einer relevanten Menge von Rasterzellen $\{c_1, \dots, c_m\}$, mit $m \geq 1$ besteht. Die Gesamtähnlichkeit ergibt sich aus der Ähnlichkeit aller relevanten Rasterzellen.
2. Vergleich einer einzelnen Rasterzelle hinsichtlich der gesuchten Farbe und/oder Textur.
3. Aufgrund der Ergebnisse von (1.) erfolgt die Evaluation der gesamten grafischen Anfrage, die aus n Anfrageregionen r_1, \dots, r_n , mit $n \geq 1$ besteht.

Diese einzelnen Schritte werden in den folgenden Abschnitten genauer diskutiert.

4.1. Ähnlichkeit einzelner Regionen

Um die Ähnlichkeit hinsichtlich einer einzelnen Anfrageregion zu bestimmen, wird zunächst jede Rasterzelle bestimmt, die ganz oder teilweise von der Region überdeckt wird (vgl. Abb. 3). Diese Rasterzellen bilden die Menge der relevanten Zellen $\{c_1, \dots, c_n\}$.

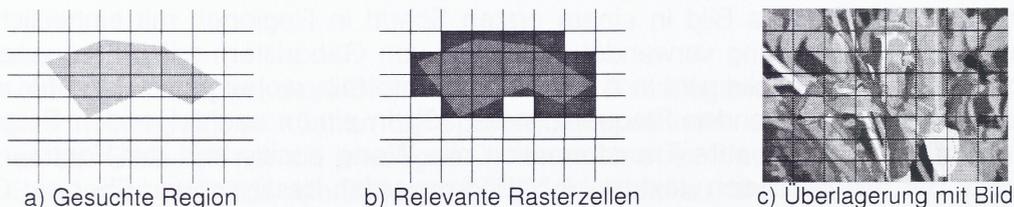


Abbildung 3: Bestimmung relevanter Rastererelemente

Ein Bild wird mit einer Fläche verglichen, indem jede relevante Rasterzelle mit dem Index der korrespondierenden Zelle verglichen wird (vgl. Abschnitt 4.2). Um festzustellen, ob das Bild für die geprüfte Region innerhalb der Toleranz liegt und damit zur Ergebnismenge gehört, wird geprüft, ob das Kriterium der minimalen Abdeckung (Dichte) erfüllt ist. Dieses Kriterium kann für jede einzelne Fläche durch den Benutzer eingestellt werden. Der Defaultwert ist $minCov_r = 0,65$, was einer Mindestabdeckung von 65 % entspricht. Um das Abdeckungskriterium zu bewerten, muss die Größe

der gesuchten Fläche ermittelt werden. Dafür wird für jede relevante Rasterzelle c der Füllgrad $fill(c, col_r)$ bestimmt (col_r bezeichnet die Farbe der Region r). Der Füllgrad einer Rasterzelle ist ein Wert zwischen 0 und 1. Die Größe einer Region ist gegeben durch:

$$sizeRegion(r) = \sum_{k=1}^n fill(c_k, col_r).$$

Dann werden die Ähnlichkeitswerte für die einzelnen Rasterzellen summiert und mit der geforderten minimalen Abdeckung verglichen. Das Abdeckungskriterium ist erfüllt, wenn:

$$\sum_{k=1}^n cellSimilarity(c_k, i) > sizeRegion(r) \cdot minCov_r$$

c_k steht für die relevanten Rasterzellen einer Fläche und i für das Bild, mit dem verglichen wird. Die Funktion $cellSimilarity$ liefert einen Wert zwischen 0 und 1, der ein Maß für die Ähnlichkeit einer Rasterzelle c_k mit der korrespondierenden Rasterzelle des jeweiligen Bildes i ist (vgl. Abschnitt 4.2). Ist das Abdeckungskriterium erfüllt, wird das Bild i der Ergebnismenge der Region r zugeordnet. Die Ähnlichkeit der Fläche wird wie folgt berechnet:

$$regionSimilarity(r, i) = \frac{\sum_{k=1}^n cellSimilarity(c_k, i)}{n}$$

So wird für jede Fläche eine Ergebnismenge aus ähnlichen Bildern ermittelt. Wird die Option *irgendwo* bei der Suche aktiviert, dann wird die jeweilige Fläche überall im Bild gesucht. Das bedeutet, dass alle Rasterzellen betrachtet werden müssen.

