

# Relevance Feedback Methoden

Julia Eckert

Vortrag innerhalb des Seminars  
„Inhaltsbasierte Bildsuche“  
Universität Freiburg, WS2004/2005

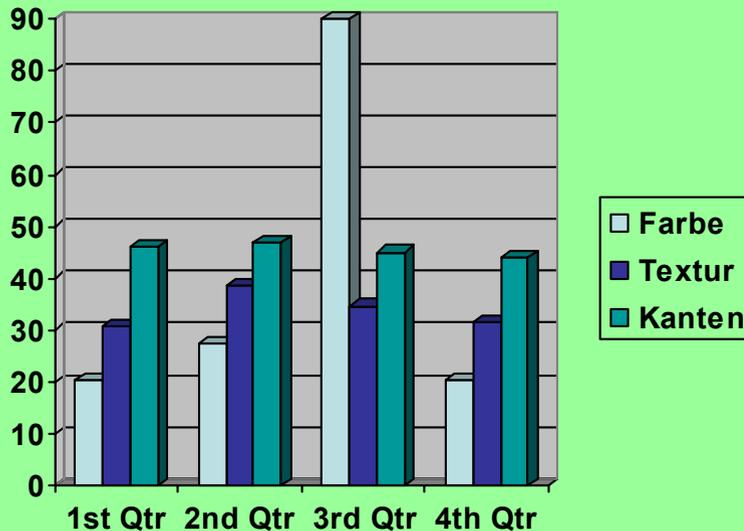
# Gliederung

- Motivation
- Theoretische Ansätze
  - Wechselndes Anfrageobjekt
  - Verwendung von Klassifikatoren
  - Gewichtungsmethode
- Implementierung
- Ergebnisse
- Diskussion

# Motivation

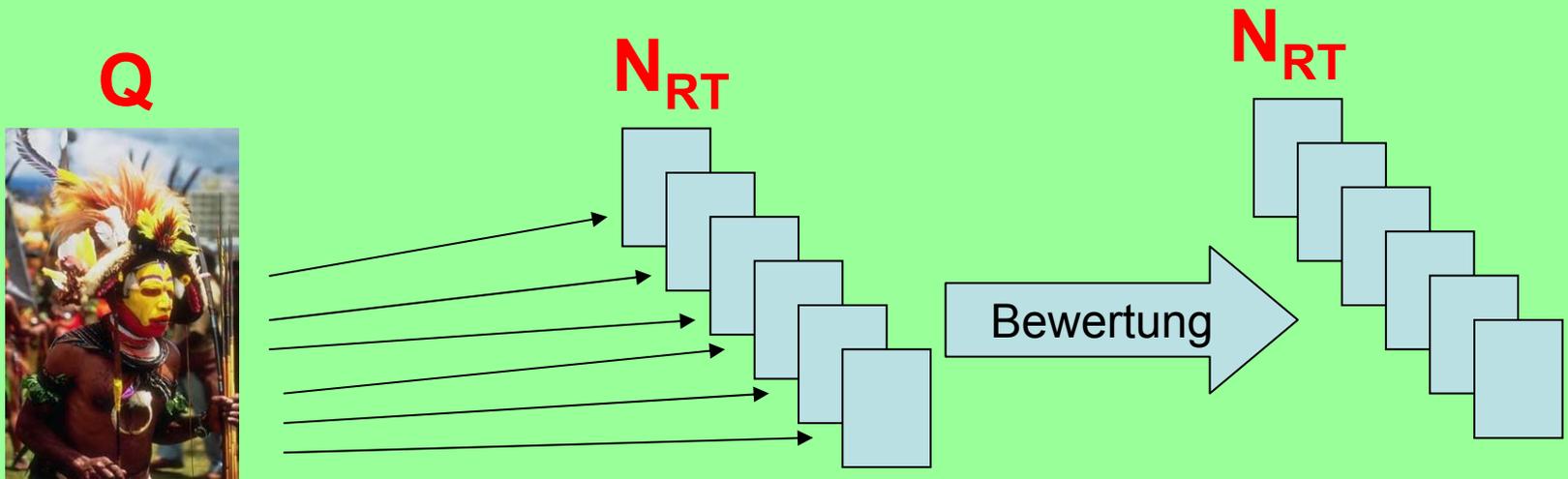
Diskrepanz:

