Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

Part 2: Similarity and Dissimilarity Measures

Part 3: Efficient Similarity Query Processing

Thomas Seidl, Christian Beecks, RWTH Aachen

Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

- Einführung Ähnlichkeitssuche
- Anfragetypen zur Ähnlichkeitssuche
- Repräsentationen für Multimedia-Objekte

Beispiel Ähnlichkeitssuche für Bilder

- Aufgabe
  - Archiv mit 10^x Bildern (Katalog, Flickr, WWW, …)
  - Frage: Ist dort ein bestimmtes Kunstwerk abgebildet?

- Herausforderung:
  - Suche nach “identischer Binärrepräsentation” ist zu strikt, Abweichungen sind möglich

- Abweichungen im Beispiel
  - unterschiedliche Größe (Skalierung; Auflösung; Farbtiefe)
  - Unterschiedl. Tönung der Farben
  - abweichende Ausschnittbildung
  - hinzugefügter Rand / Beschriftung
  - …

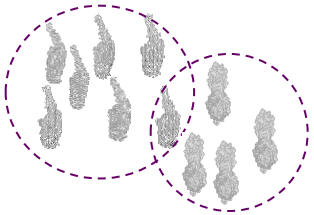


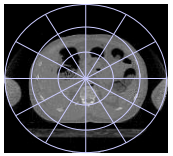





Anwendungsbeispiele

- Life Sciences
  - Bioinformatik: Genomics, Proteomics
  - Medizin: Suche in Bildarchiven zur Diagnoseunterstützung





- Maschinenbau
  - KFZ-Entwicklung: Reduktion von Wiederholteilen (Kosten!)
  - Flugzeugbau: Räumliche Einbauuntersuchung

- Multimedia
  - Bild- und Videoarchive: Content-Based Retrieval

- E-Commerce
  - Customer Relationship Management, Recommendation Systems, Web Usage Mining

Suche in Multimedia-Datenbanken

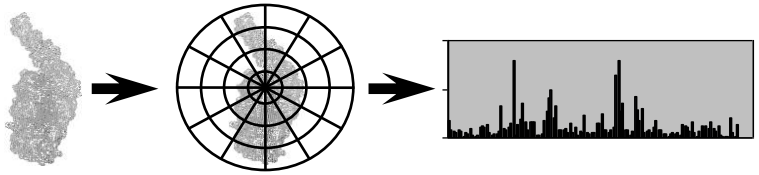
- Multimediale Objekte
  - Bilder, Videos, Audio, Graphen, Sequenzen, Formen, ...
- Suchen in Multimedia-Datenbanken
  - Primärschlüssel (z.B. Dateiname) nur begrenzt hilfreich
  - Erschließung über Schlagwörter und/oder Metadaten
    - großer Aufwand für manuelle Verschlagwortung
    - Schlagwörter decken nur begrenzte Aspekte ab
    - keine Unterstützung für Anfragen bezüglich nicht erfaßter Aspekte
  - Inhaltsbasierte Suche
    - Suchkriterien werden automatisch aus den Objekten extrahiert
    - Möglichkeiten: Farben, Texturen, Formen, Struktur, ...

Inhaltsbasierte Ähnlichkeitsmodelle

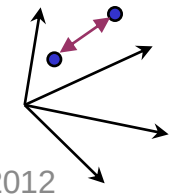
- Ziel: Inhaltsbezogene Suche
  - Suche zu einem Anfrageobjekt ähnliche Objekte in der DB (oft subjektiv)

- Phänomen der Ähnlichkeit durch Ähnlichkeitsmodell beschreiben
  - Formalisierung von Ähnlichkeit durch mathematische Definitionen
  - Qualität eines Ähnlichkeitsmodells oft nicht mathematisch messbar
  - Wie sinnvoll ein Ähnlichkeitsmodell ist, hängt von der Anwendung ab
  - Manchmal rechtfertigen Einsichten aus anderen wissenschaftlichen Disziplinen die Wahl eines Modells

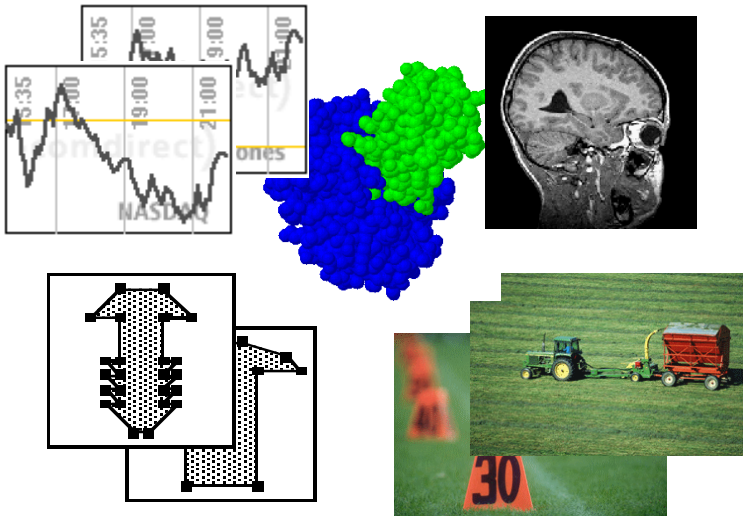
- Hier: Featurebasierte Ähnlichkeit
  - Schritt 1: Feature Transformation



  - Schritt 2: Bewerten der Ähnlichkeit



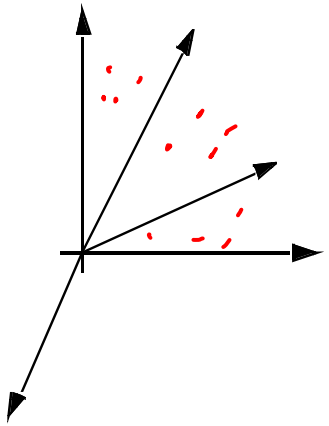
Merkmalsbasierte Ähnlichkeit: Features



Feature-Transformation

- Histogramm-Bildung
- Moment-Invarianten
- Rechteck-Abdeckung
- Formhistogramme
- Fourier-Transformation
- Signaturen
- usw.

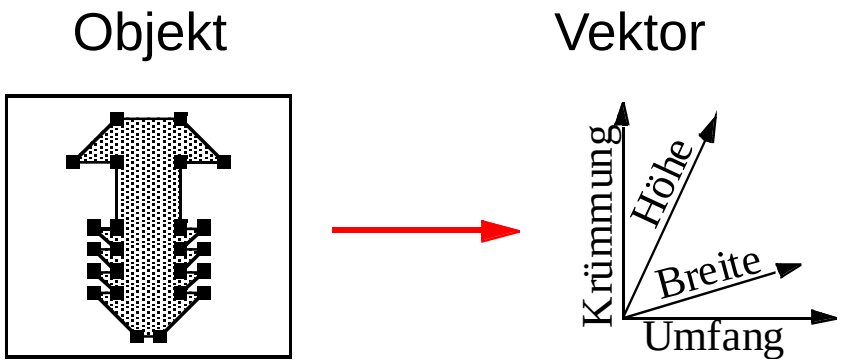
Feature-(Vektor)raum



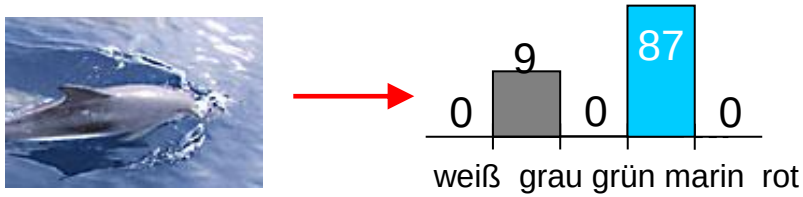
- Feature-Transformation, Feature-Extraktion
  - Abbildung komplexer Objekte in einen hochdimensionalen Feature-Raum
  - Objekte werden z.B. durch Feature-Vektoren charakterisiert
  - Ein einzelner Punkt im Feature-Raum kann verschiedene zueinander ähnliche Objekte repräsentieren
  - Features müssen anwendungsspezifische Anforderungen erfüllen

Beispiele für Featuretransformation

- Beispiel geometrische Objekte
  - Länge, Breite: skalare Werte
  - Krümmungsverlauf: Sequenz (z.B. LWL-Codierung = Länge-Winkel-Länge-…)



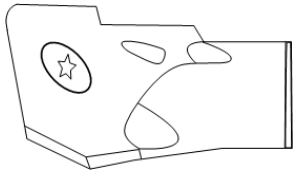
- Beispiel Farbbilder
  - Farbhistogramm: Häufigkeit von Pixeln bestimmter Farbtöne
  - Typische Dimensionen: 64, 256



  - Farbsignaturen: Flexible Anpassung an Bildinhalte
  - Eigenschaften werden als „Menge“ gespeichert

Beispiele für Featuretransformation

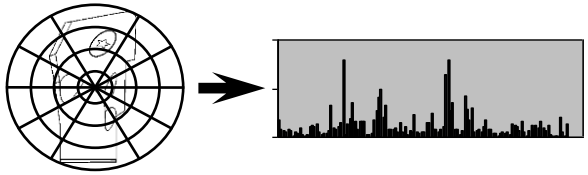
- Beispiel: Vergleich von technischen Zeichnungen



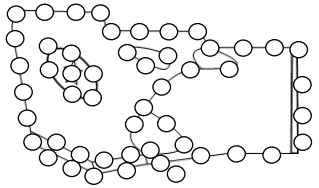
technische Zeichnung

Form

Formhistogramme

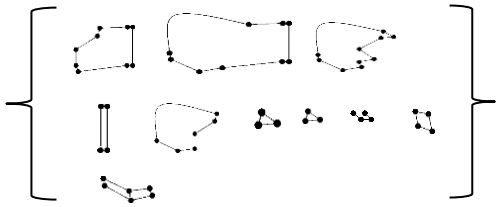


Voxeldarstellung

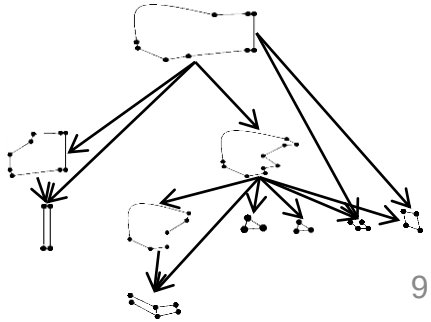


Zusammensetzung

Mengen

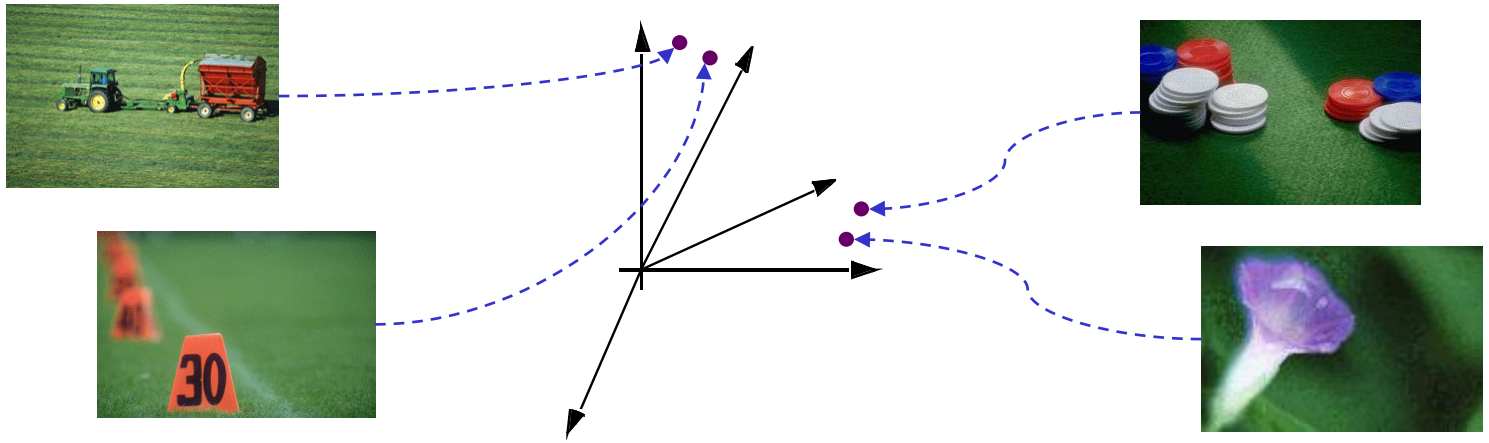


Graphen



Merkmalsbasierte Ähnlichkeit: Distanzen

- Distanzbasierte Ähnlichkeit
  - Annahme: Ähnliche Objekte → ähnliche Repräsentationen
  - Geringer Abstand der Repräsentationen = hohe Ähnlichkeit der Objekte
  - Unterschiedliche Definitionen von Distanzen (Euklid, etc.)



