Datenbankaspekte der Bildsuche: Effizienz, Interaktion und Evaluierung

Vorlesung Bildverarbeitung 2, Teil 6
Wintersemester 2001/2002
Ullrich Köthe, FB Informatik, Uni Hamburg

Beschleunigung der Suche in Bilddatenbanken (1)

Grundproblem: traditionelle Ansätze für effiziente Suche versagen bei Nachbarschaftssuche in hochdimensionalen Räumen (Speicherkomplexität noch schlimmer)

- lineare Suche:
  $O(k\,m)$ m...Anzahl der Objekte, k...Dimension
- Verallgemeinerung des Binärbaumes auf k Dimensionen (abwechselnde Teilung aller Achsen)
  $O\left(2^k \log(m)\right)$ nur effizienter als lineare Suche solange $k < \log(m)$
- kD-Baum: Auswahl der zu teilenden Achse wie bei *median cut*
  $O(k\,m)$
- *random sampling* Algorithmus (Meiser ´93):
  $O(k^5 \log(m))$

Beschleunigung der Suche in Bilddatenbanken (2)

- **genauere Analyse: Suchzeit hängt ab von:**
  - Anzahl der Vergleiche N_V und Zeit pro Vergleich t_V
  - Anzahl der linearen Speicherzugriffe N_L und Zeit pro Zugriff t_L
  - Anzahl der *random* Speicherzugriffe N_R und Zeit pro Zugriff t_R
  - falls Daten auf Festplatte (große Datenbank): $t_R \sim 8\ t_L \ldots 10\ t_L$, falls im Speicher $t_R \sim 2\ t_L \ldots 4\ t_L$
- **falls t_V dominiert: Verfahren ist effizient, wenn es Anfragebild nicht mit allen Datenbankbildern vergleicht (aber berücksichtige konstanten Faktor!)**
- **falls Speicherzugriffszeit dominiert: effizientes Verfahren darf maximal 10% bzw. 30% der Seiten lesen**

⇨ **optimierte Baumstrukturen maximal bis ca. 10D**

Beschleunigung der Suche in Bilddatenbanken (3)

- **zur Zeit aussichtsreichster Kandidat für schnelle Suche: Clusterung der Daten**
  - lineare Suche nach richtigem Cluster
  - lineare Suche im richtigen Cluster
  - möglichst Clusterhierarchie
- **Beschleunigung der linearen Suche:**
  - falls Distanzberechnung aufwendig (z.B. *earth mover distance*):
    - Definition einer schnell zu berechnenden Approximation, die die eigentliche Distanz unterschätzt
    - keine Distanzberechnung, falls $d_{approx} >$ current-d_{min}
- **lineare Suche funktioniert auch, wenn Distanz entsprechend der Aufgabe modifiziert wird (Gewichte), aber komplexe Indexstrukturen lassen dies nicht zu**

Beschleunigung der Suche in Bilddatenbanken (4)

andererseits:

- **Laufzeitschranken der Baumstrukturen sind *worst case bounds***
  - der schlechteste Fall tritt typischerweise auf, wenn die NN-Anfrage instabil, also ohnehin fragwürdig ist
  - Verbesserung möglich, wenn realistische Verteilungen verwendet
- ***Range Queries* sind einfacher (sublineare Indizes)**
  - Suche Bilder, die höchstens δ von der Anfrage entfernt sind
  - aber: wie wählt man δ günstig?
- **näherungsweise NN-Anfragen sind einfacher (sublineare Indizes)**

⇨ **notwendig: besseres Verständnis der Daten und der Nutzeranforderungen in realistischen Anwendungen**

Kombination mehrerer Distanzen (1)

- **oft besteht die Ähnlichkeitsfunktion aus mehreren unabhängigen Komponenten, z.B. Farbhistogramm, Textursignatur, Richtungshistogramm etc.**

  ⇨ **Kombination zu einer Distanz notwendig**

  ⇨ **Einzeldistanzen müssen vergleichbar gemacht werden (Normierung)**
- **Möglichkeit 1:** Benutzung der Rangordnung
  - suche in jeder Komponente unabhängig, ordne die Antworten
  - erstelle die kombinierte Rangordnung aufgrund des mittleren Rangs jedes Bildes über alle Teildistanzen:

$$d_{combined}(x_i, q) = \sum_k \text{rang}(d_k(x_i, q))$$

Kombination mehrerer Distanzen (2)