Low-level-Feature → subjektive Bewertung



# Relevance Feedback in CBIR-Systemen:

- Kombination aus beiden
- mehrere iterative Schritte



# Gliederung

- Motivation
- Theoretische Ansätze
  - Wechselndes Anfrageobjekt
  - Verwendung von Klassifikatoren
  - Gewichtungsmethode
- Implementierung
- Ergebnisse
- Fragen

# Relevance Feedback Methoden

- Wechselndes Anfrageobjekt

Bewertung → neue Anfrage

- Klassifikatoren am Beispiel von SVM

Bewertung → Relevant/Irrelevant

[vgl. J. Ick, 2004]

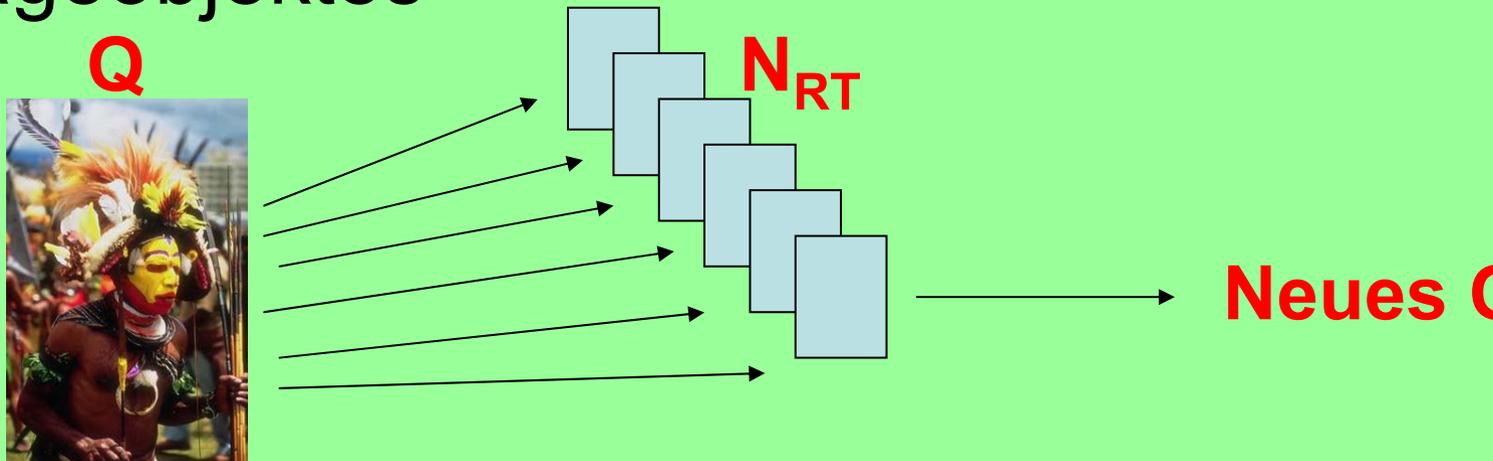
- Gewichtungsmethode

Bewertung → unterschiedliche Gewichte

[vgl. Y. Rui u.a., 1998]

# Wechselndes Anfrageobjekt

- Start: Anfrageobjekt  $Q$
- Rückgabe: die  $N_{RT}$  ähnlichsten Bilder
- Daraus: Wahl eines neuen Anfrageobjektes