4.2. Ähnlichkeit einzelner Rasterzellen

Dieser Abschnitt beschreibt, wie eine als relevant ermittelte Rasterzelle aus einer Anfrage mit der korrespondierenden Rasterzelle zu einem Bild aus dem Datenbestand verglichen wird. Bei einer Suche hinsichtlich Farbe ergibt sich die Ähnlichkeit $cellSimilarity$ aufgrund der Berechnung von $colSimilarity$, bei einer Suche hinsichtlich Textur entsprechend aus dem Wert für $texSimilarity$. Bei einer kombinierten Farb- und Textursuche ergibt sich $cellSimilarity$ aus dem Mittelwert von $colSimilarity$ und $texSimilarity$.

Die Bestimmung der Texturähnlichkeit ist einfach, da eine Rasterzelle entweder mit einer bestimmten Textureigenschaft indiziert ist oder nicht. Der Wert für $texSimilarity(c, tex_r)$ ist 1, wenn das Bild an der Position der Rasterzelle c mit der Textureigenschaft tex_r der gesuchten Region indiziert ist. Andernfalls ist $texSimilarity(c, tex_r) = 0$.

Die Bestimmung der Farbähnlichkeit erfolgt folgendermaßen: Jede Rasterzelle überschneidet sich mit einer oder mehreren Farbregionen aus der Farbsegmentierung des Bildes. Die Funktion $fill(c, col_n)$ liefert einen Wert zwischen 0 und 1, der angibt, zu welchem Grad die Rasterzelle c mit der Farbe col_n gefüllt ist. Die Farbähnlichkeit des Bildes an der Position der Rasterzelle c mit einer Farbe col_r wird unter Berücksichtigung aller n in dem Element c vorkommenden Farben col_k berechnet durch:

$$colSimilarity(c, col_r, tol_h, tol_s, tol_b) = \sum_{k=1}^n compareCol(col_k, col_r, tol_h, tol_s, tol_b) \cdot fill(c, col_k)$$

Die Funktion $compareCol$ vergleicht zwei Farben col_k und col_r unter Berücksichtigung der eingestellten Toleranzen für die Farbe (tol_h), Sättigung (tol_s) und Helligkeit (tol_b) (vgl. Abschnitt 4.3). Das Ergebnis von $compareCol$ wird für alle Farben die in dem Bild an der Position von Rasterzelle c vorkommen mit dem Füllfaktor der Farbe gewichtet und über alle Farben aufsummiert.

4.3. Farbvergleich im HLS-Farbraum

PictureFinder vergleicht Farben im HLS-Farbraum [6], [10]. Die Übereinstimmung hinsichtlich des Farbwertes (H), der Sättigung (S) sowie der Helligkeit (L) werden separat ermittelt. Durch einen gewichteten Mittelwert werden diese drei Werte zusammengeführt und ergeben so die Gesamtübereinstimmung. Da die Farben im HLS-Farbraum nicht äquidistant über die Farbkreis verteilt sind, haben wir jeweils einen *Kernbereich* (CR*) und einen *erweiterten Bereich* (ER*) über die zugehörigen Winkel für die Farben Rot, Violett, Blau, Grün, Gelb und Orange definiert. In Tabelle 1 sind die Farben und die korrespondierenden Kern- sowie erweiterten Bereiche mit ihrem Winkel

aufgelistet. CRL steht dabei für *core range left*. Die anderen Abkürzungen sind analog zu interpretieren. Die unterschiedlichen Bereiche kommen durch die nicht äquidistanten Abstände der Farben zueinander zustande. Beispielsweise ist der Bereich der gelben Farben sehr schmal (ca. 20°). Im Gegensatz dazu belegt der Bereich der blauen Farben ca. ein Viertel des gesamten Farbkreises. Die Gewichtung der einzelnen Farben orientiert sich am Color Naming System (CNS) [1]. Für jede ganze Farbe (vgl. Tabelle 1) wird in beiden Richtungen der Kernbereich zwischen den jeweils angrenzenden Viertelfarben definiert. Der erweiterte Bereich wird begrenzt durch die angrenzenden Dreiviertelfarben (vgl. auch [8]).