- **Möglichkeit 2:** gewichtete Summe der Einzeldistanzen

$$d_{combined}(x_i, q) = \sum_k w_k d_k(x_i, q)$$

  - Standardgewichte: $E[w_k\, d_k(x_i, x_j)]$ für alle Komponenten konstant
- **Möglichkeit 3:** Umrechnung der Distanzen in Wahrscheinlichkeiten (gelernt aus Trainingsdaten)
  - Wahrscheinlichkeit, daß x_i und q bezügl. k ähnlich sind, z.B.:

$$p_{ähnlich,k}(x_i, q) = e^{-d_k(x_i,q)^{B_k}/\sigma_k^{B_k}}, \quad B_k, \sigma_k: \text{ Parameter}$$

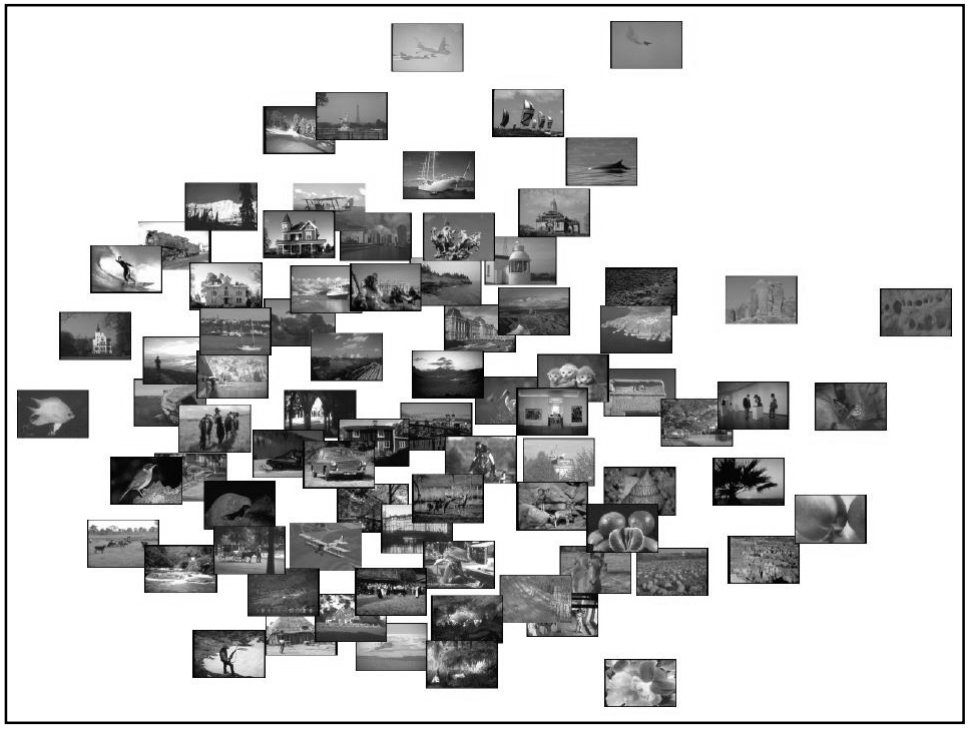
  - Kombination: probabilistisches „und“ bzw. „oder“

$$p_{ähnlich}(x_i, q) = \Pi_k\, p_{ähnlich,k}(x_i, q)$$

$$p_{ähnlich}(x_i, q) = 1 - \Pi_k (1 - p_{ähnlich,k}(x_i, q))$$

Optimale Präsentation der Suchergebnisse (1)

- **üblich: Präsentation der k ähnlichsten Bilder entsprechend ihrer Rangordnung**
  - Vorteil: gut passende Bilder fallen oft sofort ins Auge
  - Nachteil: Bilder, die sich untereinander ähneln, sind nicht unbedingt nebeneinander
- **Alternative: berücksichtige gegenseitige Ähnlichkeit bei der Präsentation**
  - distanzerhaltende Dimensionsreduktion für die ersten k Bilder auf 2 Dimensionen
  - Präsentation im neuen Koordinatensystem – „*star field display*“
  - Benutzer bevorzugen geordnete Präsentation in den meisten Fällen (aber manchmal „Maskierung“ der besten Antwort)