# Wechselndes Anfrageobjekt

## Nachteil dieser Methode:

- System startet jedes mal neu
- Die Suche kann somit sehr lange fort dauern

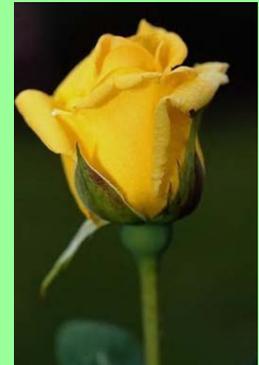
Q



Q

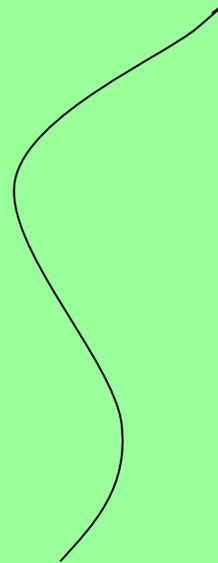


Q



# RF mit Support Vektor Maschinen

- Aufteilung in relevante und irrelevante Bilder
- Ordnung der Bilddatenbank

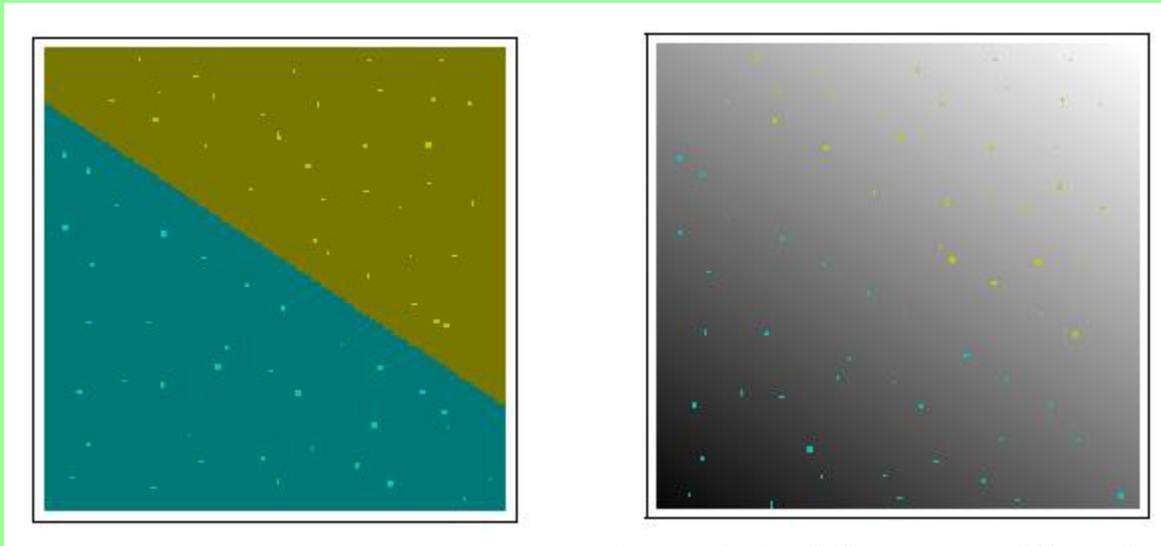


# RF mit Support Vektor Maschinen

- Zuordnung der Repräsentanten-Vektoren einer Beispiel-Sequenz :
  - positive Klasse (= 1);
  - negative Klasse (= -1);
- Gesucht: Funktion, die alle unbewerteten Elemente der Datenbank einer der beiden Klassen zuordnet.

# RF mit Klasse-Zwei-SVM:

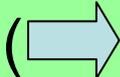
Separierung der beiden Klassen erfolgt durch eine Hyperebene in einem höherdimensionalen Feature-Raum:  
Die  $N_{RT}$  besten Bilder liegen auf der positiven Seite am nächsten zur trennenden Hyperebene:  
(  $\Rightarrow$  Support Vektoren)

