

1. Einführung
2. Ablauf des Multimedia Information Retrievals
3. Daten eines Multimedia-Retrieval-Systems
4. Feature
5. Eignung verschiedener Retrieval-Systeme

Besonderheiten der Verwaltung und des Retrievals von
Multimedia-Daten

Ablauf des Multimedia Information Retrievals

der Verwaltung und des Retrievals von Multimedia- Daten

im Vergleich zu Textdokumenten

Datenvolumen: großer Speicherplatzbedarf (z.B. Video-
Objekte)

implizierte Semantik: nutzerrelevante Semantik implizit
in Medien-Objekten

Heterogenität: viele Medien-Typen und Speicherformate

der Verwaltung und des Retrievals von Multimedia- Daten

im Vergleich zu Textdokumenten (Forts.)

komplexe Multimedia-Objekte: Kombination
verschiedener Medien-Typen, Referenzen und Einbettung

Ein-/Ausgabegeräte: Geräte für Datenimport und
Darstellung

Reduzierung des Datenvolumens durch Komprimierung
verlustfrei versus verlustbehaftet
schnelle Algorithmen für Vorverarbeitung der Medien-
Objekte und Ähnlichkeitsberechnung
zeitkritische Medien-Typen erfordern ausgeklügelte
Ressourcen-Verwaltung
Beispiel: Video-Archiv auf Tertiärspeicher
kompakte Präsentation (etwa Thumbnails)

nutzerrelevante Semantik ist häufig nicht explizit
z.B. Bildobjekte implizit in Pixeldaten
Extraktion von Feature-Werten nötig zur expliziten
Quantifizierung bestimmter Eigenschaften
zeitkritische Medien-Typen erfordern ausgeklügelte
Ressourcen-Verwaltung
Beispiel: Video-Archiv auf Tertiärspeicher
kompakte Präsentation (etwa Thumbnails)

Problem: Semantik ist subjektiv
z.B. Rasterbilder: ein Nutzer interessiert sich für
Gebäude, der andere für Personen
Ideal: exakte, objektive, semantik-unterstützende Feature-
Extraktion
jedoch:

- ♦ wenn automatisch
→ ungenau und low-level Semantik
- ♦ wenn manuell
→ aufwändig, subjektiv, aber high-level Semantik

Anfragesprache muss Spezifikation von Suchbedarf
ermöglichen

unterschiedliche Medien-Typen und Speicherformate
erfordern unterschiedliche Feature-Extraktion und
Ähnlichkeitsberechnungen
System muss Erweiterung um neue Typen und Formate
ermöglichen
Datenunabhängigkeit: medientyp- und
speicherformatunabhängige Anfragen
z.B. zeige alle Medien-Objekte zu einer Person an

Medien-Umsetzung: etwa Umwandlung Text in Audio
Speicherformatumwandlung
spezielle Metadaten bzgl. Medien-Typ und Speicherformat
notwendig

komplexes Medien-Objekt: enthält mehrere eigenständige
Medien-Objekte
Aggregation durch Referenzierung und Einbettung
für Verwaltung: Zerlegung in nichtkomplexe Medien-
Objekte und Strukturdaten

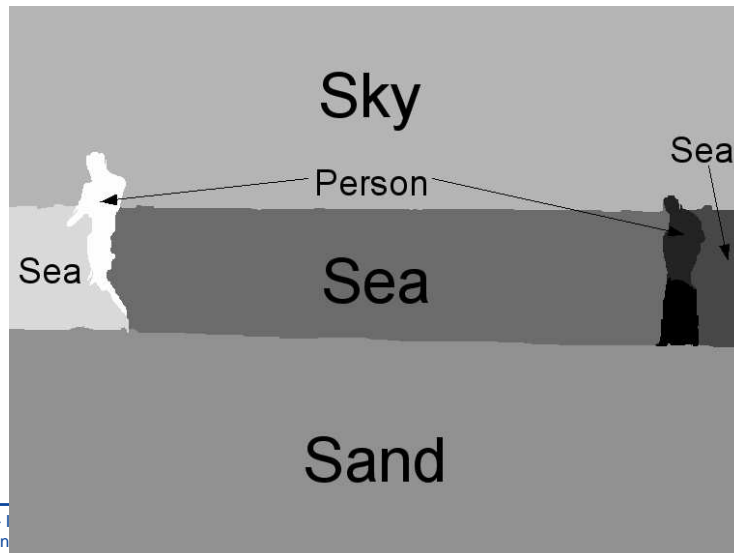
Verwaltung von Strukturdaten zur Rekonstruktion und
Retrieval