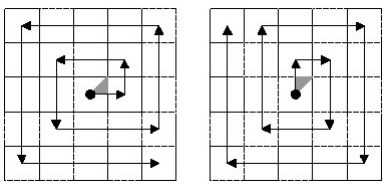


Optimale Präsentation der Suchergebnisse (2)

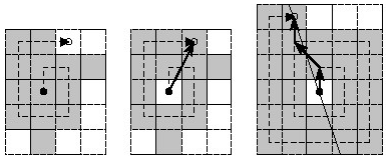
- **Bestimmung der 2D-Koordinaten:**
  - *multi-dimensional scaling*
  - FastMap-Algorithmus (müßte für Präsentationszwecke reichen, aber keine systematischen Untersuchungen)
  - wenn Distanz aus mehreren Komponenten zusammengesetzt ist, benutze die zwei wichtigsten Komponenten als Achsen, z.B. Farbähnlichkeit vs. Texturähnlichkeit oder Farbähnlichkeit vs. Richtungshistogramm
- **Problem: Thumbnails sind relativ groß ⇨ meist treten Überlappungen auf**
  - Verdeckung stört sehr
  - Verschiebung der Bilder notwendig, so daß Überlappung minimiert oder ausgeschlossen wird

Optimale Präsentation der Suchergebnisse (3)

- **drei Ansätze für überlappungsfreie Darstellung des *star field display:***
  - Lösung des multi-dimensional scaling Problems mit der Nebenbedingung, daß nur ganzzahlige Koordinaten erlaubt sind ... schwierig
  - siehe vorige Seite: ordne Objekt der nächstgelegenen Zelle zu; Konfliktlösung ähnlich wie beim Hashing: falls beste Zelle besetzt, wähle die nächstbeste usw.



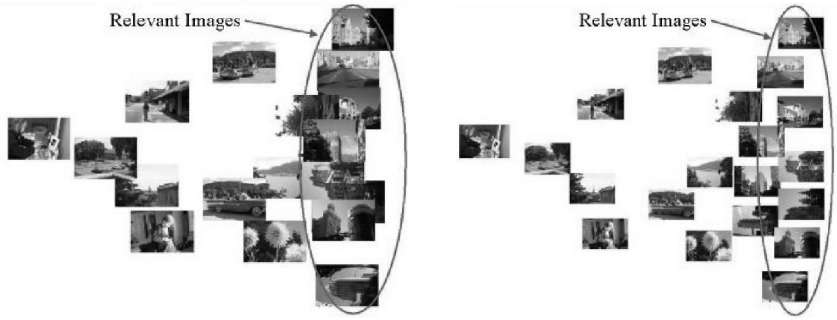
Varianten der Konfliktlösung



Optimale Präsentation der Suchergebnisse (4)

– Verschieben und Verkleinern der Thumbnails: Optimieren einer Zielfunktion, die Überlappung und Verschiebung gegeneinander wichtet, so daß Minimalgröße nicht unterschritten wird:

$$\text{minimiere } J = \text{overlap}(p) + \lambda \, \text{translation}(p)$$

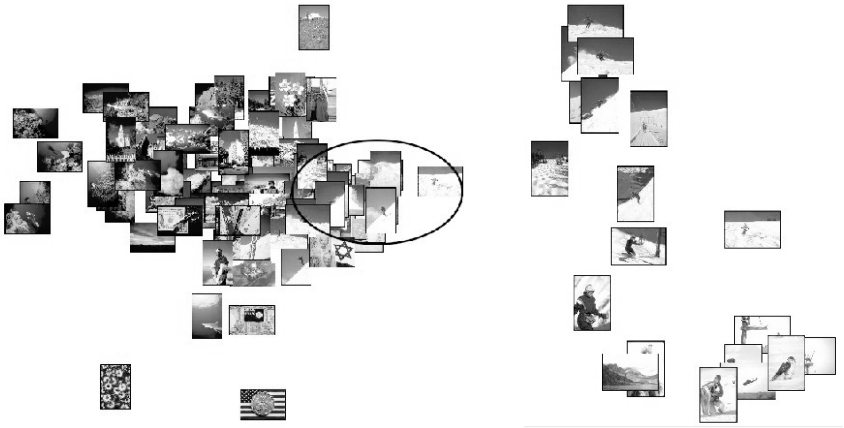


Relevanz-Feedback (1)