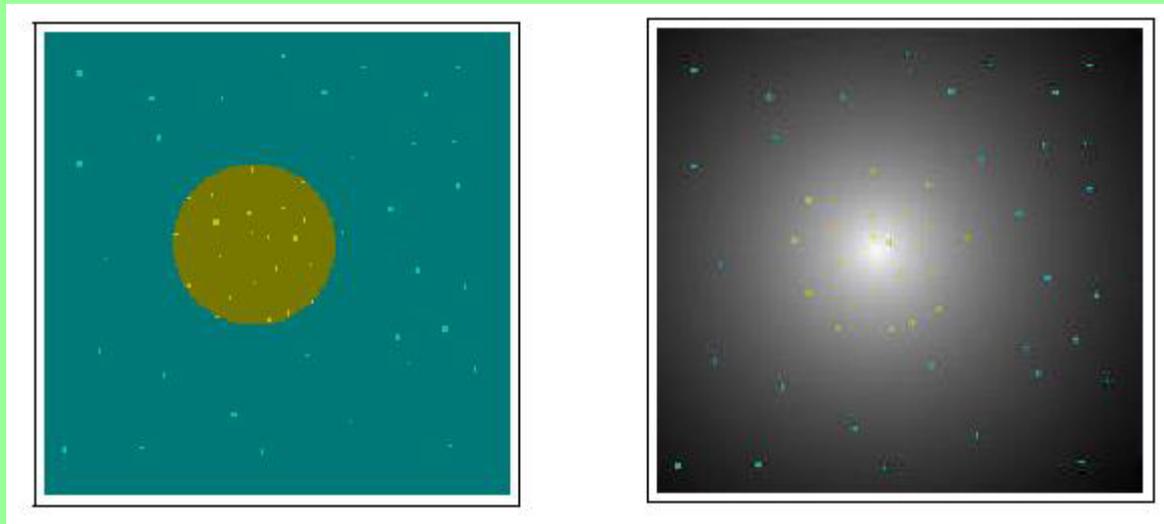


# RF mit Klasse-Eins-SVM:

Separierung der relevanten Bilder durch eine Hypersphäre in einem höherdimensionalen Feature-Raum:

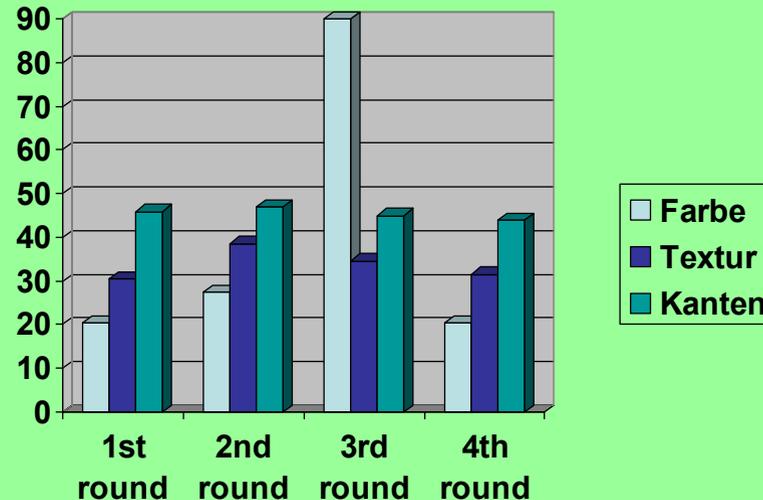
Die  $N_{RT}$  besten Bilder liegen am nächsten zum Mittelpunkt der trennenden Hypersphäre:

( Support Vektoren)



# Gewichtungsmethode

- Bewertung → Feature-Analyse
- Feature-Analyse → neue Gewichtung



# Formalisierung der Bildobjekte

$$O = O(D, F, R)$$

- $D$  = Rohdaten des Bildobjektes
- $F = \{ f_i \}$ , Menge der low-level Feature
- $R = \{ r_{ij} \}$ , Menge der Repräsentanten
- $r_{ij} = [r_{ij1}, \dots, r_{ijk}, \dots, r_{ijK}]$
- Ähnlichkeitsmaß  $M = \{ m_{ij} \}$