- Alternativen
  - Ähnlichkeitswerte (Scoring, o.ä.) statt Distanzen

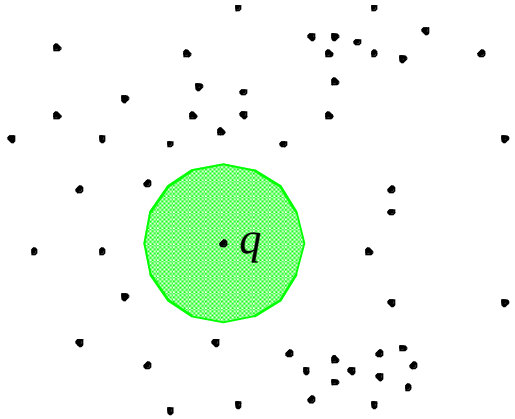
Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

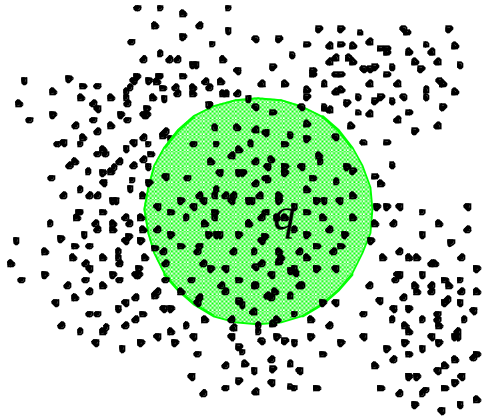
- Einführung Ähnlichkeitssuche
- Anfragetypen zur Ähnlichkeitssuche
- Repräsentationen für Multimedia-Objekte

Bereichsanfragen

Anfrageparameter	Anfrageobjekt q, maximaler Ähnlichkeitsabstand ε
Ergebnismenge	$sim_\epsilon(q) = \{o \in DB \mid d(o,q) \leq \varepsilon\}$
Anzahl der Ergebnisse	im vorhinein unbekannt, zwischen 0 und |DB| → Problem: wie ε wählen?
Ergebnisbereich	spezifizierter Bereich ε



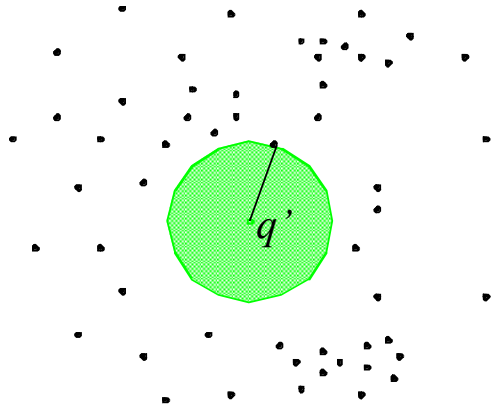
ε klein: keine (wenige) Ergebnisse



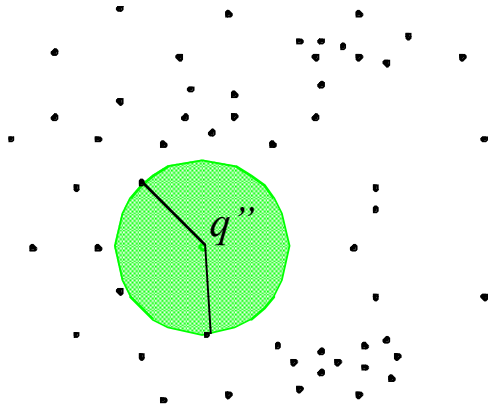
ε sehr groß: sehr viele Ergebnisse

Nächste-Nachbarn-Anfrage

Anfrageparameter	Nur Anfrageobjekt q
Ergebnismenge	$NN(q) = \{o \in DB \mid \forall o' \in DB: d(o,q) \leq d(o',q)\}$, bzw. $NN(q) = SOME\ \{o \in DB \mid \forall o' \in DB: d(q,o) \leq d(q,o')\}$
Anzahl der Ergebnisse	Genau 1 bzw. mindestens 1
Ergebnisbereich	im vorhinein unbekannt, $\varepsilon_1 = \min\{d(o,q) \mid o \in DB\}$



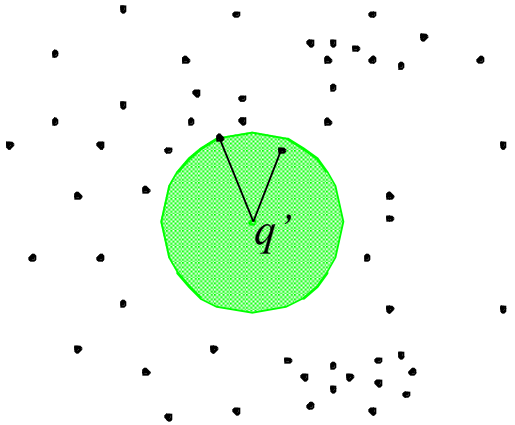
Eindeutiger nächster Nachbar



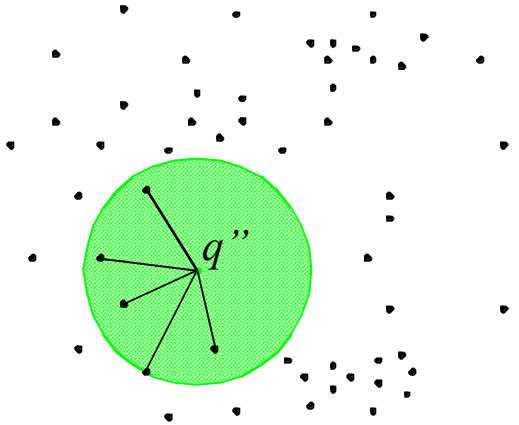
mehrere nächste Nachbarn

k-Nächste-Nachbarn-Anfrage

Anfrageparameter	Anfrageobjekt q, Anzahl gewünschter Ergebnisse k
Ergebnismenge	Kleinste Menge $NN_q(k) \subseteq DB - \{q\}$ mit $\left\|NN_q(k)\right\| \geq k$ und: $\forall o \in NN_q(k)\ \forall o' \in (DB - NN_q(k)): d(o,q) < d(o',q)$
Anzahl der Ergebnisse	k (mindestens, oder auch genau k)
Ergebnisbereich	Im vorhinein unbekannt, $\varepsilon_k = \max\{d(o,q) \mid o \in NN_q(k)\}$



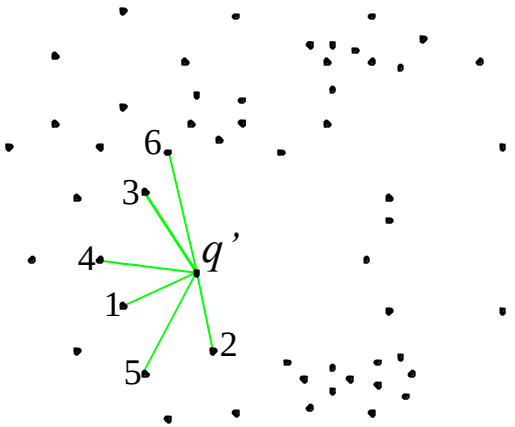
k=2



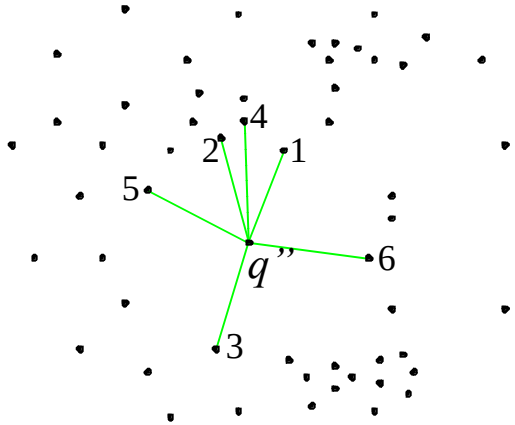
k=5

Inkrementelle Nächste-Nachbarn-Anfrage

Anfrageparameter	Anfrageobjekt q, Aufrufe von getnext(k_i)
Ergebnismenge	$NN_q(k)$ mit $k = \sum_{i=1}^{h} k_i$ für h Aufrufe von getnext(k_i), d.h. Folge $\langle NN_q(k_1), NN_q(k_1 + k_2), NN_q(k_1 + k_2 + k_3), \ldots \rangle$
Anzahl der Ergebnisse	$k = \sum_{i=1}^{h} k_i$ für h Aufrufe von getnext(k_i)
Ergebnisbereich	im vorhinein unbekannt, $\varepsilon_k = \max\{d(o,q) \mid o \in NN_q(k)\} = d(o_k, q)$



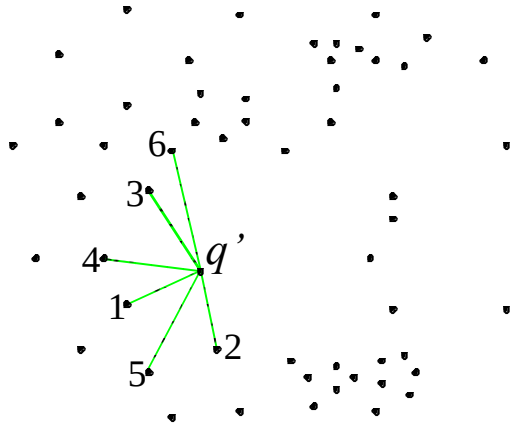
6x getnext(1) um q‘



6x getnext(1) um q“

Ranking-Anfrage

Anfrageparameter	Anfrageobjekt q
Ergebnismenge	Funktion $o: \{1, \ldots, \lvert DB \rvert\} \rightarrow DB$ sortiert Datenbank nach einer vorgegebenen Distanzfunktion *d*: $\forall i, j \in \{1, \cdots, \lvert DB \rvert\}: i < j \Rightarrow d(o(i), q) \leq d(o(j), q)$
Anzahl der Ergebnisse	|DB|, bzw. k < |DB| im Falle eines partiellen Rankings
Ergebnisbereich	Abstand von Anfrageobjekt zum weitesten Objekt



Resultierende Folge:
$\langle o_1, o_2, o_3, o_4, \ldots \rangle$

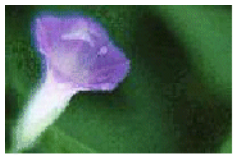
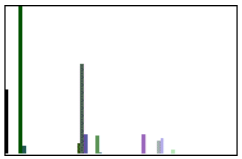
Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

- Einführung Ähnlichkeitssuche
- Anfragetypen zur Ähnlichkeitssuche
- Repräsentationen für Multimedia-Objekte

Inhaltsbasierte Merkmale von Bildern

- Farbe
  - Farbhäufigkeitsverteilungen [HSE+ 95]
  - Eigenschaft einzelner Pixel

- Textur
  - Beschaffenheit von Bildsegmenten (Pixelregionen) (z.B. Holzmaserung, Kieselsteine, Karomuster)
  - Evaluierung verschiedener Distanzfunktionen [PBRT 99]


- Formen (Konturen)
  - Algebraische Moment-Invarianten [TC 91] [FBF+ 94]
  - Pixelbasierte Ähnlichkeitsmodelle [WJ 96] [AKS 98]
  - Vgl. Kapitel „Ähnlichkeit von geometrischen Formen“



- Systeme zur Inhaltsbasierten Suche
  - QBIC: Query By Image (and Video) Content, IBM Almaden Research Center
  - ImageMiner: Technologie-Zentrum Informatik, Universität Bremen
  - VisualSeek: Center for Telecom Research, Columbia Univ., NY
  - MARS: Multimedia Analysis and Retrieval System. U. Illinois/Urbana-Champaign
  - Surfimage: INRIA Rocquencourt, France
  - … und viele mehr!

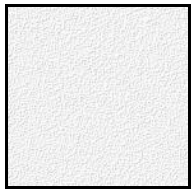
Texturen in Bildern (Modell aus QBIC)

Faloutsos, Barber, Flickner, Hafner, et al.: Efficient and Effective Querying by Image Content. JIIS 3, 231-262, 1994.
Tamura, Mori, Yamawaki: Textural Features Corresponding to Visual Perception.. IEEE TSMC 8(6), 460-473, 1978.

- Textur beschreibt Beschaffenheit von Bildsegmenten (dargestellte Oberflächen)

- Gerichtetheit, Orientiertheit (Directionality)
  - Vorhandensein von (Vorzugs-)Richtungen
  - Beispiel: Mauerfugen versus Kieselsteine
    aus Verteilung der Gradientenrichtungen in den Bildern


- Kontrast (Contrast)
  - Lebendigkeit (Unruhe) eines Musters
  - Beispiel: weiße Wand versus Sand
  - Berechnung aus Varianz im Grauwerthistogramm



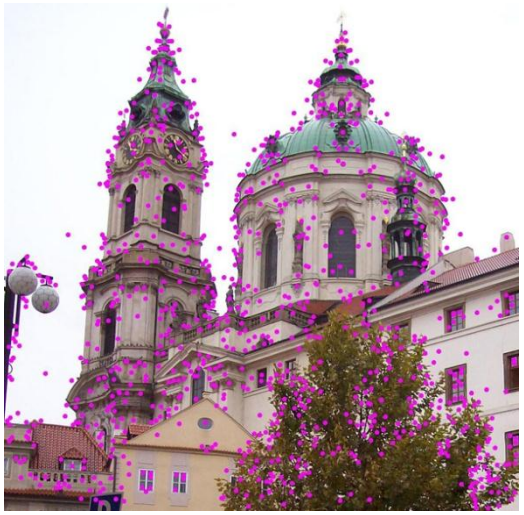

- Granularität (Coarseness)
  - Größenordnung der Textur
  - Beispiele: Sand vs. Kieselsteine; feine vs. grobe Mauer
  - Berechnung durch über das Bild verschobene Fenster
    unterschiedlicher Größe


Weitere Typen von Merkmalen

Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek: Evaluating Color Descriptors for Object and Scene Recognition. IEEE TPAMI 2010.