- **Wie kann man die Suche an die aktuelle Aufgabe anpassen?**
  - Problem: Übersetzung von abstrakten Konzepten in Steuerparameter für die Bildanalyse ist schwierig/unmöglich
  - ⇨ System muß günstige Einstellungen durch Feedback lernen
- **Idee des Relevanz-Feedback:**
  - Präsentation einer ersten Antwortmenge
  - Markieren relevanter/irrelevanter Bilder durch den Benutzer
  - automatische Modifikation der Ähnlichkeitsfunktion
  - Präsentation eines neuen Suchergebnisses
  - Iteration bis der Benutzer das gesuchte Bild findet

Relevanz-Feedback (2)

„star field display“ hilft bei der Auswahl relevanter Bilder



Relevanz-Feedback (3)

- **zwei Strategien**
  - Modifikation der Ähnlichkeitsfunktion
  - Modifikation der Anfrage
- **einfachste Realisierung:**
  - wenn die Ähnlichkeitsfunktion eine gewichtete Summe aus verschiedenen Teilen ist ⇨ Anpassen der Gewichte

$$d(x_i, q) = \sum_{k=1}^{K} r_k \, w_k \, d_k(x_i, q)$$

r_k: Relevanzgewichte
w_k: Normierung

  - setze zunächst $r_k = 1/K$ (gleiche Gewichtung aller Verfahren)
  - betrachte die ersten N Bilder, die jedes Verfahren zurückliefert:

$$RT_k^{(t)} = \{B_1, \ldots, B_N\}_k^{(t)} \quad t: \text{ Nummer der aktuellen Iteration}$$

Relevanz-Feedback (4)

– **Aktualisierung der Gewichte nach Rui et al.:**

$$score(B) = \begin{cases} 3 & \text{wenn } B \text{ sehr relevant} \\ 1 & \text{wenn } B \text{ relevant} \\ 0 & \text{wenn keine Angabe zu } B \\ -1 & \text{wenn } B \text{ irrelevant} \\ -3 & \text{wenn } B \text{ totel irrelevant} \end{cases}$$

$$r_k' = r_k^{(t)} + \sum_{B \in RT_k^{(t)}} score(B)$$

$$r_k^{(t+1)} = \frac{r_k'}{\sum_k r_k'}$$

Relevanz-Feedback (5)

– **Aktualisierung der Gewichte nach Aksoy et al.:**

Varianz von Merkmal k im Antwortset der Iteration t :

$$\left(\sigma_k^{(t)}\right)^2 = E\left[\left(Feature_k\right)^2\right]^{(t)} - \left(E\left[Feature_k\right]^{(t)}\right)^2$$

analog Varianz von Merkmal k für die relevanten Bilder:

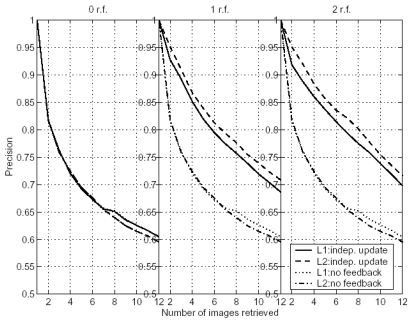
$$\left(\sigma_{k,relevant}^{(t)}\right)^2 = E\left[\left(Feature_{k,relevant}\right)^2\right]^{(t)} - \left(E\left[Feature_{k,relevant}\right]^{(t)}\right)^2$$

$$r_k' = \frac{\sigma_k^{(0)}}{\sigma_{k,relevant}^{(t)}}, \qquad r_k^{(t+1)} = \frac{r_k'}{\sum_k r_k'}$$

– r_k ist groß, wenn σ_k groß und $\sigma_{k,\ relevant}$ klein

Relevanz-Feedback (6)

- **Verbesserung der Verfahren:**
  - nicht nur die Verfahren variabel gewichten, sondern auch die Summanden innerhalb jedes Verfahrens
  - Menge der relevanten Bilder über alle Iterationen akkumulieren
- **Ergebnisse nach 2 Iterationen: ca. 20% höhere *precision***



Distance	0 rf	1 rf	2 rf
L_1	0.60	0.69 (13.53%)	0.70 (1.71%)
L_2	0.60	0.71 (19.03%)	0.72 (1.06%)