- ♦ Beispiel: Word-Dokument enthält Text und Bilder
- ♦ Beispiel: Comics enthalten Figuren und Sprechblasen
(Text)

Aggregation aufgrund Segmentierung

- ♦ Beispiel: Segmentierung auf Bildern in
Vordergrundobjekte und Hintergrund





Abhängigkeit von Medien-Typ und Speicherformat

- ♦ Bild → Bildschirm,
- ♦ Ton → Lautsprecher, ..

System muss Gerät des Nutzers kennen

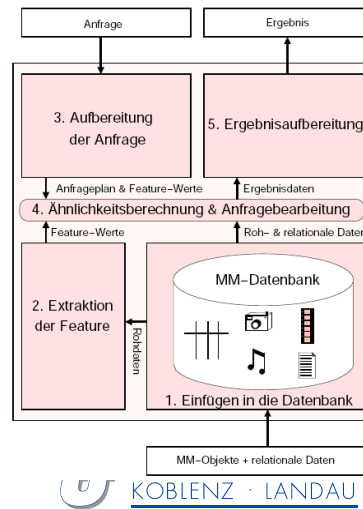
→ Nutzer- und Hardware-Profile mit Präferenzen

System verwendet Profil zur

- ♦ Unterstützung der Anfrageformulierung
- ♦ garantierte Darstellbarkeit von Anfrageergebnissen erfordert u.U. Medienumsetzung oder Speicherformatumwandlung
- ♦ Optimierung: Reduzierung benötigter Ressourcen etwa Netzwerkbandbreite

Besonderheiten	Herausforderungen
Datenvolumen	Komprimierung kompakte, interne Darstellung schnelle Ähnlichkeitsberechnung Ressourcenverwaltung kompakte Präsentation
implizite Semantik	Feature-Extraktion Subjektivität Ungenauigkeit Anfrageformulierung
Heterogenität	Medien-Typen und Speicherformate Medien-Typunabhängigkeit und Medien-Umsetzung Speicherformatunabhängigkeit und Speicherformatumwandlung Metadaten zur Interpretation
komplexe Multimedia-Objekte	Zerlegung und Rekonstruktion Strukturdaten Segmentierung
Ein-/Ausgabegeräte	Nutzer- und Hardware-Profile Nutzerpräferenzen Anfrageformulierung Darstellbarkeit Optimierung

4.2 Ablauf des Multimedia Information Retrieval



1. Einfügen in die Multimedia-Datenbank
2. Extraktion der Feature-Werte
3. Aufbereitung der Anfrage
4. Anfragebearbeitung und Ähnlichkeitsberechnung
5. Ergebnisaufbereitung

Dynamische vs statische Feature-Extraktion <isweb>

statisch: Anfrageaufbereitung nach der Feature-Extraktion

- Vorteil: Geschwindigkeit durch Verwendung spezieller Indexstrukturen
- Nachteil: unflexibel bzgl. unvorhersehbaren Ähnlichkeitsanfragen vordefinierte Features: z.B. in ORDBMS Farbverteilung, Textur, Gestalt