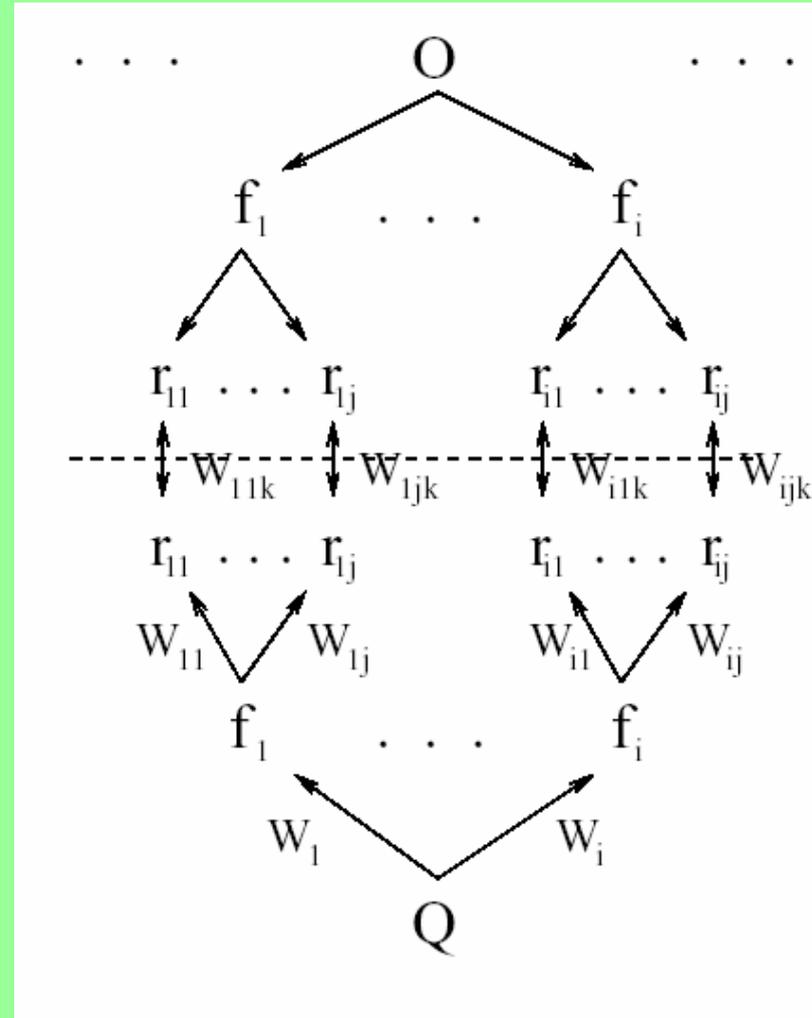
# Features

- $f_1$  = Farbe;
- $f_2$  = Textur;
- $f_3$  = Kantenmerkmale;
  
- $r_{11}$  = Farb-co-occurrence Matrizen;
- $r_{12}$  = Farbmomente;
- $r_{21}$  = Textur;
- $r_{22}$  = Invariante Merkmale;
- $r_{31}$  = Kantenbasierte Merkmale;

# Anfrage-Objekt Q

## Gewichtung

- $f_i \longrightarrow W_i$
- $r_{ij} \longrightarrow W_{ij}$
- $r_{ijk} \longrightarrow W_{ijk}$



# Initialisierung der Gewichte

$$W_i = W0_i = \frac{1}{I} \quad I = \text{Anzahl der Feature}$$

$$W_{ij} = W0_{ij} = \frac{1}{J_i} \quad J_i = \text{Anzahl der Repräsentanten}$$

$$W_{ijk} = W0_{ijk} = \frac{1}{K_{ij}} \quad K_{ij} = \text{Länge des Repräsentanten-Vektors}$$

# Berechnungsvorschriften

Objektähnlichkeit:

$$S(r_{ij}) = m_{ij}(r_{ij}, W_{ijk})$$

Featureähnlichkeit:

$$S(f_i) = \sum_j W_{ij} S(r_{ij})$$

Gesamtähnlichkeit:

$$S = \sum_i W_i S(f_i)$$

# Normalisierung

- Intra-Normalisierung:  
Normalisierung der einzelnen Vektor -  
Komponenten  $r_{ijk}$  innerhalb des  
Repräsentanten-Vektors  $r_{ij}$ .
- Inter-Normalisierung:  
Normalisierung der einzelnen Objekt-  
Ähnlichkeiten  $S(r_{ij})$  innerhalb der  
Gesamtähnlichkeit  $S$ :