Farbe/Bereich	ERL	CRL	CRR	ERR
Rot	305	345	10	25
Violett	260	285	320	350
Blau	145	200	265	290
Grün	75	105	160	215
Gelb	35	50	70	113
Orange	8	20	40	55

Tabelle 1: Grenzen der Kern- (CR*) sowie der erweiterten (ER*) Bereiche (in Grad gemäß CNS)

Beim Vergleich zweier Farbwinkel wird zunächst ein Zielbereich hinsichtlich des Referenzwinkels bestimmt:

- 1) Ist die Farbe innerhalb des Kernbereiches, dann ist der korrespondierende Kern- und erweiterte Bereich zulässig/gültig.
- 2) Ist die Farbe im linken erweiterten Bereich (ohne Kernbereich), ist das Intervall [ERL:CRR] zulässig/gültig.
- 3) Ist die Farbe im rechten erweiterten Bereich (ohne Kernbereich), ist das Intervall [CRL:ERR] zulässig/gültig.
- 4) Ist die Farbe zwischen zwei Bereichen, dann ist die Vereinigung von 1) bis 3) zulässig/gültig.

Das Intervall wird in Abhängigkeit der Toleranz, die der Benutzer für die Übereinstimmung des Farbwinkels definiert hat, verkleinert. Bei vollständiger Übereinstimmung der beiden Winkel ist das Resultat 1.0. Innerhalb des Bereichs wird die Bewertung linear zwischen 0 und 1 skaliert.

4.4. Bewertung einer Anfrage durch eine Kombination von Flächenähnlichkeiten

Für jede Region r_n , mit $n \geq 1$, wird eine Menge von möglichen Ergebnisbildern ermittelt. Wie in Abschnitt 4.1 beschrieben, bestimmt die Funktion $regionSimilarity(r_n, i)$ die Ähnlichkeit einer Region r_n mit dem Index eines Bildes i . Das Resultat (score) ist ein Wert zwischen 0 (nicht ähnlich) und 1 (maximal ähnlich). Um nun eine komplette Anfrage zu bewerten, werden die Ergebnismengen für jede einzelne Region miteinander durch *AND*, *OR* oder *soft-AND* verknüpft:

- *AND* bedeutet in diesem Zusammenhang, dass das Bild in jeder Ergebnismenge vorkommt. Die Gesamtbewertung für das Ranking ergibt sich aus dem Mittelwert über alle Bewertungen der einzelnen Regionen.
- *OR* bedeutet, dass das Bild in mindestens einer Ergebnismenge vorkommt. Die ermittelte Bewertung ist das Maximum der einzelnen Bewertungen.
- *Soft-AND* bedeutet, dass das Bild in mindestens einer Ergebnismenge vorkommt. Die Bewertung ein gewichteter Mittelwert über alle einzelnen Bewertungen der Regionen. Die einzelnen Regionen erhalten jeweils ein Gewicht ϖ_{r_n} , mit $\sum_{k=1}^n \varpi_{r_k} = 1$, welches die Wichtigkeit der Region innerhalb der Anfrage ausdrückt. Die Ähnlichkeitswerte der einzelnen Regionen bezüglich eines Bildes i werden mit ihrem Gewicht multipliziert und aufsummiert:

$$querySimScore(i) = \sum_{k=1}^n regionSimilarity(r_k, i) \cdot \varpi_{r_k}$$

Bei skizzierten Anfragen kann das Gewicht durch den Benutzer eingestellt werden. Bei Ähnlichkeitssuchanfragen werden die Gewichte heuristisch bestimmt (vgl. Abschnitt 4.5).

In der endgültigen Ergebnisliste finden sich nur Bilder wieder, deren Gesamtbewertung einen Schwellwert überschreiten (standardmäßig 0,5).

4.5. Bewertung von Ähnlichkeitssuchanfragen

Die Ähnlichkeitssuche basiert auf automatisch ermittelten Regionen, die mittels der Farbsegmentierung des Beispielbildes bestimmt werden (vgl. Abschnitt 2). Nur Regionen einer bestimmten Mindestgröße werden betrachtet. Die Textur wird dann berücksichtigt, wenn der überwiegende Teil der Farbfläche eine einheitliche Textur aufweist. Regionen ab einer bestimmten Größe bezogen auf die Gesamtfläche werden mit *AND* verknüpft. Kleinere Regionen werden mit *soft-AND* verknüpft und wirken sich daher in erster Linie auf das Ranking aus. Wir haben folgende Heuristiken hinsichtlich der zu setzenden Gewichte realisiert:

- *Gewichtung der Größe* (ω_{size}): Je größer die Region, desto wichtiger ist sie für das Ähnlichkeitsempfinden. Also wird sie stärker gewichtet.
- *Gewichtung der Sättigung/Helligkeit* (ω_{sl}): Besonders gesättigte oder helle Regionen werden stärker gewichtet.
- *Gewichtung der Position* (ω_{pos}): Zentrale Regionen sind in der Regel wichtiger als Regionen am Rand, so dass diese stärker gewichtet werden.