dynamisch: Feature-Extraktion nach der Anfrageaufbereitung

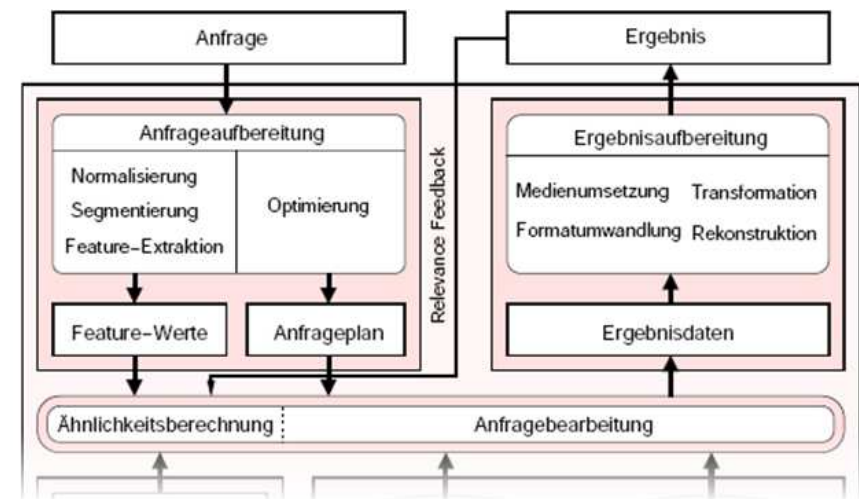
- Vorteil: flexible Anpassung der Feature-Extraktion an Anfrage
- Nachteil: hoher Aufwand

Kombination

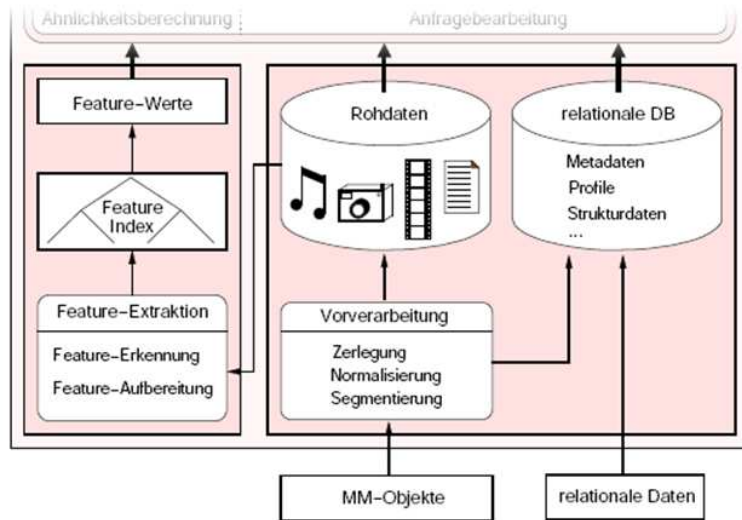
- vorhandene Feature-Werte verwenden
- wenn nötig, neue extrahieren und aufbewahren

Verwaltung & Retrieval von Multimedia-Daten: <isweb>

Ablauf (1)



Ablauf (1)



- Unterscheidung Multimedia-Objekte, Metadaten, Strukturdaten und relationale Daten
- Metadaten: etwa bzgl. Medien-Typ, Speicherformat, Profildaten
- relationale Daten: etwa Namen von abgebildeten Personen, Aufzeichnungszeitpunkt

- Zerlegung komplexer Multimedia-Objekte in nichtkomplexe Multimedia-Objekte und Strukturdaten
- Normalisierung der Medien-Objekte zur Unterdrückung von Störfaktoren
 - z.B. Verzerrungen bei Bildaufnahmen, Hintergrundgeräusche bei Audio-Daten
- Segmentierung der Medien-Objekte
 - z.B. Bildsegmentierung in Regionen, Videosegmentierung in Szenen

- Feature-Erkennung: Algorithmen zur Ableitung von Feature-Werten
 - ◆ Abhängigkeit von Medien-Typ und Speicherformat, beabsichtigte Ähnlichkeitsberechnung
 - ◆ Beispiel: Gabor-Filter für Texturen, Bewegungsvektoren in Video-Objekten

Feature-Aufbereitung

- ♦ Skalierung
- ♦ Entfernen von Abhängigkeiten
- ♦ Entfernen überflüssiger oder vernachlässigbarer Feature-Werte
z.B. KLT, LSI

Ablegen im Feature-Index

Anfrage zerlegbar in klassische und Ähnlichkeitsanfrage
Optimierung der klassischen Anfrage
Ähnlichkeitsanfrage anhand Anfrageobjekt: Schritte wie
beim Einfügen in Datenbank und Extraktion