D. G. Lowe: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2004.

- SIFT (Scale Invariant Feature Transform):
  - Bilder liegen im Grauwertformat vor, d.h. jedes Pixel wird auf seinen entsprechenden Grauwert [0, 255] abgebildet
  - SIFT Deskriptor beschreibt die lokale Form einer Region um einen interessanten Punkt basierend auf Gradienten Informationen
  - Invariant gegenüber Translation, Rotation und Skalierung



- Es gibt viele Varianten:
  - C-SIFT
  - RGB-SIFT
  - OpponentSIFT
  - ...

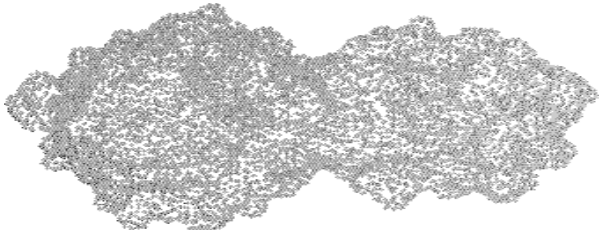
Objekte als Verteilungen von Features

- Statistische Formalisierung des Objektbegriffs
  - Sei ein $\mathcal{F}$ (beliebiger) Featureraum
  - Objektfunktion O ordnet jedem Feature ein nicht-negatives Gewicht zu:

$$O: \mathcal{F} \rightarrow \mathbb{R}_0^+$$

  - Statistische Sicht: Objekt ist (Beobachtung einer) Verteilung über dem Featureraum, O ist Wahrscheinlichkeitsdichtefunktion (pdf).

- Beispiele
  - Punktewolke als charakteristische Funktion $W: \mathbb{R}^3 \rightarrow \{0,1\}$
  - Pixelbild als Abbildung von Pixel in einen Farbraum, $\mathrm{P}: \Im_w \times \Im_h \rightarrow RGB$



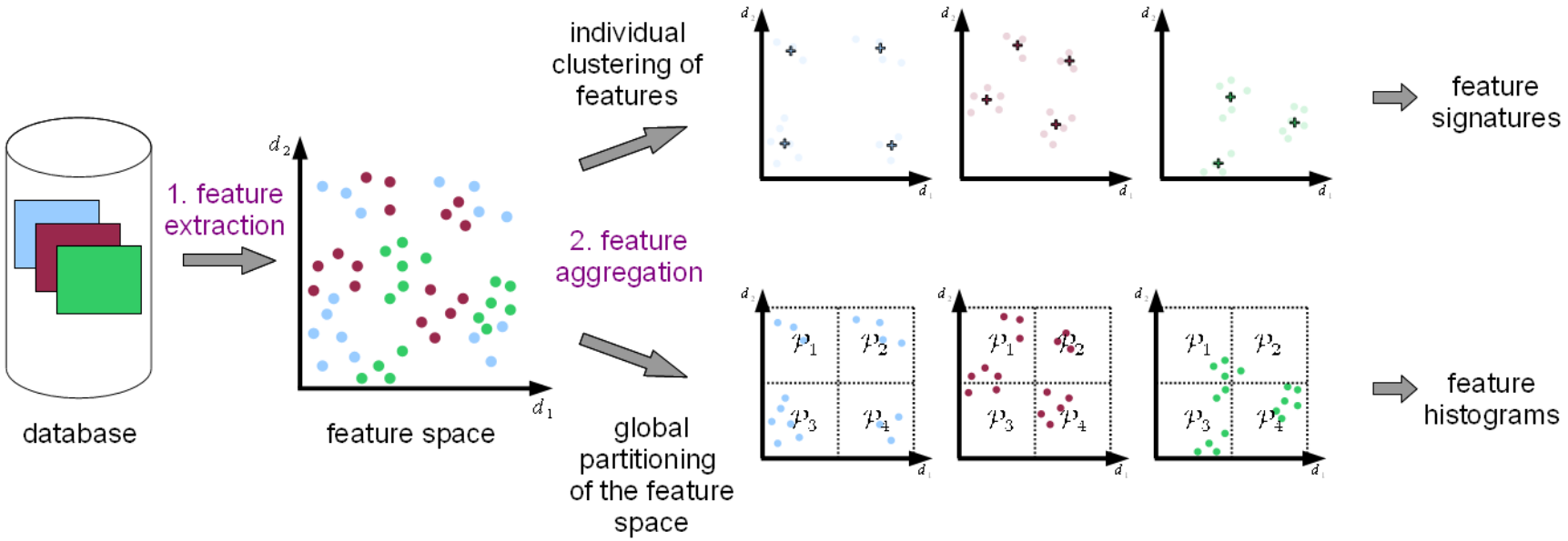
Endliche Repräsentation von Verteilungen

- Partitionierung des Featureraums
  - Zerlege Featureraum in r Partitionen (Zellen, Bins).
  - Formalisierung: $\mathcal{F} = \bigcup_{i=1}^{r} F_i$ mit $F_i \cap F_j = \emptyset$ für $i \neq j$
  - Die Partitionierung kann auf fixem Gitter beruhen oder datenbasiert z.B. auf Clustering (Bsp. K-means oder EM-Algorithmus)
  - Oft wählt man Repräsentanten f_i für Partitionen F_i, z.B. Clusterzentren

- Diskrete Repräsentation der Verteilungen
  - Partitionierung ermöglicht endliche Repräsentation,
    $$O: \{F_1, F_2, \ldots, F_r\} \rightarrow \mathbb{R}_0^+$$
  - Die Dichte der Verteilung wird aggregiert: $O(F_i) = \int_{f \in F_i} O(f)\, df$
  - Dies entspricht einer Quantisierung der Verteilung.

Featurebasierte Objektrepräsentation

Beecks C., Uysal M.S., Seidl T.: A Comparative Study of Similarity Measures for Content-Based Multimedia Retrieval. Proc. IEEE ICME 2010.



1. Feature extraction: Relevante Informationen werden aus den Objekten extrahiert und in einem Merkmalsraum dargestellt.
2. Feature aggregation: Die Merkmalsverteilungen werden in eine kompakte Merkmalsrepräsentation zusammengefasst.

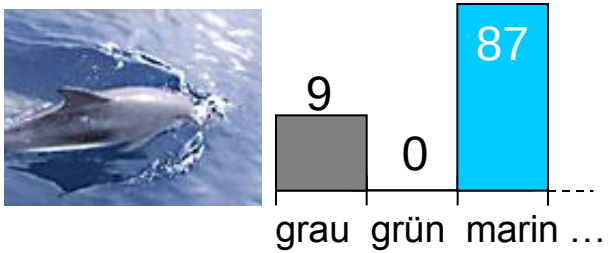
Bsp: Ähnlichkeitsmodell mit Farbhistogrammen

- Wahl der Merkmalsextraktion und -repräsentation:

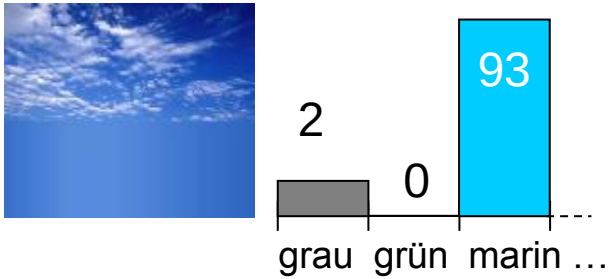
  Merkmale: z.B. Farben im RGB-Farbraum

  Repräsentation: z.B. Farbhistogramme
  - Festlegen von Farbraumzellen (hier: grau, grün, marin, …)
  - Zählen der jeweiligen Pixel (hier in Prozent)

Anfrageobjekt q



Datenbankobjekt o_1



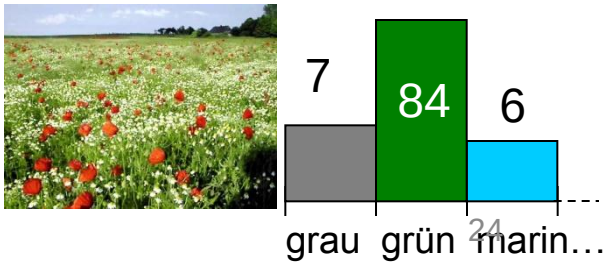
- Wahl einer Distanzfunktion:

  Euklidische Distanz $L_2(q,o) = \sqrt{\sum_{i=1}^{n}(q_i - o_i)^2}$

$$d(q,o_1) = \sqrt{(9-2)^2 + (0-0)^2 + (87-93)^2 + \ldots} \approx 9{,}21$$

$$d(q,o_2) = \sqrt{(9-7)^2 + (0-84)^2 + (87-6)^2 + \ldots} \approx 116{,}72$$

Datenbankobjekt o_2



Merkmalsrepräsentation: Signaturen

- Signaturen (auch adaptive Histogramme bzw. individuelles Binning)
  - Motivation: Fixe Partitionierung des Farbraumes für alle Bilder beschreibt Farbhäufigkeitsverteilung einzelner Bilder nur ungenau
  - Idee: Eine Partitionierung des Merkmalsraums pro Bild

- Formalisierung:
  - Sei $C^o = \{c_1^o, \ldots, c_n^o\}$ eine Partitionierung der Merkmale $f_1^o, \ldots, f_k^o \in R$ eines Objektes o mit Gewichten $\{w_1^o, \ldots, w_n^o\}$
  - Dann ist die Signatur S^o wie folgt definiert: $S^o = \{\langle c_i^o, w_i^o \rangle, i = 1, \ldots, n\}$
  - Beispiel: Centroid $c_i^o = \frac{\sum_{f \in c_i^o} f}{|c_i^o|}$ und Gewicht $w_i^o = \frac{|c_i^o|}{k}$

Merkmalsrepräsentation: Signaturen (2)

- Beispiel (5-dimensionaler Merkmalsraum: Farbe + Position):
  - Repräsentanten sind durch Mittelpunkte der farbigen Kreise dargestellt
  - Gewichte werden durch die Radien der Kreise beschrieben

- Vorteile:
  - Signaturen passen sich an individuelle Bildinhalte an
- Nachteile:
  - Vergleich mittels adaptiven Distanzfunktionen kann zu größeren Berechnungszeiten führen

Verteilungen bilden einen Vektorraum

- Die Menge $\mathbb{R}^{\mathcal{F}} = \{O: \mathcal{F} \to \mathbb{R}\}$ bildet mit + und * einen Vektorraum:
  - Addition: $\forall o_1, o_2 \in \mathbb{R}^{\mathcal{F}}: (o_1 + o_2) \in \mathbb{R}^{\mathcal{F}}, (o_1 + o_2)(f) = o_1(f) + o_2(f)$
  - Skalare Multiplikation: $\forall s \in \mathbb{R}, o \in \mathbb{R}^{\mathcal{F}}: (s * o) \in \mathbb{R}^{\mathcal{F}}, (s * o)(f) = s * o(f)$
  - Wir müssen auch negative Gewichte zulassen: $(-1 * o)(f) = -o(f)$

- Eine Basis für den Vektorraum $\left(\mathbb{R}^{\mathcal{F}}, +, *\right)$
  - Zu jedem Feature $f_0 \in \mathcal{F}$ gibt es eine charakteristische Verteilung $o_{f_0} \in \mathbb{R}^{\mathcal{F}}$ mit $o_{f_0}(f_0) = 1$ und $o_{f_0}(f) = 0$ für alle $f \in \mathcal{F}$.
  - Die charakteristischen Verteilungen zu verschiedenen Features $f_1, f_2 \in \mathcal{F}$ sind linear unabhängig.
  - Die charakteristischen Verteilungen $\left\{o_f \in \mathbb{R}^{\mathcal{F}} | f \in \mathcal{F}\right\}$ bilden eine Basis.
  - Jede (komplexe) Verteilung $o \in \mathbb{R}^{\mathcal{F}}$ lässt sich als Linearkombination aus den Basisvektoren darstellen:

$$o = w_1 * o_{f_1} + w_2 * o_{f_2} + \cdots + w_r * o_{f_r}$$

Partitionsbasierter Vektorraum

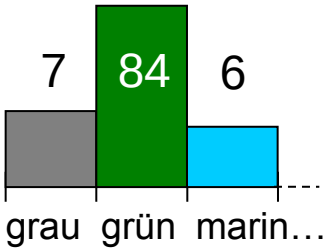
- Auch Verteilungen über Partitionen sind linear unabhängig
  - Partitionierung: $\mathcal{F} = \bigcup_{i=1}^{r} F_i$ mit $F_i \cap F_j = \emptyset$ für $i \neq j$
  - Verteilung über Partitionen: $O: \{F_1, F_2, \ldots, F_r\} \rightarrow \mathbb{R}$

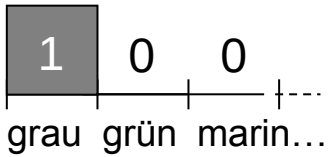
- Partitionen als Grundlage für Histogramme / Signaturen
  - Die Verteilungen zu verschiedenen Partitionen $F_1, F_2 \subset \mathcal{F}$ sind linear unabhängig.
  - Jede Verteilung über einer Partitionierung $\mathcal{F} = \bigcup_{i=1}^{r} F_i$ lässt sich als Linearkombination aus diesen Vektoren darstellen:

$$o = w_1 * o_{F_1} + w_2 * o_{F_2} + \cdots + w_r * o_{F_r} = \sum_{F_i \in Part_o} w_i o_{F_i}$$

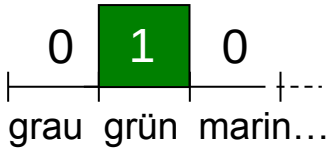
Histogramme – Verteilungssicht

- Histogramme
  - Einheitliche Partitionierung für alle Objekte („global / fixed binning“).
  - Gewichtsvektor $(w_1, w_2, \ldots, w_r)$ reicht aus, um Objekt zu repräsentieren.
  - Anfrageobjekte $(q_1, q_2, \ldots, q_r)$ folgen derselben Partitionierung.
  - Objektraum $\mathbb{R}^r$ ist Standardvektorraum mit r Dimensionen.