Relevanz-Feedback (7)

- **Bestimmung optimaler Gewichte:**
  - Fisher´s Lineare Diskriminanten: maximale Klassentrennung
  - 2 Klassen: relevant und irrelevant mit bekannter Zuordnung

Vektor der Antworten der Verfahren für Bild i

$\vec{x}_i = [x_1,...,x_K]_i$ (x_k: Merkmale oder Distanzen)

$\vec{\mu}_{relevant}$, $\vec{\mu}_{irrelevant}$: Klassenmittelwerte

$$S_W = \sum_{i \in relevant} (\vec{x}_i - \vec{\mu}_{relevant})(\vec{x}_i - \vec{\mu}_{relevant})^T + \sum_{i \in irrelevant} (\vec{x}_i - \vec{\mu}_{irrelevant})(\vec{x}_i - \vec{\mu}_{irrelevant})^T$$

$$\vec{r} = S_W^{-1}(\vec{\mu}_{relevant} - \vec{\mu}_{irrelevant})$$

- **andere Klassifikatoren möglich (z.B. *Support Vector Machines*)**

Relevanz-Feedback (8)

- **Ändern der Query:**
  - *Multipoint Query*: betrachte alle relevanten Bilder als Anfrage

$$d_{combined}(B)^{\alpha} = \begin{cases} 0 & \text{wenn } \alpha<0 \text{ und } \exists i : d(B,B_i) = 0 \\ \dfrac{1}{N_{relevant}} \sum_{i=1}^{N_{relevant}} d_k(B,B_i)^{\alpha} & \text{sonst} \end{cases}$$

    - falls $\alpha > 0$: fuzzy „und“, $\alpha < 0$: fuzzy „oder“
  - *Query Point Movement*: neue Anfrage als Mittelwert der relevanten Bilder (auch in Verbindung mit Merkmalsgewichten)

$$\vec{x}_i = [x_1,...,x_K]_i \quad \text{Merkmalsverktor von Bild } i$$

$$\vec{q}^{(t+1)} = \frac{1}{N_{relevant}} \sum_{i \in relevant} \vec{x}_i$$

Filter Image Browsing

- **Beobachtung: nach der ersten Iteration ist ein großer Teil der Datenbank unerreichbar ⇨ Feedback nützt nichts, wenn relevante Bilder dort enthalten**

⇨ begrenze den Prozentsatz der Bilder, der in einer Iteration unerreichbar werden kann (Vendrig et al.)

1. definiere Reduktionsfaktor ϱ = 0.25 (laut Paper)
2. wähle $RT^{(0)}$ so, daß gesamte Datenbank erreichbar (falls die originale Antwort dies nicht gewährleistet, füge Bilder hinzu)
3. setze $Covering^{(t+1)} = \varrho\ Covering^{(t)}$
4. bestimme $RT^{(t+1)}$ durch *Multipoint Query* mit relevanten Bildern
5. wenn aus $RT^{(t+1)}$ weniger als $Covering^{(t+1)}$ erreichbar ist, füge Bilder hinzu, dann gehe zu 3.

- ***recall* um 20% erhöht**

Evaluierung der Bildsuche (1)

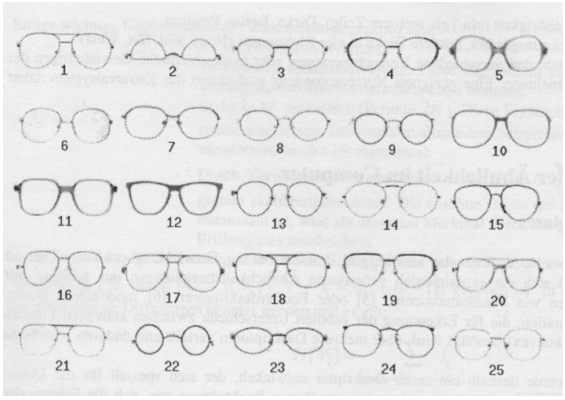
- **Evaluierung ist schwierig, weil die Bedeutung des Bildes (und damit die Ähnlichkeitsfunktion) von der Aufgabe und vom Kontext abhängt**
- **welche Datenbank verwendet man?**
  - Corel-Bildersammlung:
    - mehrere 10000 Farbbilder hoher Qualität
    - starke Clusterung (100 Bilder pro Thema), Copyrightprobleme
  - spezielle Bildarchive:
    - Pressearchive, medizinische Archive, Kunstgeschichte
    - gut, aber keine Vergleichsmöglichkeit
  - freie Bildsammlungen (z.B. VisTex, U Washington): relativ klein (<1000) und relativ speziell (z.B. Texturen)