Ähnlichkeitsanfrage: Berechnung der ähnlichen Objekte
anhand Ähnlichkeitsfunktion unter Ausnutzung Feature-
Index

bei komplexen Ähnlichkeitsanfragen: Zerlegung und
anschließende Kombination der Ergebnissequenzen

(2)

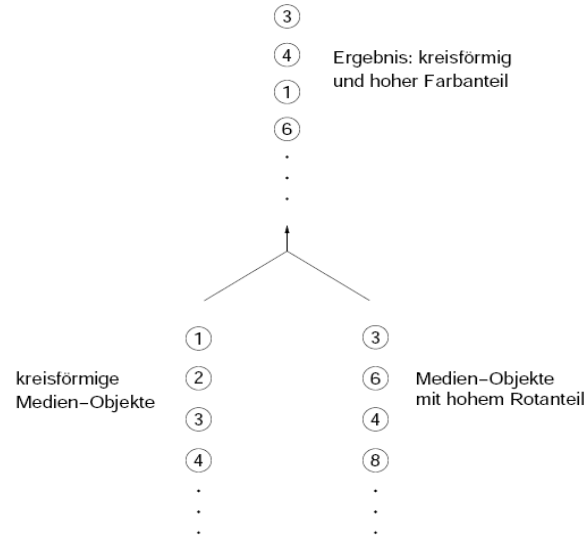
Kombination auch mit Ergebnis der klassischen
Datenbankanfrage

Beispiel: Suche nach Gemälde von Caspar David
Friedrich mit Abendstimmung

Problem:

klassisches Retrieval + Ähnlichkeitssuche
(mögliche Erweiterungen:
probabilistische Relationale Algebra,
Fuhr et al.)

Suche nach Bildern mit Sonnenuntergang



Anpassung an Vorgaben des Anwenders (Nutzer- und Geräteprofile)

Medienumsetzung, Formatumwandlung, Transformationen, Rekonstruktion komplexer Objekte

Optimierung

- ◆ Medien-Daten in verschiedenen Formaten vor Anfrage anlegen
- ◆ Reihenfolge von Transformationsschritten

zeitliche und räumliche Anordnung der Ergebnisobjekte

Interaktion mit Nutzer gewährleisten

z.B. play, stop, rewind bei Video-Objekten

Ergebnisbewertung

Anpassung von Parametern des Systems anhand der Bewertungen

Durchführung der Iterationen

Klassifikation	Daten	Beispiele
Multimedia-Objekt (Strukturdaten)	nicht-textuell	Graphen
	textuell	XML, HTML, VRML, X3D
Medien-Objekt	Dokument	Freitext, XML, HTML, VRML, X3D
	nicht-textuell	Videos, Bilder, Audio-Daten
weitere Daten	Nutzerprofil	Präferenzen
	Geräteprofil	Geräteauflösung, Gamma-Werte

Medien-Objekt: Daten eines Medien-Typs textuell oder nicht-textuell

Multimedia-Objekt: Kombination mehrerer Medien-Objekte
Zerlegung in Medien-Daten und Strukturdaten sinnvoll

Strukturdaten: nicht-textuell durch Graphen oder textuell, etwa durch Hypertext-mechanismen von Markup-Sprachen
weitere Daten

HTML-Datei:

```
<HTML>
  <HEAD>
    <TITLE> Franz Klammer </TITLE>
  </HEAD>

  <H2> Franz Klammer </H2>
  Universität Klammerstadt <BR>
  Fakultät für Informatik <BR>
  12345 Klammerstadt <BR>
  <BR>
  <B> Telefon: </B> ++49-123-45678 <BR>
  <B> Fax: </B> ++49-123-45670 <BR>
  <img align = 'right' src = 'franz.jpg' width = 150>
</HTML>
```

Zerlegung in mo-001.html:

```
<HTML>
  <H2> Franz Klammer </H2>
  Universität Klammerstadt <BR>
  Fakultät für Informatik <BR>
  12345 Klammerstadt <BR>
  <BR>
  <B> Telefon: </B> ++49-123-45678 <BR>
  <B> Fax: </B> ++49-123-45670 <BR>
</HTML>
```