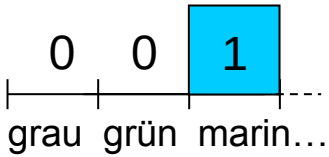


$= 7 *$ 

$+$

$84 *$ 

$+$

$6 *$ 

$+ \ldots$

$= (7, 84, 6, \ldots) \in \mathbb{R}^r$

Signaturen – Verteilungssicht

- Signaturen
  - Eigene Partitionierung für jedes Objekt („individual / adaptive binning“).
  - Repräsentation durch Partitionen&Gewichte: $\{(F_1, w_1), (F_2, w_2), \ldots, (F_{r_i}, w_{r_i})\}$

$$o = w_1 * o_{F_1} + w_2 * o_{F_2} + \cdots + w_r * o_{F_r} = \sum_{F_i \in Part_o} w_i o_{F_i}$$

- Unendliche Dimension
  - Wenn es unendliche Features $f \in \mathcal{F}$ gibt, dann hat $\mathbb{R}^{\mathcal{F}}$ unendlich viele Basisvektoren, d.h. die Dimension von $\mathbb{R}^{\mathcal{F}}$ ist unendlich.
  - In einer (endlichen) Datenbank treten nur endlich viele Basisvektoren auf.
  - Anfrageobjekte bringen jedoch eigene (neue) Partitionen mit, wir müssen also mit (potentiell) unendlich vielen Basisvektoren rechnen.

Euklidischer Abstand für Signaturen?

- Betrachte zwei Signaturen $p = \sum_{F_i \in Part_p} w_i o_{F_i}$ und $q = \sum_{F_i \in Part_q} w_i o_{F_i}$
- Betrachte den Differenzvektor

$$(p - q) = \sum_{F_i \in Part_p \cup Part_q} (p_{F_i} - q_{F_i})$$

$$= \sum_{F_i \in Part_p - Part_q} p_{F_i} - \sum_{F_i \in Part_q - Part_p} q_{F_i} + \sum_{F_i \in Part_p \cap Part_q} (p_{F_i} - q_{F_i})$$

- Bei disjunkten Partitionierungen ist euklidischer Abstand nicht sinnvoll

$$\mathrm{d}(p,q) = \sqrt{(p-q)^t (p-q)} = \sqrt{\sum_{F_i \in Part_p} {p_{F_i}}^2 + \sum_{F_i \in Part_q} {q_{F_i}}^2} = \sqrt{\|p\|^2 + \|q\|^2}$$

Zusammenfassung zu Teil 1

- Ähnlichkeitsanfragen und Objektrepräsentation

- Einführung Ähnlichkeitssuche
  - Merkmale für Multimedia-Objekte

- Anfragetypen zur Ähnlichkeitssuche
  - Bereichsanfrage, k-NN-Anfragen, Ranking

- Repräsentationen für Multimedia-Objekte
  - Verteilungen, Histogramme, SIgnaturen

Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

Part 2: Similarity and Dissimilarity Measures

Part 3: Efficient Similarity Query Processing

Thomas Seidl, Christian Beecks, RWTH Aachen

Outline

- Similarity measures
  - Kernels
- Dissimilarity measures
  - Distance functions
    - Feature histograms
      - (Weighted) Minkowski distance
      - Quadratic Form distance
    - Feature signatures
      - Matching-based approaches (e.g. Hausdorff distance)
      - Transformation-based approaches (e.g. Earth Mover's distance)
      - Correlation-based approaches (e.g. Signature Quadratic Form distance)
      - Performance evaluation
- Summary

Similarity Measures

- A similarity measure is a function $s : X \times X \rightarrow \mathbb{R}$ between two objects for which the following holds:

$$\forall x, y \in X: \; s(x,x) \geq s(x,y)$$

- Nothing is more similar than the same
- Prominent class of examples: kernels

Kernels

- A kernel is a function that corresponds to a dot product in some dot product space [HSS08]
- Given a set X, a kernel is a symmetric function $k : X \times X \to \mathbb{R}$ for which holds that:

$$\forall x, y \in X: \; k(x, y) = \langle \phi(x), \phi(y) \rangle$$

- Feature map $\phi : X \to \mathcal{H}$ and dot product space $\mathcal{H}$
- Thus, it nonlinearly generalizes on of the simplest similarity measures [S01], the canonical dot product

Positive Definite Kernels

- A kernel k is a positive definite kernel on a set X if it holds for any $n \in \mathbb{N}$, $x_1, \ldots, x_n \in X$, and $c_1, \ldots, c_n \in \mathbb{R}$:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i \cdot c_j \cdot k(x_i, x_j) \geq 0$$

If we require $\sum_{i=1}^{n} c_i = 0$, then k is a conditionally positive definite kernel

- For each positive definite kernel, one can construct a feature map and a dot product yielding the unique reproducing kernel Hilbert space [A50]

Examples of Kernels

- Gaussian kernel (pos. def. for $0 < \sigma \in \mathbb{R}$):

$$k_{\text{Gaussian}}(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

- Laplacian kernel (cond. pos. def. for $0 < \sigma \in \mathbb{R}$) [CHV99]:

$$k_{\text{Laplacian}}(x, y) = e^{-\frac{\|x-y\|}{\sigma}}$$

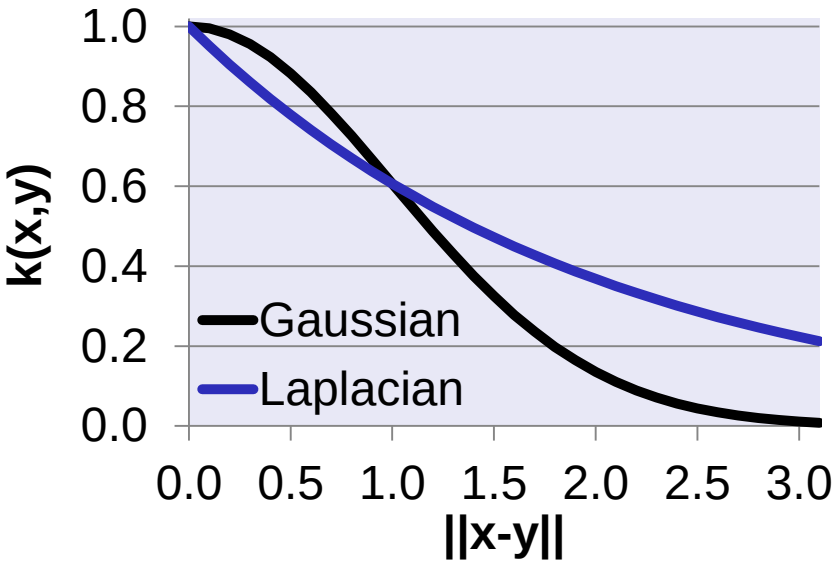
- Power kernel (cond. pos. def. for $0 < \alpha \leq 2$) [S01]:

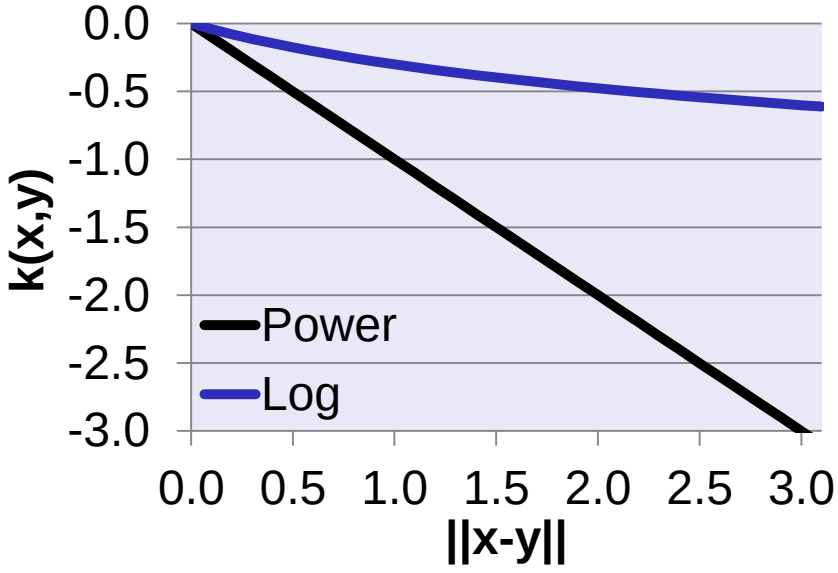
$$k_{\text{power}}(x, y) = -\|x - y\|^{\alpha}$$

- Log kernel (cond. pos. def. for $0 < \alpha \leq 2$) [BTB05]:

$$k_{\log}(x, y) = -\log(1 + \|x - y\|^{\alpha})$$

Kernel Examples cont'd





- Gaussian and Laplacian kernels show an exponentially decreasing relationship between $\|x - y\|$ and $k(x, y)$
- Both follow Shepard's universal law of generalization [S87], which states that this behavior fits multimedia features

More on Kernels

- The aforementioned kernels are usually applied to vectors, e.g., feature histograms
- There are also kernels for more complex objects:
  - kernels between sets
  - kernels between distributions
  - kernels between strings
  - kernels between trees
  - kernels between graphs
  - and many more, see for instance [G03]

Similarity versus Dissimilarity

- Kernels quantify a similarity value between two objects
- This similarity value is frequently based on a norm, see for instance the Gaussian kernel:

$$k_{\text{Gaussian}}(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

- Kernel $k_{\text{Gaussian}}(x, y)$ is based on a distance $\|x - y\|$
- Why not use the distance as dissimilarity measure?

Dissimilarity Measures

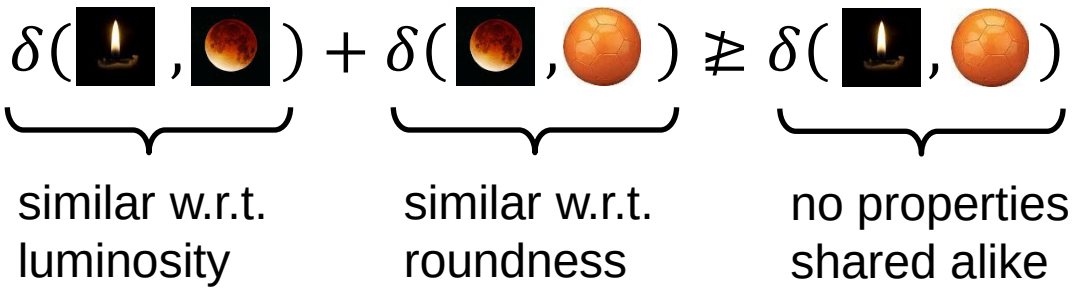
- Intuitively, a dissimilarity measure is an operator that assigns a dissimilarity value to its two arguments
- The most widespread class of dissimilarity measures is that of distance functions
- Follows the geometric distance approach:
  - Objects are positions in some (feature) space
  - Their geometric distance reflects the dissimilarity

Distance Functions

- A function $\delta: X \times X \rightarrow \mathbb{R}$ over a set X is called a distance function iff it has the following properties $\forall\, x, y \in X$:
  - reflexivity: $\delta(x, x) = 0$
  - non-negativity: $\delta(x, y) \geq 0$
  - symmetry: $\delta(x, y) = \delta(y, x)$
- A distance function δ is a metric, iff it further satisfies the following properties $\forall x, y, z \in X$:
  - identity of indiscernibles: $\delta(x, y) = 0 \Leftrightarrow x = y$
  - triangle inequality: $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$

From a Psychological Perspective…

- The distance-based approach has the advantage of a rigorous mathematical interpretation [S57]
- There is a long-lasting discussion of whether the distance properties reflect the perceived dissimilarity
- Consider the following example [J90]:



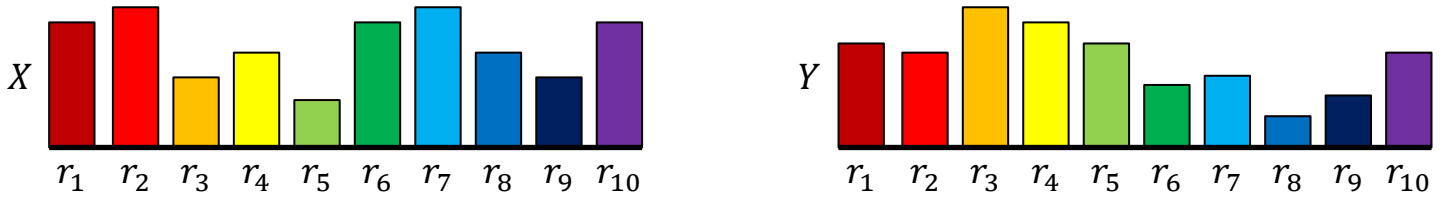
images by google image search

From a Database Perspective…

- Distance functions provide a powerful tool:
  - They allow domain experts to model their notion of dissimilarity
  - They allow database experts to design efficient query processing approaches
- Thus, database index structures can be designed by treating the distance function as a “black box”
- (we will see later how this works)

Distance Functions for Feature Histograms

- Given two feature histograms $X, Y \in \boldsymbol{H}_{\mathcal{R}}$ with, e.g., $\mathcal{R} = \{r_1, \ldots, r_{10}\}$, how can we compare them?