Evaluierung der Bildsuche (2)

- **wie definiere ich das erwartete Ergebnis?**
  - benötige realistisches Anwendungsszenarium
    - psychologische Experimente im Kontext eines Szenariums
    - ist CBR besser als herkömmliche Verfahren? (Geschwindigkeit und Recall im Vergleich zu Schlagwort-suche und Browsing)
    - wenig bekannt über die tatsächlichen Bedürfnisse der Nutzer
  - experimentelle Bestimmung des menschlichen Ähnlichkeits-empfindens
    - wie ordnen Nutzer die Bilder relativ zu einem Anfragebild?
    - wie stark hängt die Ordnung von der Aufgabe ab?
    - wie stark schwankt sie von Person zu Person und während verschiedener Sitzungen?

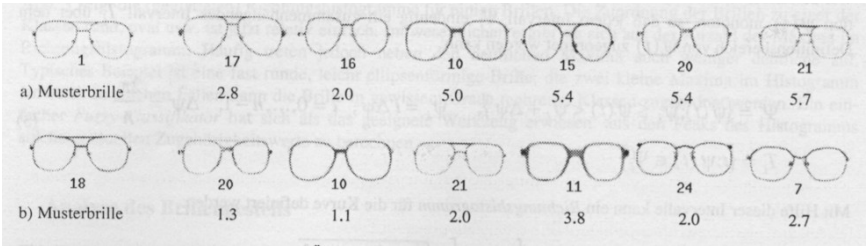
Evaluierung der Bildsuche (3)

- **Beispiel: Brillendatenbank**
  - welche Merkmale sind wichtig?



Evaluierung der Bildsuche (4)

- **Formähnlichkeit:**
  - nicht rotationsinvariant
  - wichtig: Anzahl und Lage der „Ecken“, Rundung
  - extreme Standardabweichung des Ranges bei der Bewertung durch verschiedene Personen (bis 5 Positionen!)



Evaluierung der Bildsuche (5)

- **umfassendes, anwendungszentriertes Konzept der Bewertung: Pressearchive (Markkula, Sormonen ´00)**
- **Bestimmung von *ground truth:***
  - Aufgabe: Suche Bild, das einen vorgegebenen Artikel illustriert
    - Entwicklung einer Illustrationsidee
    - Auswahl eines passenden Bildes mit herkömmlichen Mitteln
    - Zusammenstellen einer zugehörigen „thematischen Sammlung“ mit herkömmlichen Mitteln (z.B. „Kosovoflüchtlinge“)
    - Klassifikation der Sammlung in „ähnliche“ und „unähnliche“ bezüglich des Kernbildes durch mehrere VP
    - Aufstellen einer Kriterienliste für jede Bewertung

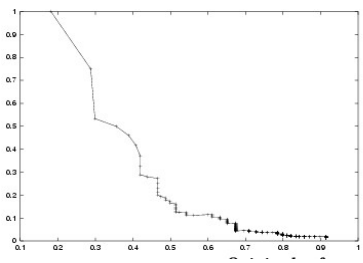
Evaluierung der Bildsuche (6)

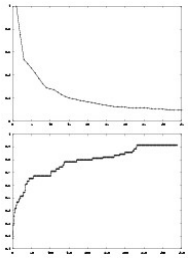
- **Ergebnis: Menge von Musteranfragen**
  - Photodatenbank mit 50000 Bildern, daraus:
    - 45 thematische Sammlungen mit je 60-400 Bildern
    - je 1-25 sind als zum Kernbild ähnlich klassifiziert (Mittel: 5)
  - menschliche Ähnlichkeitskriterien:
    - low-level (Farbe, Komposition, Blickwinkel, Entfernung, ...)
    - intermediate (Objekttyp, Objektname, Ort, Aktion, ...)
    - high-level (Atmosphäre, Gefühlsausdruck, Symbol, Konzept)
  - Konsistenz der Ähnlichkeitsmenge für verschiedene VP:
    - mit Berücksichtigung der Aufgabe 60% (16%-94%)
    - ohne Berücksichtigung der Aufgabe 48% (0%-88%)