Beispiel: Zerlegung eines Multimedia-Objekts (3)

und Strukturdaten:

```
<HTML>
  <HEAD>
    <TITLE> Franz Klammer </TITLE>
  </HEAD>
  <A HREF = mo-001.html> Link1 </A>
  <img align = 'right' src = 'franz.jpg' width = 150>
</HTML>
```

Klassifikation	Metadaten	Beispiele
inhaltsbeschreibend (interpretierend)	kontextbeschreibend	Indexvokabular, Ontologien, Thesaurus
	kontextbezogen	Identifikation, Raum- und Zeitdaten
	objektbeschreibend, nicht-textuell	Gegenstände, Personen, Eindrücke, Aktivitäten, Titel
	objektbeschreibend, textuell	Annotation, Drehbuch, Untertitel
inhaltsbezogen (nicht interpretierend)	Feature	Farbverteilung, Textur, Klangdynamik, Form
	Segmentspezifikation	Anfang und Ende einer Video-Szene, Umriss eines Bildausschnitts
inhaltsunabhängig	präsentationsbezogen	QoS, Auflösung, Layout
	aufnahmebezogen	Urheber, Aufnahmegerät
	speicherungsbezogen	Medien-Typ, Speicherformat, Speicherort

inhaltsunabhängig:
wichtig für Verwaltung und Interpretation

inhaltsbezogen:
Semantik auf niedriger Abstraktionsstufe, aber automatisch Extrahierbar

inhaltsbeschreibend: Semantik auf hoher Abstraktionsstufe, kaum automatisch extrahierbar

- ♦ *objektbeschreibend*: Beschreibung einzelner Medien-Objekte
- ♦ *kontextbezogen*: Beschreibung von Medien-Objekten bezogen auf Kollektion von Medien-Objekten
- ♦ *kontextbeschreibend*: Beschreibung einer Kollektion von Medien-Objekten

Beispiel inhaltsunabhängige und -bezogene Metadaten

Klassifikation	Metadaten	mögliche Werte
inhaltsbezogen	Feature	Anzahl _{
} = 6 Anzahl _{<H2>} = 1
inhaltsunabhängig	präsentationsbezogen	Browser = 'Netscape'
	aufnahmebezogen	Urheber = 'Klammer', Aufnahmegerät = 'emacs'
	speicherungsbezogen	Medien-Typ = 'Text', Speicherformat = 'HTML', Speicherort = 'http://...'

Beispiel inhaltsbeschreibende Metadaten

textuelle Objektbeschreibung: Multimedia-Objekt enthält Kontaktinformation und Passfoto von Mitarbeiter Franz Klammer

nicht-textuelle Objektbeschreibung: Relation mit Namen und Adressen von auf Passfotos dargestellten Personen

Beispiel inhaltsbeschreibende Metadaten (2)

kontextbeschreibende Metadaten: Kollektion enthält Mitarbeiterdaten einer Abteilung

kontextbezogene Metadaten: Zeitpunkt der Aufnahme der jeweiligen Fotos

Rohdaten:

Daten der Medien-Objekte

Registrierungsdaten: speicherungsbezogene Metadaten

Beschreibungsdaten:

inhaltsbeschreibende und inhaltsbezogene Metadaten

relationale Anfrage: SQL-Anfrage an relationales DBS

nicht-textuelle Strukturdaten, weitere Daten,

inhaltsunabhängige und nicht-textuelle,

inhaltsbeschreibende Metadaten

Anfragen an semi-strukturierte Textdaten

(XML, HTML, VRML)