Distance functions for feature histograms covered here:

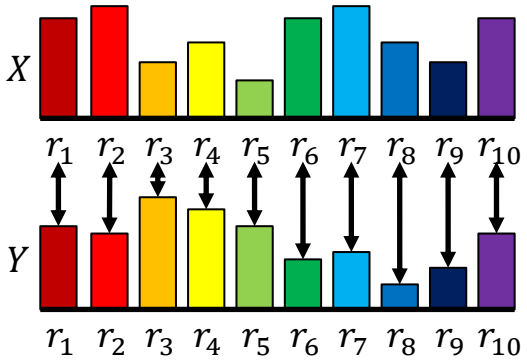
- Minkowski Distance
- Weighted Minkowski Distance
- Quadratic Form Distance

Minkowski Distance

- Bin-by-bin distance function that is defined between two feature histograms $X, Y \in \boldsymbol{H}_{\mathcal{R}}$ as:

$$\mathrm{L}_p(X,Y) = \left(\sum_{r_i \in \mathcal{R}} |X(r_i) - Y(r_i)|^p \right)^{\frac{1}{p}}$$

- Taking into account all pairwise differences:



Minkowski Distance

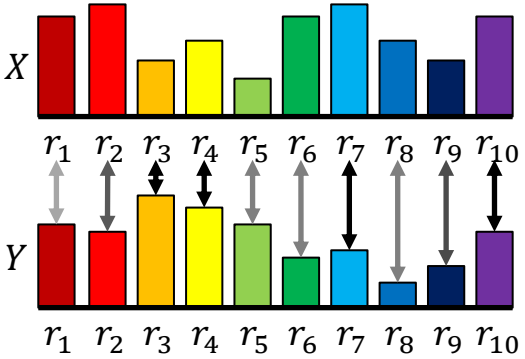
- Prominent variants:
  - $p < 1$: fractional Minkowski distance [AHK01]
  - $p = 1$: Manhattan distance
  - $p = 2$: Euclidean distance
  - $p = \infty$: Chebyshev distance $\mathrm{L}_\infty(X, Y) = \max_{r_i \in \mathcal{R}} |X(r_i) - Y(r_i)|$
- It is a metric for $1 \leq p \leq \infty$
- Although very prominent and efficient ($O(|\mathcal{R}|)$), it is inflexible w.r.t. adaptability
- Solution: weighting of representatives by a weighting function w: $\mathcal{R} \rightarrow \mathbb{R}^+$

Weighted Minkowski Distance

- Bin-by-bin distance function that is defined between two feature histograms $X, Y \in \boldsymbol{H}_{\mathcal{R}}$ as:

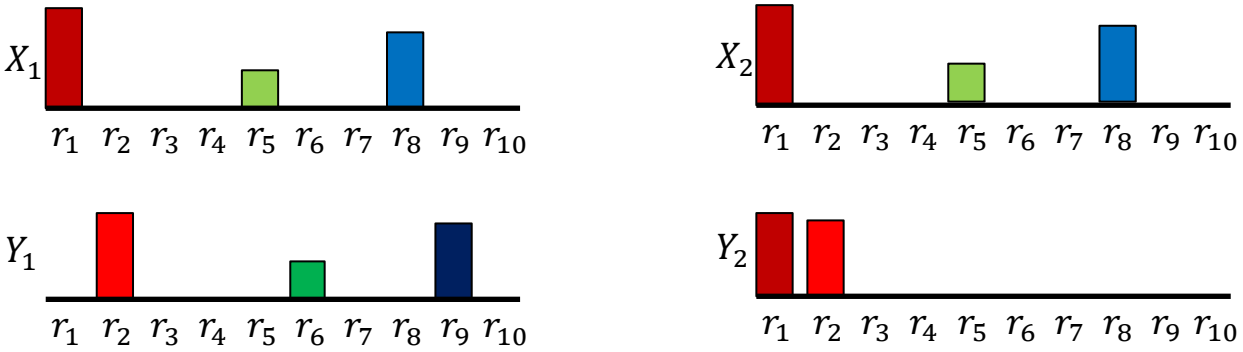
$$\mathrm{L}_{p,w}(X,Y) = \left(\sum_{r_i \in \mathcal{R}} w(r_i) \cdot |X(r_i) - Y(r_i)|^p \right)^{\frac{1}{p}}$$

- Weighting function $w: \mathcal{R} \to \mathbb{R}$ models the influence of each representative $r_i \in \mathcal{R}$:



Bin-by-bin Distances

- Problem: distance value is defined dimension-wise
- Neighboring dimensions are neglected:



- $\mathrm{L}_p(X_1, Y_1) > \mathrm{L}_p(X_2, Y_2)$
- Solution: Cross-bin distances

Cross-bin Distances

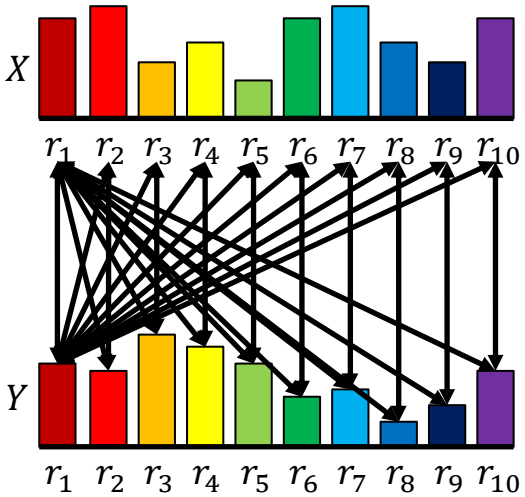
- More flexible than bin-by-bin distances
- Basic ideas:
  - Replace weighting of single representatives by a weighting of pairs of representatives
  - Model the influence not only for each single representative, but also across different representatives
  - This is done by a similarity function $s: \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{R}$

Quadratic Form Distance [I89,NBE+93,…]

- Cross-bin distance function that is defined between two feature histograms $X, Y \in \boldsymbol{H}_{\mathcal{R}}$ as:

$$\mathrm{QFD}_s(X,Y) = \left(\sum_{r_i, r_j \in \mathcal{R}} \big(X(r_i) - Y(r_i)\big) \cdot \Big(X\big(r_j\big) - Y\big(r_j\big)\Big) \cdot s(r_i, r_j) \right)^{\frac{1}{2}}$$

- Ability to model all pairwise similarities:

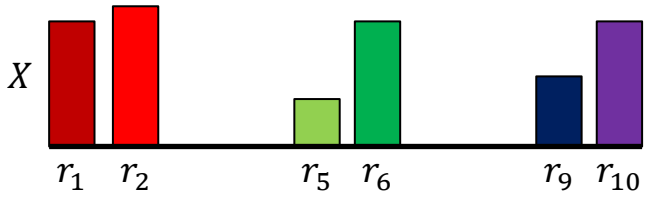


Résumé: Feature Histogram Distances

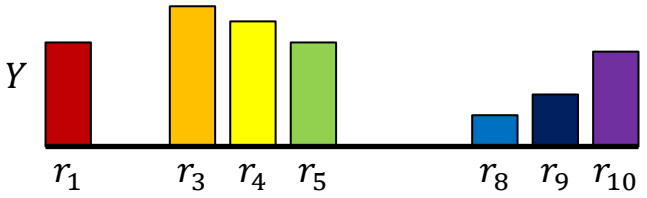
- All these distance functions are defined for feature histograms sharing the same representatives
- Weighted Minkowski distances are limited w.r.t. adaptability but show linear computation time complexity
- Quadratic Form distances are very adaptable but show quadratic computation time complexity
- Other distance functions (e.g., [RPT+01,ZL03,HRS+08]):
  - geometric measures such as cosine distance
  - information theoretic measures such as Kullback-Leibler [KL51]
  - statistic measures such as χ^2 –statistics [PHB97]

Distance Functions for Feature Signatures

- Given two feature signatures $X, Y \in \boldsymbol{S}$, how can we compare them?



$\mathcal{R}_X = \{r_1, r_2, r_5, r_6, r_9, r_{10}\}$



$\mathcal{R}_Y = \{r_1, r_3, r_4, r_5, r_8, r_9, r_{10}\}$

- Observations:
  - representatives $\mathcal{R}_X$ and $\mathcal{R}_Y$ may differ
  - necessity to compare representatives with each other
  - good idea to take into account the weights $X(r_i)$ and $Y(r_j)$

Three Competing Approaches…

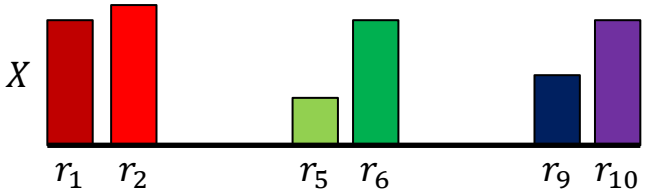
- Matching-based approach
  - Define a distance function by identifying and tieing together coincident similar parts and evaluating them with a cost function
- Transformation-based approach
  - Define a distance function by measuring the costs of transforming one feature signature into another one
- Correlation-based approach
  - Define a distance function by investigating the correlation of the representatives of the feature signatures

Matching-based Approaches

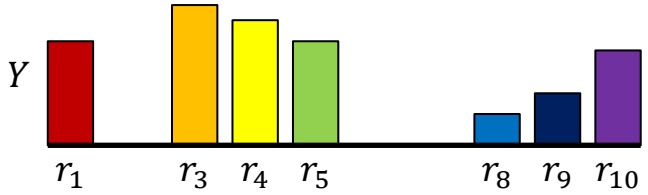
- A matching $m_{X \leftrightarrow Y} \subseteq \mathcal{R}_X \times \mathcal{R}_Y$ defines similar parts of the feature signatures $X, Y \in \boldsymbol{S}$
- If a matching $m_{X \leftrightarrow Y}$ satisfies left totality and right uniqueness, we refer to it as $m_{X \to Y}$ and describe it by a matching function $\pi_{X \to Y}: \mathcal{R}_X \to \mathcal{R}_Y$
- In this case, $m_{X \to Y}$ is completely defined by the graph of its matching function $\pi_{X \to Y}$, i.e. $m_{X \to Y} = \{(r_i, \pi_{X \to Y}(r_i)) \mid r_i \in \mathcal{R}_X\}$
- Finally, a cost function $c: 2^{\mathcal{R}_X \times \mathcal{R}_Y} \to \mathbb{R}$ is used to assess the quality of a matching

Example: Hausdorff Distance [H14,HKR93]

- Defined as the maximum nearest neighbor distance between representatives of the feature signatures:



$\mathcal{R}_X = \{r_1, r_2, r_5, r_6, r_9, r_{10}\}$



$\mathcal{R}_Y = \{r_1, r_3, r_4, r_5, r_8, r_9, r_{10}\}$

- Determine matchings:

$m_{X \to Y} = \{(r_1, r_1), (r_2, r_1), (r_5, r_5), (r_6, r_5), (r_9, r_9), (r_{10}, r_{10})\}$

$m_{Y \to X} = \{(r_1, r_1), (r_3, r_2), (r_4, r_5), (r_5, r_5), (r_8, r_9), (r_9, r_9), (r_{10}, r_{10})\}$

- Evaluate matchings by the maximum distance

Definition of the Hausdorff Distance

- Defined as the maximum nearest neighbor distance between two feature signatures $X, Y \in \boldsymbol{S}$ as :

$$\mathrm{HD}(X,Y) = \max\{c(m_{X\to Y}), c(m_{Y\to X})\}$$

- Matching $m_{X\to Y}$ gathers all nearest neighbors by the matching function $\pi_{X\to Y}(r_i) = \arg\min\{\delta(r_i, r_j) \mid r_j \in \mathcal{R}_Y\}$
- Cost function $c(m_{X\to Y}) = \max\{\delta(r_i, r_j) \mid (r_i, r_j) \in m_{X\to Y}\}$ takes the maximum of the nearest neighbor distances