Die Gewichte werden multiplikativ verknüpft $\omega_{\text{all}} = \omega_{\text{size}} * \omega_{\text{sl}} * \omega_{\text{pos}}$. Eine generierte Anfrage kann an die Nutzerinteressen adaptiert werden, indem einzelne Regionen gelöscht werden oder die Gewichtung der Regionen verändert wird.

5. Ergebnisse

PictureFinder wurde auf Bildbeständen unterschiedlicher Größe getestet. Der größte bisher analysierte und durchsuchte Bildbestand umfasst ca. eine Million Bilder. Die Antwortzeiten variieren je nach Komplexität der Anfrage, liegen jedoch in einem Bereich, der sich durchaus für den professionellen Bereich eignet. Die in diesem Abschnitt gezeigten Suchergebnisse basieren auf einer Datenbasis von 12.200 Bildern. Abbildung 4 zeigt skizzierte grafische Anfragen jeweils mit den besten 10 Ergebnissen. Die Ergebnisse zeigen eine gute visuelle Übereinstimmung mit der Suchanfrage.

Ein Beispiel für die Ähnlichkeitssuche zeigt Abbildung 5. Links ist das Originalbild dargestellt, mittig die aufgrund der Farbsegmentierung des Bildes generierte grafische Anfrage. Abbildung 6 zeigt die Ergebnisse der Anfrage ohne Adaption durch den Benutzer. Die Ergebnismenge enthält neben Bildern, die tatsächlich ebenfalls Pfauen darstellen, auch eine Reihe anderer Bilder, bei denen lediglich der Hintergrund mit der Anfrage übereinstimmt.

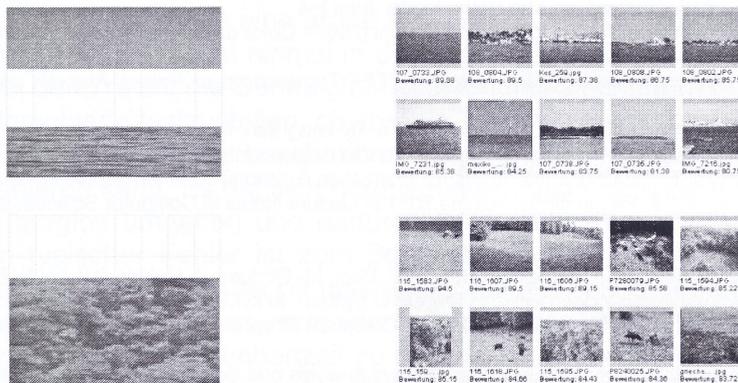


Abbildung 4: Skizzierte Suchanfragen und Ergebnisse

PictureFinder bietet dem Anwender die Möglichkeit, die generierte Suchanfrage zu adaptieren, indem er Bildregionen löscht, die für seinen Ähnlichkeitsbegriff unwichtig sind und die Toleranzen der verbleibenden Regionen variiert. Eine besonders charakteristische Eigenschaft des Pfau ist die Farbe seines Gefieders. Deswegen können die Suchergebnisse verbessert werden, indem weniger relevante Regionen entfernt werden (vgl. Abbildung 5, rechts). Bei den verbleibenden Regionen wurden die Toleranzen verringert, um relativ genau übereinstimmende Ergebnisse zu erhalten. Da uns Position und Größe, mit der der Pfau im Bild dargestellt wird, nicht wichtig ist, wurde die geforderte Dichte reduziert und gleichzeitig erlaubt, dass sich die Flächen irgendwo im Bild befinden. Die Ergebnisse der modifizierten Anfrage zeigt Abbildung 7. Es ist zu erkennen, wie der Fokus auf die mittige Region gelegt und so an die Intention des Benutzers adaptiert werden konnte.

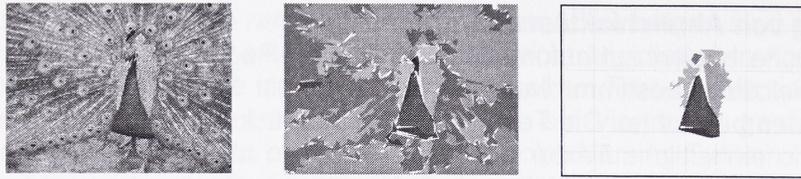


Abbildung 5: Beispielbild, automatisch generierte grafische Suchanfrage und adaptierte Suchanfrage