Text-Retrieval

textuell formulierte Daten und Metadaten

Multimedia-Retrieval

erweiterung Text-Retrieval um weitere Medien-Typen

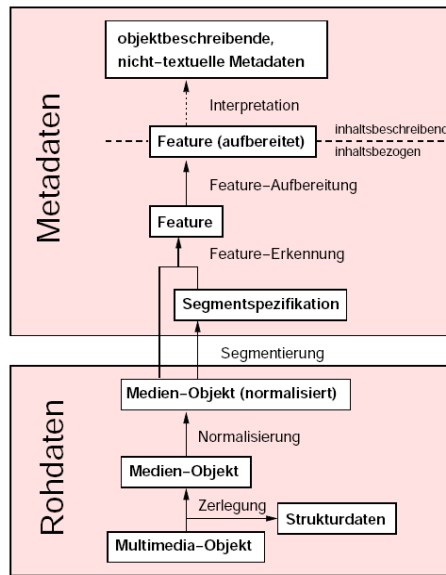
Browsen

sequentielles Durchblättern der Datenbank

Feature-Extraktion und Invarianzen

Feature-Datentypen

Feature-Aufbereitung



begrenzte semantische Ausdruckskraft automatisch extrahierter Feature-Werte

→ keine Objekterkennung i.A. möglich

Lücke zwischen menschlicher Inhaltsbeschreibung und Ausdruckskraft automatisch abgeleiteter Feature:
semantische Lücke

Beispiel: Bildsuche nach Fliegenpilz anhand Farbverteilung und Form von Bildsegmenten; Fehler bei rotweißem Spielball

Fazit: Feature repräsentieren i.A. low-level-Semantik

Datentyp für Medienobjekte: $t_{MO} \in \mathcal{T}$

Menge von N Medienobjekten:

$$MO = \{mo_1, mo_2, \dots, mo_N\} \subseteq \text{dom}(t_{MO})$$

Datentyp für Segmentspezifikation: $t_{SS} \in \mathcal{T}$

Menge von Segmentierungsfunktionen bezüglich Algorithmus A :

$$SEG = \{seg_A \mid seg_A \text{ ist Segm.-funktion}\}$$

Segmentierungsfunktion bildet Medienobjekt auf Menge von Segmentspezifikationen ab:

S ist Menge von Segmenten:

$$\text{seg}_A : \text{dom}(t_{MO}) \rightarrow \text{dom}(\text{set}(t_{SS}))$$

$$S = \{s_A \mid s_A \in \text{dom}(\text{tuple}(mo:t_{MO}, ss:t_{SS})) \\ \text{mits}_A.ss \in \text{seg}_A(s_A.mo)\}$$

Menge von Feature-Funktionen

Feature-Datentyp für Eigenschaft E:

$$F = \{f_E \mid f_E \text{ ist eine Feature-Funktion}\}$$

Feature-Funktion:

$$t_{f_E} \in \mathcal{T}$$

$$f_E : \text{dom}(\text{tuple}(mo:t_{MO}, ss:t_{SS})) \rightarrow \text{dom}(t_{f_E})$$

Sammlung von Rasterbildern

Datentyp für Rasterbild:

Segmentierung und Feature-Extraktion anhand globaler,

lokaler Farbverteilung, Form und $T_{trb} \in \mathcal{T}$

Feature-Funktionen:

$$f_{gf}, f_{lf}, f_{form}, f_{texture}$$

Sammlung von Rasterbildern (2)

vereinfachte Feature-Datentypen:

$$t_{gf} = t_{lf} = \text{tuple}(\text{rot} : \text{double}, \text{grün} : \text{double}, \text{blau} : \text{double}).$$

$$t_{form} = \text{tuple}(\text{kreis} : \text{double}, \text{rechteck} : \text{double}, \\ \text{inhalt} : \text{double}, \text{umfang} : \text{double}).$$

$$t_{texture} = \text{tuple}(\text{granulat} : \text{double}, \text{reg} : \text{double}, \\ \text{ausrichtung} : \text{double}).$$

Anforderungen an Feature-Extraktion

Adäquatheit: Werte müssen Eigenschaft angemessen ausdrücken

effiziente Berechnung

Berücksichtigung von Invarianzen:

Unabhängigkeit von ungewollten Eigenschaften

Anforderungen an Feature-Extraktion (Forts.)