Properties of the Hausdorff Distance

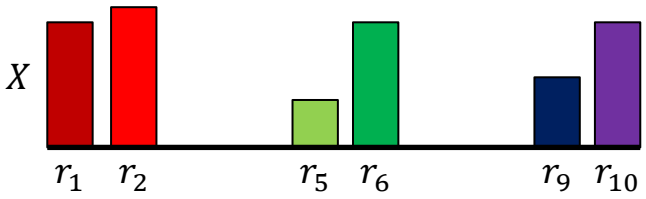
- Hausdorff distance neglects the weights $X(r_i)$ and $Y(r_j)$ of the feature signature
- It is not a metric on feature signatures
- It shows a quadratic computation time complexity
- Many variants exist:
  - E.g. the perceptually modified Hausdorff distance [PLL08], which replaces the matching function and cost function by:
    - $\pi_{X \to Y}(r_i) = \arg\min_{r_j \in \mathcal{R}_Y} \left\{ \frac{\delta(r_i, r_j)}{\min\{X(r_i), Y(r_j)\}} \right\}$
    - $c(m_{X \to Y}) = \frac{\sum_{(r_i, r_j) \in m_{X \to Y}} X(r_i) \cdot \frac{\delta(r_i, r_j)}{\min\{X(r_i), Y(r_j)\}}}{\sum_{(r_i, r_j) \in m_{X \to Y}} X(r_i)}$

Transformation-based Approaches

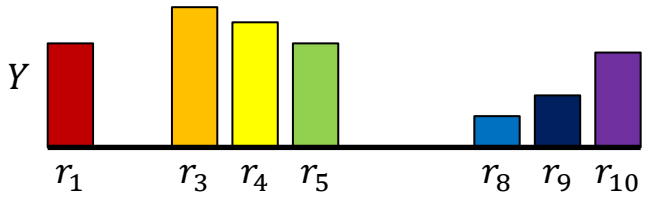
- Conceptual idea consists in transforming one feature signature into another one
- Treating transformation costs as distance
- Examples:
  - Levenshtein/Edit distance on discrete structures [L66]
    - Edit operations: insertion, deletion, substitution
    - Distance is defined as the minimum number of edit operations
  - Dynamic Time Warping distance on times series [I75,SC78]
    - Warping operation: replication
    - Distance is defined as the minimum number of replications
  - Earth Mover's Distance on feature signatures [RTG00]

Earth Mover's Distance (EMD)

- Intuitively, it solves a transportation problem:
  - Consider $X(r_i)$ as mound of earth
  - Consider $Y(r_j)$ as holes in the ground
  - Let f_{r_i,r_j} denote the amount of earth moved from r_i to r_j (flow)
  - Let $\delta(r_i, r_j)$ denote the cost between r_i and r_j



$\mathcal{R}_X = \{r_1, r_2, r_5, r_6, r_9, r_{10}\}$



$\mathcal{R}_Y = \{r_1, r_3, r_4, r_5, r_8, r_9, r_{10}\}$

- How to minimize the sum of moving earth from X to Y?

Definition of the EMD

- Defined as the minimum cost flow $F \in \mathbb{R}^{|\mathcal{R}_X| \times |\mathcal{R}_Y|}$ over all possible flows $f_{r_i,r_j} = F_{ij} \in \mathbb{R}$ as:

$$\mathrm{EMD}(X,Y) = \min_F \left\{ \frac{\sum_{r_i \in \mathcal{R}_X} \sum_{r_j \in \mathcal{R}_Y} f_{r_i,r_j} \cdot \delta(r_i, r_j)}{\min\left\{ \sum_{r_i \in \mathcal{R}_X} X(r_i), \sum_{r_j \in \mathcal{R}_Y} Y(r_j) \right\}} \right\}$$

- Subject to the following constraints:
  - $\forall r_i \in \mathcal{R}_X, r_j \in \mathcal{R}_Y: f_{r_i,r_j} \geq 0$ (non-negative flows)
  - $\forall r_i \in \mathcal{R}_X: \sum_{r_j \in \mathcal{R}_Y} f_{r_i,r_j} \leq X(r_i)$ (limited source)
  - $\forall r_j \in \mathcal{R}_Y: \sum_{r_i \in \mathcal{R}_X} f_{r_i,r_j} \leq Y(r_j)$ (limited target)
  - $\sum_{r_i \in \mathcal{R}_X} \sum_{r_j \in \mathcal{R}_Y} f_{r_i,r_j} = \min\left\{ \sum_{r_i \in \mathcal{R}_X} X(r_i), \sum_{r_j \in \mathcal{R}_Y} Y(r_j) \right\}$ (min. flow)

Properties of the EMD

- Finding an optimal solution can be computed based on a specific variant of the simplex algorithm [HL90]
- Exponential computation time complexity
- Average empirical computation time complexity between $O(|\mathcal{R}_X|^3)$ and $O(|\mathcal{R}_X|^4)$ [SJ08], for $|\mathcal{R}_X| \geq |\mathcal{R}_Y|$
- It is a metric iff the feature signatures are normalized, i.e. $\sum_{r_i \in \mathcal{R}_X} X(r_i) = \sum_{r_j \in \mathcal{R}_Y} Y(r_j)$, and δ is a metric

Correlation-based Approaches

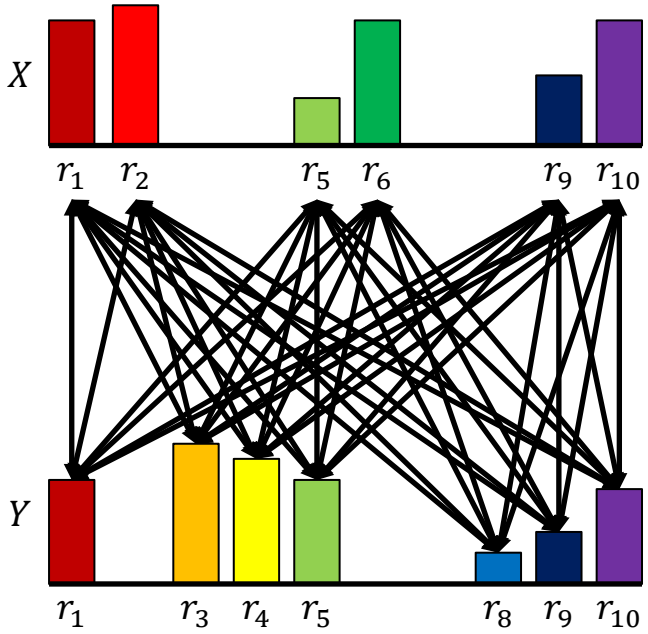
- Idea: compare all representatives of the feature signatures with each other
- Similarity correlation based on a similarity function s between two feature signatures $X, Y \in \boldsymbol{S}$:

$$\langle X, Y \rangle_s = \sum_{r_i \in \mathcal{R}_X} \sum_{r_j \in \mathcal{R}_Y} X(r_i) \cdot Y(r_j) \cdot s(r_i, r_j)$$

- High correlation is expected when the representatives are close to each other
- Example: weighted correlation distance [LL04]

Signature Quadratic Form Distance (SQFD)

- The Signature Quadratic Form distance [BUS10] is a generalization of the Quadratic Form distance
- It takes into account all pairwise similarities:



Definition of the SQFD

- Defined between two feature signatures $X, Y \in \boldsymbol{S}$ as:

$$\mathrm{SQFD}(X, Y) = \sqrt{\langle X - Y, X - Y \rangle_S}$$

- Explicit use of "difference signature" $(X - Y) \in \boldsymbol{S}$
- Efficient computation:

$$\langle X - Y, X - Y \rangle_S = \underbrace{\langle X, X \rangle_S}_{\text{Similarity correlation of } X} - \underbrace{\langle X, Y \rangle_S - \langle Y, X \rangle_S}_{\text{Similarity correlation between } X \text{ and } Y} + \underbrace{\langle Y, Y \rangle_S}_{\text{Similarity correlation of } Y}$$

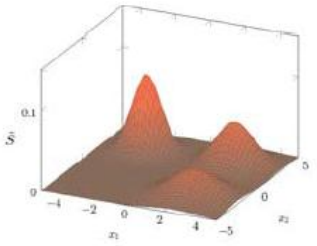
Properties of the SQFD

- It shows quadratic computation time complexity
- It is necessary to adapt the similarity function to specific domains
- SQFD is a metric on $\boldsymbol{S}$ iff the similarity function is a strictly positive definite function
- SQFD is a metric on $\boldsymbol{S}_\lambda$ iff the similarity function is a strictly conditionally positive definite function
- Luckily, the Gaussian kernel yields a metric SQFD

Properties of the SQFD

- Widespread applicability
  - For instance to Gaussian mixture models [BIK+11]

(a) Picturesque sunset



(b) Continuous distribution

- Promising indexability
  - Steerable by adapting the similarity function [BLS+11]
  - Beyond metric indexing: Ptolemaic indexing [HSL+12]
- Impressive parallelization
  - For instance on GPUs [KSL+12]

Performance Evaluation

- Databases: Corel Wang [WLW01], Coil100 [NNM96], MIR Flickr [HL08]
- 7-dimensional Features: color, position, and texture

- Task: image retrieval
- Evaluation measure: mean average precision [MRS08]

Performance Evaluation (Wang database)

- The following results are taken from [BUS10a]

Table 2. Mean average precision values for the *Wang* database.

features	SQFD	HD	PMHD	WCD	EMD	α
color	**0.586**	0.317	0.439	0.567	0.580	2.9
color + texture	**0.620**	0.345	0.476	0.604	0.599	1.5
color + position	**0.568**	0.391	0.461	0.536	0.563	1.2
color + pos. + text.	**0.613**	0.308	0.476	0.591	0.598	0.9
mean:	**0.597**	0.340	0.463	0.575	0.585	

- Mean average precision order:
  - HD < PMHD < WCD < EMD < SQFD
- SQFD performs best

Performance Evaluation (Coil100 database)

Table 3. Mean average precision values for the *Coil100* database.

features	SQFD	HD	PMHD	WCD	EMD	α
color	0.811	0.731	**0.828**	0.772	0.809	2.9
color + texture	**0.770**	0.477	0.696	0.728	0.750	1.5
color + position	**0.802**	0.498	0.664	0.743	0.706	0.7
color + pos. + text.	**0.776**	0.425	0.606	0.726	0.710	0.6
mean:	**0.790**	0.533	0.699	0.742	0.744	

- Mean average precision order:
  - HD < PMHD < WCD < EMD < SQFD
- SQFD shows best average performance
- PMHD shows highest peak performance

Performance Evaluation (MIR Flickr database)

Table 4. Mean average precision values for the *MIR Flickr* database.

features	SQFD	HD	PMHD	WCD	EMD	α
color	**0.322**	0.318	**0.322**	0.321	**0.322**	0.7
color + texture	**0.343**	0.317	0.331	0.338	0.337	1.0
color + position	**0.321**	0.316	0.319	0.317	0.314	0.2
color + pos. + text.	**0.343**	0.307	0.322	0.335	0.333	0.6
mean:	**0.332**	0.315	0.324	0.328	0.327	

- Mean average precision order:
  - HD < PMHD < EMD < WCD < SQFD
- SQFD performs best

Performance Evaluation (UCID database)

- We also evaluated more complex descriptors on the UCID database [SS04]:

	HD	PMHD	EMD	WCD	$SQFD_M$	$SQFD_L^{DB}$	$SQFD_L^{Q}$	$SQFD_L^{\hat{d}}$	$SQFD_G^{DB}$	$SQFD_G^{Q}$	$SQFD_G^{\hat{d}}$
colormomentinvar.	0.451	**0.476**	0.462	0.460	0.460	0.460	0.460	0.404	0.464	0.465	0.421
colormoments	0.475	**0.554**	0.520	0.518	0.516	0.516	0.517	0.502	0.498	0.500	0.493
huehistogram	0.522	0.658	0.631	0.628	0.632	0.657	0.659	0.643	0.640	**0.659**	0.649
nrghistogram	0.551	**0.670**	0.620	0.609	0.610	0.624	0.632	0.619	0.604	0.612	0.610
opponenthistogram	0.497	**0.659**	0.627	0.618	0.619	0.641	0.647	0.626	0.617	0.632	0.622
rgbhistogram	0.472	0.577	0.548	0.571	0.582	0.599	**0.601**	0.571	0.572	0.589	0.575
transformedcolorhist.	0.437	0.509	0.533	**0.571**	0.570	0.535	0.523	0.441	0.554	0.554	0.528
csift	0.497	0.632	0.628	0.631	0.630	0.630	0.630	0.600	0.632	**0.644**	0.629
hsvsift	0.468	0.602	0.599	0.605	0.601	0.601	0.601	0.582	0.605	**0.618**	0.613
huesift	0.549	0.664	0.659	0.657	0.656	0.656	0.656	0.641	0.656	**0.670**	0.657
opponentsift	0.470	0.573	0.563	0.574	0.575	0.575	0.575	0.567	0.589	**0.605**	0.602
rgbsift	0.443	0.525	0.519	0.540	0.543	0.543	0.543	0.534	0.553	**0.561**	0.560
rgsift	0.506	0.630	0.621	0.617	0.613	0.613	0.613	0.586	0.623	**0.639**	0.624
sift	0.435	0.499	0.496	0.532	0.531	0.531	0.531	0.524	0.547	**0.553**	0.548
transformedcolorsift	0.443	0.523	0.515	0.541	0.543	0.543	0.543	0.533	0.553	**0.561**	0.560
average:	0.481	0.583	0.570	0.578	0.579	0.582	0.582	0.558	0.580	**0.591**	0.579

- SQFD performs best on average

Performance Evaluation (UCID database)