6. Zusammenfassung

In dieser Arbeit haben wir unser PictureFinder System vorgestellt. Dabei sind wir auf die Formulierung einer grafischen Anfrage eingegangen und haben beschrieben, wie die Ähnlichkeit über Farb- und Texturmerkmale bestimmt wird. Ein wesentlicher Vorteil der im PictureFinder System realisierten Methode zur Suche ähnlicher Bilder besteht in der Adaptierbarkeit der Anfrage. PictureFinder erlaubt dem Nutzer die Ähnlichkeit so zu formulieren, wie er sie verstehen möchte. Ein weiterer wichtiger Aspekt ist die Skalierbarkeit. Auch große Bildbestände mit über eine Million Bildern können mit PictureFinder indexiert und in akzeptabler Zeit durchsucht werden.



Abbildung 6: Ergebnisse der Ähnlichkeitssuchanfrage aus Abbildung 5 (Mitte)



Abbildung 7: Ergebnisse zur adaptierten Ähnlichkeitssuchanfrage aus Abbildung 5 (rechts)

Referenzen

- [1] T. Berk, L. Brownston, and A. Kaufman (1982): A new color-naming system for graphics languages. *IEEE CG&A*, 2(3):37–44
- [2] B. Bradshaw (2000): Semantic based image retrieval: A probabilistic approach. In *Proceedings of the eighth ACM international conference on Multimedia table of contents Marina del Rey*, pages 157–176
- [3] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik (1999): Blobworld: A system for Region-based Image Indexing and Retrieval. In *Proceedings of the Third International Conference on Visual Information and Information Systems, VISUAL'99, LNCS 1614*, pages 506 – 516, Amsterdam, The Netherlands, June 2-4
- [4] L. Cinque, G. Ciocca, S. Levialdi, A. Pellicano, and R. Schettini (2001): Color-based image retrieval using spatial-chromatic histograms. *Image and Vision Computing*, 19:979–986
- [5] Y. Cheng (1995): Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799
- [6] M.D. Fairchild (1998): *Color Appearance Models*. Addison Wesley Longman, Inc.
- [7] J. Fauqueur and N. Boujemaa. Image retrieval by regions: Coarse segmentation and fine color description. In Chang S.-K., Z. Chen, and S.-Y. Lee, editors, *Recent Advances in Visual Information Systems: 5th International Conference, VISUAL 2002, Hsin Chu, Taiwan, March 11-13, 2002. Proceedings, volume 2314 of Lecture Notes in Computer Science*, pages 24–35, Berlin, 2002. Springer.
- [8] W.D. Fellner (1992): *Computergrafik*, BI Wissenschaftsverlag
- [9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker (1995): Query by Image and Video Content: The QBIC System, In *IEEE Computer*, vol. 28, no. 9
- [10] J. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes (1993): *Computer Graphics - Practice and Principles*. Addison-Wesley Publishing Company, second edition
- [11] Th. Hermes and A. Miene (2002): Domain-independent texture analysis and description. *Künstliche Intelligenz*, 16(1):64–69
- [12] Q. Iqbal and J.K. Aggarwal. Combining structure, color and texture for image retrieval: A performance evaluation. In *16th International Conference on Pattern Recognition (ICPR)*, Quebec City, QC, Canada, August 11 - 15, 2002, vol. 2, pages 438–443, 2002.
- [13] A.K. Jain and F. Farrokhnia (1991): Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186
- [14] B. Melzer, A. Miene, and Th. Hermes (2002): Bewegungsanalyse in Bildfolgen auf der Basis eines n:m-Matchings von Farbregionen. In *Proc. of 8. Workshop Farbbildverarbeitung FarbBV 2002*, 10.-11. Oktober
- [15] N. Papamarkos and N. Nikolaou (2002): Color image retrieval using a fractal signature extraction technique. *Engineering Application of Artificial Intelligence*, 15:81–96
- [16] A. Natsev, R. Rastogi, and K. Shim (1999): WALRUS: A Similarity Retrieval Algorithm for Image Databases. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 395–406
- [17] B.G. Prasad, S.K. Gupta, and K.K. Biswas. Color and shape index for region based image retrieval. In C. Arcelli, L.P. Cordella, and G. Sanniti di Baja, editors, *4th International Workshop on Visual Form, IWVF4 Capri, Italy, May 28-30, 2001, volume 2059 of Lecture Notes in Computer Science*, pages 716–725, Berlin, 2001. Springer.
- [18] W. Skarbek and A. Koschan (1994): *Colour Image Segmentation – A Survey*. Technical Report 94-32, TU Berlin
- [19] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380