Minimalität: minimale Anzahl von Feature-Werten

Orthogonalität: orthogonale Feature-Werte

Feature-Extraktion =

Feature-Erkennung + Feature-Aufbereitung

Feature-Extraktion	Anforderung
Feature-Erkennung	Adäquatheit Effizienz Invarianzen
Feature-Aufbereitung	Minimalität Orthogonalität

Beispiele von Störungen:

verfälschende Lichtverhältnisse bei Foto-Aufnahmen

Knackgeräusche bei Tonaufnahmen

Artefakte bei Videoaufzeichnungen

Unabhängigkeit von bestimmten Operationen, z.B.:
 Rotation, Translation, Skalierung von Bildern
 Farbverschiebung bei farbbasierten Features
 Lautstärkeänderung bei gesprochenen Texten

Feature-Extraktionsfunktion
 soll unabhängig von parametrisierbarer
 Transformationsfunktion g sein *invariant*
 Menge G enthalte alle anhand g parametrisierten
 Transformationsfunktionen

invariante Extraktionsfunktion:

$$f_{invariant}(o) = \frac{1}{|G|} \int_G f(g(o)) dg.$$

diskrete Parametrisierung:

$$f_{invariant}(o) = \frac{1}{|G|} \sum_{g \in G} f(g(o)).$$

Feature-Extraktion f auf Grauwertbild I

Invarianz bzgl. Translation und Rotation von Punkten p

$$g^{x,y,\varphi}(p) = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} p + \begin{pmatrix} x \\ y \end{pmatrix}$$

Anwendung auf $m \times n$ -Grauwertbild mit Modulo-Berechnung:

Optimierungen zur effizienten Berechnung möglich (hier nicht diskutiert)

$$f_{invariant}(I) = \frac{1}{m * n * 360} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \sum_{\varphi=0}^{359} f(g^{x,y,\varphi}(I))$$

allgemeine Invarianzen

werden eher generell für Feature gefordert (etwa Störfaktoren)

- ◆ Normalisierung beseitigt Störfaktoren
- ◆ Beispiel: Unabhängigkeit von Textlänge bei Textdokumenten
- ◆ Beispiel: Auflösung und Dateiformat von Rasterbildern

spezielle Invarianzen

werden eher für bestimmte Feature gefordert

- ◆ Beispiel Meeresaufnahmen: Ausrichtung der Wellen sei irrelevant
- ◆ Beispiel Suche nach Stoffmustern: Invarianz bzgl. Rotation

spezielle Invarianzen (Forts.)

- ◆ Problem mit speziellen Invarianzen: Abhängigkeit von konkreter Anfrage
 - schwer vorhersehbar
 - dynamische Feature-Extraktion

Abhängigkeit von gewünschter Eigenschaft und Extraktionsalgorithmus
häufige Datentypen:

- Punkt: `array [1..n] (real)`
- Binärdaten: `array [1..n] (boolean)`
- Intervall: `array [1..n] (tuple (unten : real, oben : real))`
- Sequenz: `list (t)` mit $t \in \mathcal{T}$
- 2-dimensionale Region: `list (tuple (x : real, y : real))`
- Histogramm

Punktdatentyp ist relativ universell und Ausgangspunkt vieler mathematischer und algorithmischer Verfahren

Intervall: pro Intervall zwei Dimensionen

- ♦ unten, oben sind einzelne Werte
Problem Abhängigkeit: $\text{unten} \leq \text{oben}$
- ♦ Mittelwert und Intervallbreite
Problem Abhängigkeit am Rand des Wertebereichs

Binärdaten: pro Eigenschaft eine Dimension mit Wertebereich $\{0,1\}$

Sequenz und Region:

Problem variable Werteanzahl; ein Ausweg:

- ♦ Anwendung diskreter Fouriertransformation
- ♦ Abbilden auf feste Anzahl Fourierkoeffizienten
- ♦ Vorteil: Kompaktheit, Abbildungsfehler nicht lokalisierbar

häufig verwendeter Datentyp

legt Menge disjunkter n -dimensionaler Intervalle (Bins) fest

pro Bin ein Häufigkeitswert

Beispiel: Farbhistogramm im 3-dimensionalen Farbraum

Varianten von Histogrammen:

- ♦ feste Anzahl einheitlicher Intervalle
→ Punkdatentyp
- ♦ variable Anzahl und variable Intervalle

```
list(tuple(intervall: t_intervall,
wert : integer))
```

oder Darstellung Intervall durch Zentroid (Punkt)

```
list(tuple(punkt: t_punkt, wert : integer))
```

Varianten von Histogrammen (Forts.):