- We also evaluated more complex descriptors on the UCID database [SS04]:

	HD	PMHD	EMD	WCD	$SQFD_M$	$SQFD_L^{DB}$	$SQFD_L^{Q}$	$SQFD_L^{\hat{d}}$	$SQFD_G^{DB}$	$SQFD_G^{Q}$	$SQFD_G^{\hat{d}}$
colormomentinvar.	0.451	**0.476**	0.462	0.460	0.460	0.460	0.460	0.404	0.464	0.465	0.421
colormoments	0.475	**0.554**	0.520	0.518	0.516	0.516	0.517	0.502	0.498	0.500	0.493
huehistogram	0.522	0.658	0.631	0.628	0.632	0.657	0.659	0.643	0.640	**0.659**	0.649
nrghistogram	0.551	**0.670**	0.620	0.609	0.610	0.624	0.632	0.619	0.604	0.612	0.610
opponenthistogram	0.497	**0.659**	0.627	0.618	0.619	0.641	0.647	0.626	0.617	0.632	0.622
rgbhistogram	0.472	0.577	0.548	0.571	0.582	0.599	**0.601**	0.571	0.572	0.589	0.575
transformedcolorhist.	0.437	0.509	0.533	**0.571**	0.570	0.535	0.523	0.441	0.554	0.554	0.528
csift	0.497	0.632	0.628	0.631	0.630	0.630	0.630	0.600	0.632	**0.644**	0.629
hsvsift	0.468	0.602	0.599	0.605	0.601	0.601	0.601	0.582	0.605	**0.618**	0.613
huesift	0.549	0.664	0.659	0.657	0.656	0.656	0.656	0.641	0.656	**0.670**	0.657
opponentsift	0.470	0.573	0.563	0.574	0.575	0.575	0.575	0.567	0.589	**0.605**	0.602
rgbsift	0.443	0.525	0.519	0.540	0.543	0.543	0.543	0.534	0.553	**0.561**	0.560
rgsift	0.506	0.630	0.621	0.617	0.613	0.613	0.613	0.586	0.623	**0.639**	0.624
sift	0.435	0.499	0.496	0.532	0.531	0.531	0.531	0.524	0.547	**0.553**	0.548
transformedcolorsift	0.443	0.523	0.515	0.541	0.543	0.543	0.543	0.533	0.553	**0.561**	0.560
average:	0.481	0.583	0.570	0.578	0.579	0.582	0.582	0.558	0.580	**0.591**	0.579

- SQFD performs best on average

Summary

- Similarity measures
  - Kernels
- Dissimilarity measures
  - Distance functions
    - Feature histograms
      - (Weighted) Minkowski distance
      - Quadratic Form distance
    - Feature signatures
      - Matching-based approaches (e.g. Hausdorff distance)
      - Transformation-based approaches (e.g. Earth Mover's distance)
      - Correlation-based approaches (e.g. Signature Quadratic Form distance)

References

- **[AHK01]** Aggarwal, C. C.; Hinneburg, A. & Keim, D. A. On the Surprising Behavior of Distance Metrics in High Dimensional Spaces
  Proceedings of the International Conference on Database Theory, **2001**, 420-434
- **[A50]** Aronszajn, N. Theory of reproducing kernels
  Trans. Amer. Math. Soc., **1950**, *68*, 337-404
- **[BIK+11]** Beecks, C.; Ivanescu, A. M.; Kirchhoff, S. & Seidl, T. Modeling Image Similarity by Gaussian Mixture Models and the Signature Quadratic Form Distance
  Proceedings of the IEEE International Conference on Computer Vision, **2011**, 1754-1761
- **[BLS+11]** Beecks, C.; Lokoc, J.; Seidl, T. & Skopal, T. Indexing the signature quadratic form distance for efficient content-based multimedia retrieval
  Proceedings of the ACM International Conference on Multimedia Information Retrieval, **2011**
- **[BUS10]** Beecks, C.; Uysal, M. S. & Seidl, T. Signature Quadratic Form Distance
  Proceedings of the ACM International Conference on Image and Video Retrieval, **2010**, 438-445
- **[BUS10a]** Beecks, C.; Uysal, M. S. & Seidl, T. A comparative study of similarity measures for content-based multimedia retrieval
  Proceedings of the IEEE International Conference on Multimedia and Expo, **2010**, 1552-1557
- **[BTB05]** Boughorbel, S.; Tarel, J.-P. & Boujemaa, N. Conditionally Positive Definite Kernels for SVM Based Image Recognition
  Proceedings of the IEEE International Conference on Multimedia and Expo, **2005**, 113-116
- **[CHV99]** Chapelle, O.; Haffner, P. & Vapnik, V. Support vector machines for histogram-based image classification
  IEEE Transactions on Neural Networks, **1999**, *10*, 1055-1064
- **[G03]** Gärtner, T. A survey of kernels for structured data
  SIGKDD Explorations, **2003**, *5*, 49-58
- **[H14]** Hausdorff, F. Grundzüge der Mengenlehre
  Von Veit, **1914**

References

- **[HSL+12]** Hetland, M. L.; Skopal, T.; Lokoc, J. & Beecks, C. Ptolemaic Access Methods: Challenging the Reign of the Metric Space Model
  Information Systems, Elsevier, **2012**
- **[HL90]** Hillier, F. & Lieberman, G. Introduction to Linear Programming
  McGraw-Hill, **1990**
- **[HL08]** Huiskes, M. J. & Lew, M. S. The MIR flickr retrieval evaluation
  Proceedings of the ACM International Conference on Multimedia Information Retrieval, **2008**, 39-43
- **[HKR93]** Huttenlocher, D. P.; Klanderman, G. A. & Rucklidge, W. Comparing Images Using the Hausdorff Distance
  IEEE Transactions on Pattern Analysis and Machine Intelligence, **1993***, 15*, 850-863
- **[HRS+08]** Hu, R.; Rüger, S. M.; Song, D.; Liu, H. & Huang, Z. Dissimilarity measures for content-based image retrieval
  Proceedings of the IEEE International Conference on Multimedia and Expo, **2008**, 1365-1368
- **[HSS08]** Hofmann, T.; Schölkopf, B. & Smola, A. Kernel methods in machine learning
  The annals of statistics, JSTOR, **2008**, 1171-1220
- **[I89]** Ioka, M. A Method of Defining the Similarity of Images on the Basis of Color Information
  IBM Research, Tokyo Research Laboratory, **1989**
- **[I75]** Itakura, F. Minimum prediction residual principle applied to speech recognition
  IEEE Transactions on Acoustics, Speech and Signal Processing, IEEE, **1975***, 23*, 67-72
- **[J90]** James, W. The principles of psychology.
  Henry Holt and Company, **1890**
- **[KSL+12]** Krulis, M.; Skopal, T.; Lokoc, J. & Beecks, C. Combining CPU and GPU architectures for fast similarity search
  Distributed and Parallel Databases, **2012***, 30*, 179-207
- **[KL51]** Kullback, S. & Leibler, R. A. On Information and Sufficiency
  The Annals of Mathematical Statistics, Institute of Mathematical Statistics, **1951***, 22*, 79-86

References

- **[LL04]** Leow, W. K. & Li, R. The analysis and applications of adaptive-binning color histograms
  Computer Vision and Image Understanding, **2004**, *94*, 67-91
- **[L66]** Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals
  Soviet Physics Doklady, **1966**, *10*, 707-710
- **[MRS08]** Manning, C.; Raghavan, P. & Schutze, H. Introduction to information retrieval
  Cambridge University Press Cambridge, **2008**, *1*
- **[NNM96]** Nene, S.; Nayar, S. K. & Murase, H. Columbia Object Image Library (COIL-100)
  Department of Computer Science, Columbia University, **1996**
- **[NBE+93]** Niblack, W.; Barber, R.; Equitz, W.; Flickner, M.; Glasman, E.; Petkovic, D.; Yanker, P.; Faloutsos, C. & Taubin, G. QBIC project: querying images by content, using color, texture, and shape
  Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, **1993**, *1908*, 173-187
- **[PLL08]** Park, B. G.; Lee, K. M. & Lee, S. U. Color-Based Image Retrieval Using Perceptually Modified Hausdorff Distance
  EURASIP Journal of Image and Video Processing, **2008**, *2008*
- **[PHB97]** Puzicha, J.; Hofmann, T. & Buhmann, J. M. Non-parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval
  Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, **1997**, 267-272
- **[RTG00]** Rubner, Y.; Tomasi, C. & Guibas, L. J. The Earth Mover's Distance as a Metric for Image Retrieval
  International Journal of Computer Vision, **2000**, *40*, 99-121
- **[RPT+01]** Rubner, Y.; Puzicha, J.; Tomasi, C. & Buhmann, J. M. Empirical Evaluation of Dissimilarity Measures for Color and Texture
  Computer Vision and Image Understanding, **2001**, *84*, 25-43
- **[SC78]** Sakoe, H. & Chiba, S. Dynamic programming algorithm optimization for spoken word recognition
  IEEE Transactions on Acoustics, Speech and Signal Processing, IEEE, **1978**, *26*, 43-49
- **[SS04]** Schaefer, G. & Stich, M. UCID: an uncompressed color image database
  Proceedings of the SPIE, Storage and Retrieval Methods and Applications for Multimedia, **2004**, 472-480
- **[S01]** Schölkopf, B. The kernel trick for distances
  Advances in neural information processing systems, MIT; 1998, **2001**, 301-307

References

- **[S57]** Shepard, R. N. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space *Psychometrika, Springer New York,* **1957***, 22*, 325-345
- **[S87]** Shepard, R. Toward a universal law of generalization for psychological science *Science, American Association for the Advancement of Science,* **1987***, 237*, 1317-1323
- **[SJ08]** Shirdhonkar, S. & Jacobs, D. W. Approximate earth mover's distance in linear time *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition,* **2008**
- **[WLW01]** Wang, J. Z.; Li, J. & Wiederhold, G. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **2001***, 23*, 947-963
- **[ZL03]** Zhang, D. & Lu, G. Evaluation of similarity measurement for image retrieval *Proceedings of the IEEE International Conference on Neural Networks and Signal Processing,* **2003***, 2*, 928 - 931

Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

Part 2: Similarity and Dissimilarity Measures

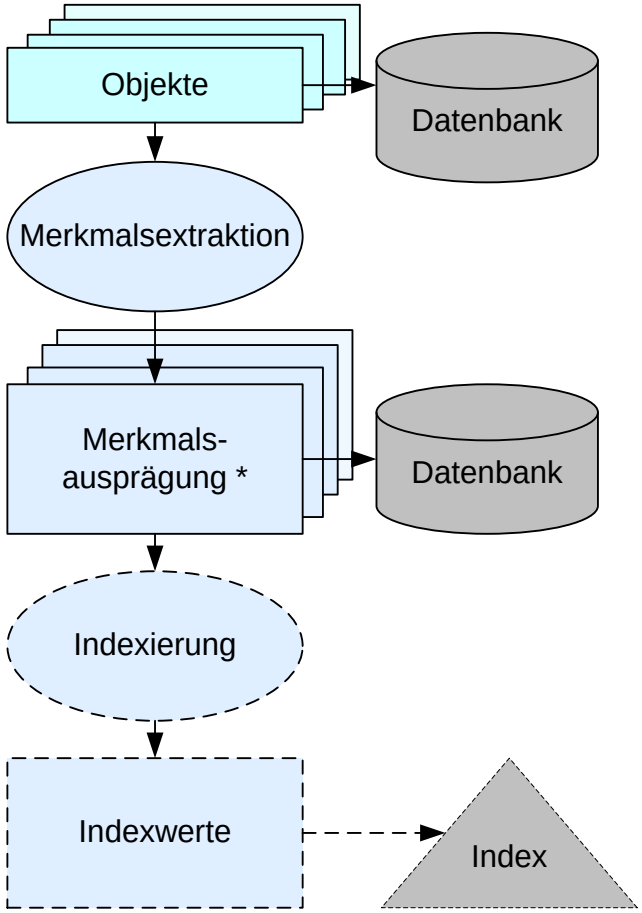
Part 3: Efficient Similarity Query Processing

Thomas Seidl, Christian Beecks, RWTH Aachen

Outline

- Given a distance-based similarity model
  - Feature representation, e.g. feature signature
  - Distance function, e.g. SQFD
- How can we process distance-based similarity queries efficiently?
- How to avoid time-intensive sequential scan?
  - Multistep range query architecture
  - Multistep kNN query architecture
  - Metric filtering/indexing
  - Ptolemaic metric filtering/indexing