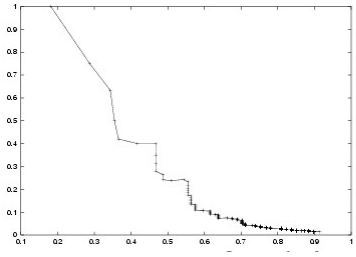
Evaluierung der Bildsuche (7)

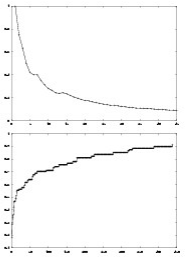
- **Wie soll die Performance beschrieben werden?**
  - Rang des ersten korrekten Bildes
  - durchschnittlicher Rang der korrekten Bilder (normiert)
  - Precision: $$precision = \frac{|relevant \cap retrieved|}{|retrieved|}$$
    - Precision bei Größe 20, 50, Gesamtzahl relevant
    - average Precision: miß Precision, wenn ein weiteres korrektes Bild angezeigt wird und mittle diese Werte
  - Recall: $$recall = \frac{|relevant \cap retrieved|}{|relevant|}$$
    - Recall für Precision 0.5 und für 100 Antwortbilder
  - Precision/Recall-Graph

Evaluierung der Bildsuche (8)









Precision, Recall und PR-Graphen aus dem Projekt

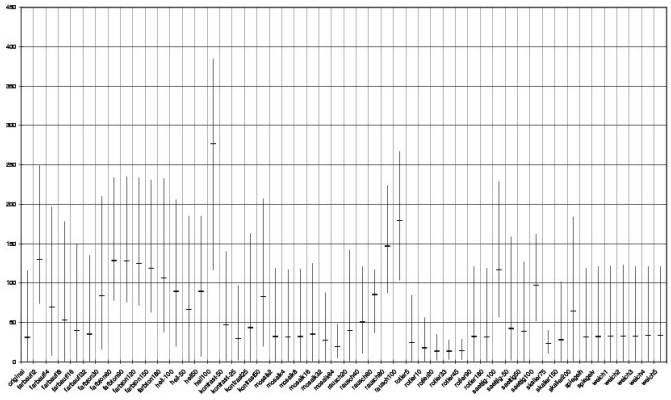
LUV

Coocurrence

Evaluierung der Bildsuche (9)

Messung der Veränderung von precision und recall für bestimmte Bildstörungen, Vergleich der Verfahren

Luv-Farb-histogramm robust gegen Rauschen, geometrische Transforma-tionen, aber anfällig für Farbver-schiebungen



Evaluierung der Bildsuche (10)

- **Ergebnisse der Pressearchivstudie**
  - Excalibur und ein eigenes System wurden getestet
  - gemessen average Rank (ca. 50), average Precision (ca. 20%) und weitere Maße
  - Ergebnis:
    - Bildsuche war signifikant besser als zufällige Sortierung
    - low-level Merkmale sind mit menschlicher Ähnlichkeitsbewertung korreliert
    - Aspektverhältnis der Bilder ist überraschend wichtig
    - keine Parameterkombination war für alle Beispiele die beste

    ⇨ Notwendigkeit des Relevance Feedback

Evaluierung der Bildsuche (11)

Evaluierung des Relevanz-Feedback (La Cascia et al.):

- Datenbank mit automatisch extrahierten Schlüsselwörtern (WWW), Farbhistogrammen, Richtungshistogrammen
- die besten 100 Antworten werden jeweils angezeigt
- Generierung der „Page 0“ durch Schlüsselwortsuche
- weitere Suche durch Relevanz-Feedback (maximal 4 Iterationen)

- **Aufgabe: Finden eines vorgegeben Bildes (das nicht selbst als Anfragebild verwendet werden darf)**
- **Ergebnis:**
  - Datenbankgröße 10 000: 35% der Testbilder konnten gefunden werden, bei Größe 100 000 nur noch 18% (in 4 Iterationen)
  - durchschnittlich wurden 2 Iterationen benötigt
  - ohne Relevance-Feedback wurden nur 20% bzw. 9% gefunden