- ♦ Invarianz durch relative statt absolute Häufigkeiten (real statt integer)

Motivation Aufbereitung

fehlende Minimalität

- ♦ hoher Speicherplatzverbrauch
- ♦ hoher Berechnungsaufwand
- ♦ ineffiziente Indexstrukturen (Fluch der hohen Dimensionen)

fehlende Orthogonalität

- ♦ Abhängigkeiten innerhalb Feature-Werte möglich
- ♦ Problem: keine Manipulation von Werten isoliert möglich
→ schwere Realisierung etwa von Invarianzen

4.5 Eignung verschiedener Retrieval-Modelle <is web>

boolesches Modell
Fuzzy-Modell
Vektorraummodell

klassische Retrieval-Modelle wurden für Text-Retrieval entwickelt

Boolesches Modell

<is web>

Eignung:

basiert auf Mengentheorie und boolescher Algebra
keine Unterstützung gradueller Ähnlichkeitswerte

Boolesches Modell (2)

<is web>

Semantik ist für viele Anwendungen zu scharf
Negativbeispiel: Bildähnlichkeitssuche
Vorteil: Konstruktion komplexer Anfragen mittels boolescher
Junktoren

Fuzzy-Modell

<is web>

Eignung:

Verallgemeinerung boolesches Modell um graduelle
Zugehörigkeitswerte (Ähnlichkeitswerte)
Junktoren auf Zugehörigkeitswerten ermöglichen Konstruktion
komplexer Anfragen

Import von Zugehörigkeitswerten (Ähnlichkeitswerten) aus Vektorraummodell
ermöglicht Einbindung relationaler Anfragen
Beispiel: Suche nach Gemälde von Caspar David Friedrich mit Abendstimmung

Eignung:
Feature-Werte von Medienobjekten als geometrische Objekte im Vektorraum
Verfahren der linearen Algebra für Retrieval einsetzbar

gute Grundlage zur Ermittlung von Ähnlichkeitswerten
viele Distanz- und Ähnlichkeitsfunktionen verfügbar

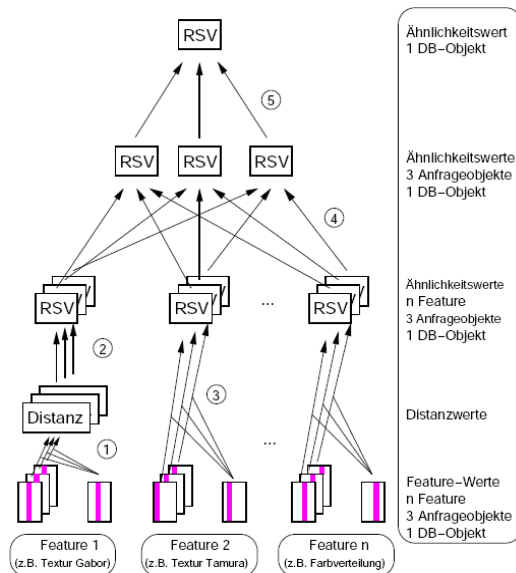
viele Möglichkeiten zur Berechnung von Ähnlichkeitswerten
hier vereinfachtes, verallgemeinertes Multimedia-Modell

Voraussetzungen:

- ♦ Feature-Werte gruppiert nach Feature und Extraktionsverfahren
- ♦ Ähnlichkeitswert (RSV für *retrieval status value*) pro Medienobjekt und Ähnlichkeitsanfrage ein Ähnlichkeitswert aus [0,1]

Voraussetzungen (Forts.):

- ♦ mehrere Feature-Gruppen: Anfrage mit mehreren Gruppen
- ♦ mehrere Anfrageobjekte
z.B. aufgrund Nutzerprofil, positiv bewertete Objekte bei Anfrage-Iteration



Ähnlichkeitswert
1 DB-Objekt

Ähnlichkeitswerte
3 Anfrageobjekte
1 DB-Objekt

Ähnlichkeitswerte
n Feature
3 Anfrageobjekte
1 DB-Objekt

Distanzwerte

Feature-Werte
n Feature
3 Anfrageobjekte
1 DB-Objekt