Aufbereitung von Daten für das Information Retrieval

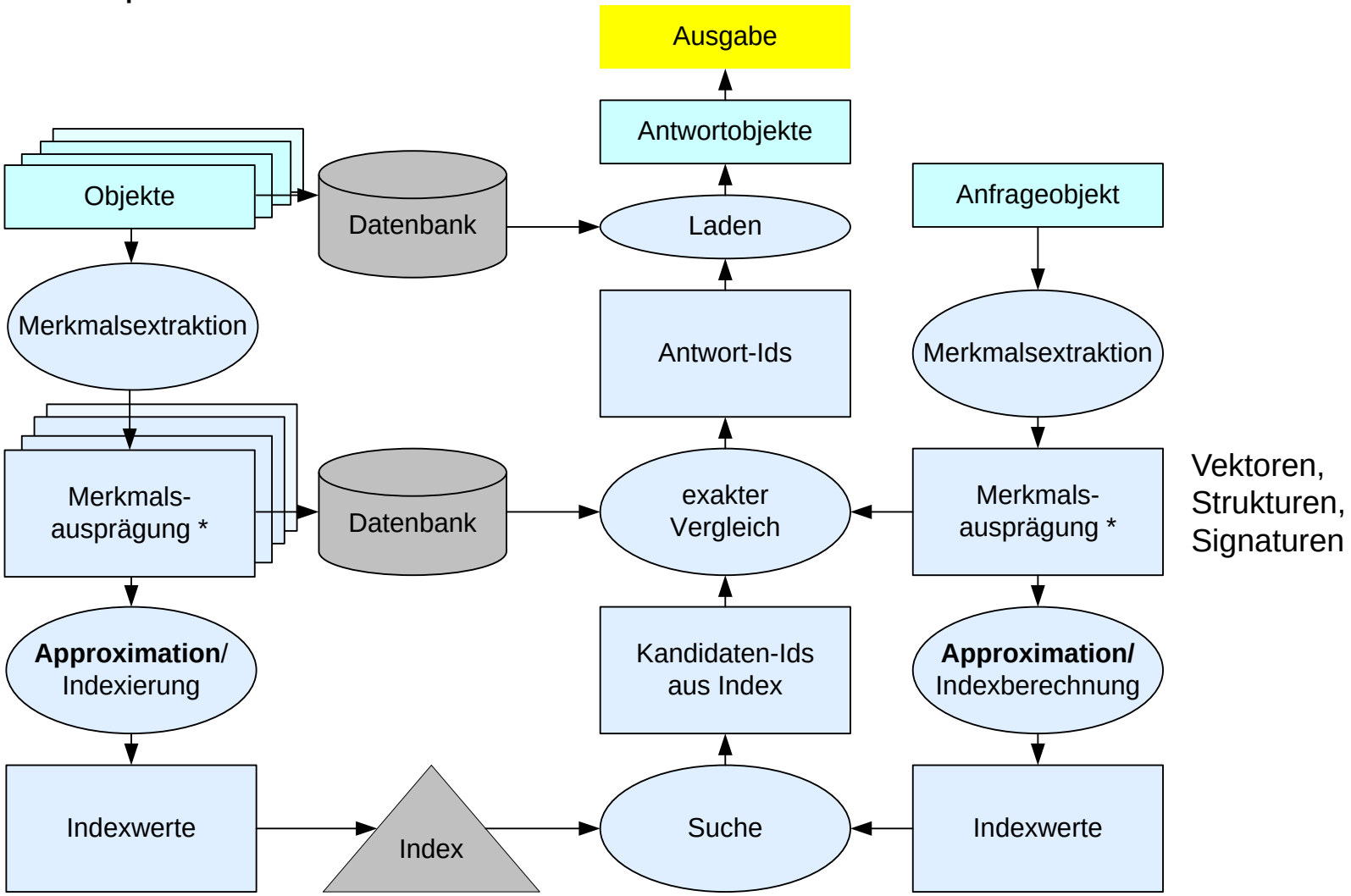


- Merkmalsextraktion: Bestimme relevante Merkmale der Objekte
- *Merkmalsausprägungen* sind z.B. Vektoren, Signaturen, Strukturen
- *Indexierung* dient der Beschleunigung der Anfragebearbeitung

Prozesse des Information Retrieval

Aufbauprozess

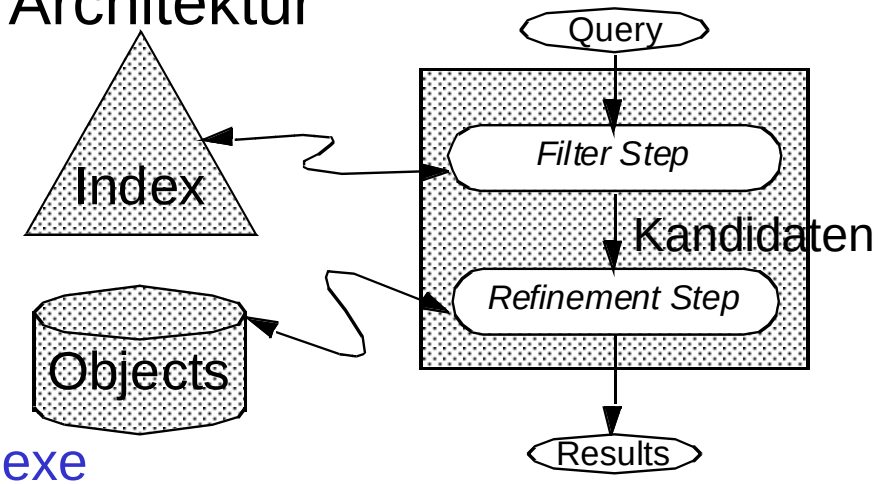
Anfrageprozess



Mehrstufige Anfragebearbeitung

- Approximationen können nicht gleichzeitig vollständige und korrekte Antworten liefern (sonst wären das keine Approximationen)
  - Korrektheit: keine falschen Objekte liefern
  - Vollständigkeit: kein richtiges Objekt verwerfen

  → Benutze mehrstufige Architektur



- Filterschritt(e):
  - Ziel: schnell (kleine) Menge von Kandidaten generieren
  - verwenden Approximationen
  - verwenden typischerweise Indexe

- Verfeinerungsschritt(e):
  - exakte Überprüfung der Kandidaten

Multistep Range Query Architecture

- Query: $range\,(q,\varepsilon) = \{o \in DB \mid \delta\,(q,o) \leq \varepsilon\}$
- Use a filter distance LB for which holds that

$$\forall p, o: \mathrm{LB}(p,o) \leq \delta(p,o)$$

- Algorithm

for each $O \in \mathrm{DB}$:

if $\mathrm{LB}(Q,O) > \epsilon$ **then** continue
else if $\delta(Q,O) \leq \epsilon$ **then** insert O into $result$

return $result$

- Filter distance LB guarantees complete results

Filter: $cand = \{o \in DB \mid LB(o,q) \leq \varepsilon\}$

Refine: $range\,(q,\varepsilon) = \{o \in cand \mid \delta(q,o) \leq \varepsilon\}$

Multistep Range Query Architecture

- Query: $range\,(DB, \delta, q, \varepsilon) = \{o \in DB \mid \delta\,(q, o) \leq \varepsilon\}$
- Use a filter distance LB for which holds that

$$\forall p, o: \mathrm{LB}(p, o) \leq \delta(p, o)$$

- Algorithm

for each $O \in \mathrm{DB}$:

if $\mathrm{LB}(Q, O) > \epsilon$ **then** continue
else if $\delta(Q, O) \leq \epsilon$ **then** insert O into $result$

return $result$

- Filter distance LB guarantees complete results

Filter: $cand = range(DB, LB, q, \varepsilon) = \{o \in DB \mid LB(o, q) \leq \varepsilon\}$
Refine: $range(DB, \delta, q, \varepsilon) = range(cand, \delta, q, \varepsilon)$

Multistep kNN Query Architecture

- How to use filter distance LB to process k-nearest-neighbor queries efficiently?
- Algorithm:
  1. generate a $ranking$ of the database in ascending order w.r.t. $\mathrm{LB}(Q,\cdot)$
  2. insert the first k objects of the $ranking$ into a $queue$
  3. if it holds that $\mathrm{LB}(Q, ranking.next()) \leq \max_{R \in queue} \delta(Q,R)$ then insert object $ranking.next()$ into the $queue$
  4. continue with 3. as long as $ranking.next() \neq \emptyset$
  5. return $queue$
- This algorithm is optimal wrt. number of refined candidates

Seidl T., Kriegel H.-P.: *Optimal Multi-Step k-Nearest Neighbor Search*. ACM SIGMOD 1998, 154-165.

Böhm C., Berchtold S., Keim D. A.: *Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases*. ACM Computing Surveys 33(3): 322-373 (2001)

Example Lower Bounds

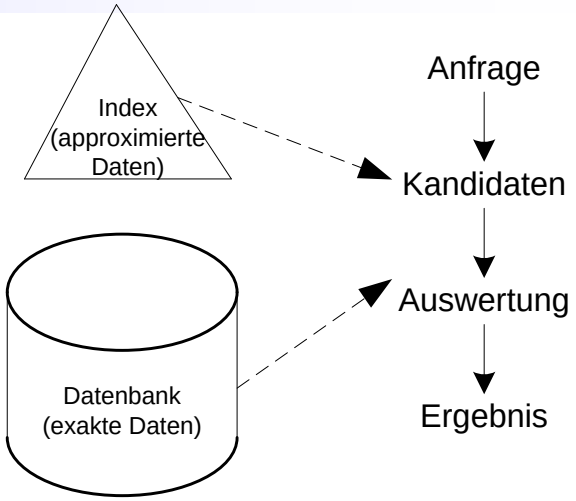
- Lower bounds for the Minkowski distance:
  - $\mathrm{LB}(X,Y) = \left(\sum_{r_i \in \mathcal{R}'} |X(r_i) - Y(r_i)|^p\right)^{\frac{1}{p}} \leq \left(\sum_{r_i \in \mathcal{R}} |X(r_i) - Y(r_i)|^p\right)^{\frac{1}{p}} = \mathrm{L}_p(X,Y)$ for $\mathcal{R}' \subseteq \mathcal{R}$ (dimensionality reduction)
- Lower bounds for the Quadratic Form distance:
  - greatest lower bounding similarity function []
  - geometric box and sphere lower bounds []
- Lower bounds for the Earth Mover's distance:
  - $\mathrm{LB}(X,Y) = \delta\left(\sum_{r_i \in \mathcal{R}_X} r_i \cdot X(r_i), \sum_{r_j \in \mathcal{R}_Y} r_j \cdot X(r_j)\right) \leq \mathrm{EMD}(X,Y)$

Beyond Domain-specific Lower Bounds

- These filter distance functions serve as lower bounds
- They depend, however, on the distance function
- Isn’t it possible to define a lower bound irrespective of one specific distance function?

Qualitätsüberlegungen

- Gütekriterien für mehrstufige Anfragebearbeitung:
  - Korrektheit: Verfahren liefert nur richtige Antworten
  - Vollständigkeit: keine richtigen Antworten fehlen
  - Effizienz: kurze Reaktionszeit



- Gütekriterien für Filter: ICES
  - **I**ndexable: Indextauglichkeit
    - Filterfuntion soll indextauglich sein
  - **C**omplete: Vollständigkeit
    - Für 100%-Vollständigkeit: keine „false dismissals“ ⇔ im Filterschritt gehen keine richtigen Antworten verloren
    - Falls < 100%: einige richtige Antworten werden nicht gefunden
      - Nicht immer notwendig: s. Web-Suchmaschinen
      - Ergebnis ist „gut“, falls der Benutzer damit zufrieden ist
    - Beispiel PAC-NN: „probably approximate (correct) nearest neighbor“
  - **E**fficient: schnelle Einzelberechnung von Filterdistanzen
    - z.B. lineare Komplexität der Filterfunktion bzgl. Dimension
  - **S**elective: möglichst kleine Kandidatenmenge
    - Filterschritt soll möglichst wenige Kandidaten erzeugen

Beispiel: Dimensionsreduktion

- Wahl der Dimensionsreduktion
  - Wähle $r_1 \ldots r_{16}$ als die 16 << 256 Dimensionen mit der größten Varianz
  - Beschränke Definition der Filterdistanz auf diese 16 Dimensionen $p' = (p[r_1], p[r_2], \ldots, p[r_{16}])$

Originalobjekte (256D) → Filterobjekte (16D)

$$d_{256}(p,q) = \sqrt{\sum_{i=1}^{256} |p[i] - q[i]|^2} \qquad d_{16}(p',q') = \sqrt{\sum_{i=1}^{16} |p[r_i] - q[r_i]|^2}$$

- Untersuche ICES für 16D-Filter

Ohne Wurzel → auch epsilon quadrieren !!!! (Übung)

  - **E**fficient: gut, lineare Komplexität von Euklid auf wenigen Dimensionen
  - **I**ndexable: gut wegen Wahl der niedrigen Filterdimension (z.B. X-tree)
    - Im Index w erdennur auf 16D reduzierte Vektoren gespeichert
  - **C**omplete: gegeben, Filter liefert untere Schranke der exakten Distanz:

$$d_{16}(p',q') = \sqrt{\sum_{i=1}^{16} |p[r_i] - q[r_i]|^2} \quad \leq \quad \sqrt{\sum_{i=1}^{256} |p[i] - q[i]|^2} = d_{256}(p,q)$$

  - **S**elective: abhängig von gewählter Reduktionsheuristik und von Daten
    - Möglicher Ansatz: Wähle Dimensionen mit größter Varianz oder Entropie
    - Gut, da Dimensionen mit großer Varianz zu großen Filterdistanzen führen

Adaptive Similarity Search in Multimedia Databases

Part 1: Similarity Queries and Object Representation

Part 2: Similarity and Dissimilarity Measures

Part 3: Efficient Similarity Query Processing

Thomas Seidl, Christian Beecks, RWTH Aachen

http://dme.rwth-aachen.de/