# Intra-Normalisierung

Sei  $M =$  Größe der Bilddatenbank und  $V := r_{ij}$ ;

Repräsentanten für Bild  $m$ :  $V_m =$

$[V_{m,1}, \dots, V_{m,k}, \dots, V_{m,K}]$ ;

Alle Vektoren  $V_m$  in  
einer Matrix:

$$v = [v_{m,k}] = \begin{pmatrix} v_{1,1} & \cdot & \cdot & \cdot & v_{1,K} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ v_{m,1} & & & & v_{m,K} \end{pmatrix};$$

# Normalisierung der Zeilen von $V$

Gaußsche Normalisierung:

$$v_{m,k} = \frac{v_{m,k} - \mu_k}{3\sigma_k};$$

# Inter-Normalisierung

Berechnung der Ähnlichkeit  $S_{m,n}(r_{ij})$  für jedes Bildpaar:

$$S_{m,n} = m_{ij}(r_{ij}, W_{ijk})$$

mit  $\mu_{ij}$  und  $\sigma_{ij}$  von  $S_{m,n}$ .

Ähnlichkeitsmaß des Anfrageobjektes Q:

$$S'_{m,Q}(r_{ij}) = m_{ij}(r_{ij}, W_{ijk})$$

# Inter-Normalisierung

Gaußsche Normalisierung:

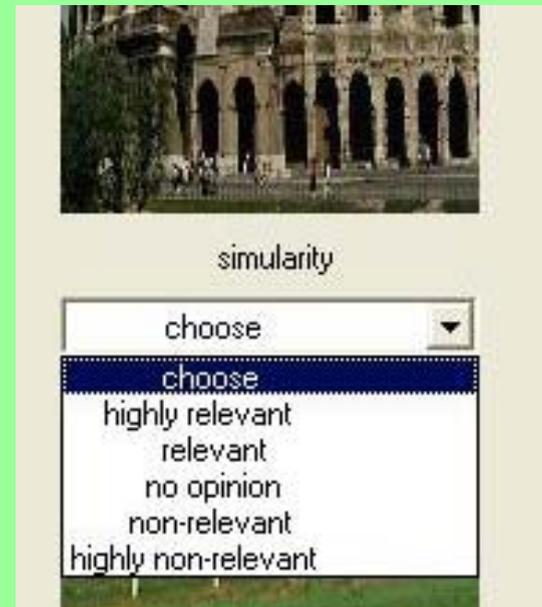
$$S'_{m,Q}(r_{ij}) = \frac{S_{m,Q}(r_{ij}) - \mu_{ij}}{3\sigma_{ij}};$$

$$S''_{m,Q}(r_{ij}) = \frac{S'_{m,Q}(r_{ij}) + 1}{2};$$

# Feedback Runde

- Objekte der Datenbank sortiert nach Gesamtähnlichkeit
- Der Anwender bewertet die ausgegebenen Bilder mit:

- „highly relevant“
- „relevant“
- „no opinion“
- „non-relevant“
- „highly non-relevant“



# Gewichts Update

- Update von  $W_{ij}$  (inter-weight);  
Bewertungen innerhalb der  
Gesamtähnlichkeit  $S$ .
- Update von  $W_{ijk}$  (intra-weight);  
Beiträge der einzelnen Komponenten  $r_{ijk}$   
innerhalb des Repräsentantenvektors  $r_{ij}$ .

# Score

= 3, falls: „highly relevant“

= 1, falls: „relevant“

Score<sub>*I*</sub> = 0, falls: „no opinion“

= -1, falls: „non-relevant“

= -3, falls: „highly non-relevant“

$I = 0, \dots, NRT;$

# inter-weight:

Die  $N_{RT}$  ähnlichsten Objekte gemäß der Gesamtähnlichkeit  $S$ :

$$RT = [RT_1, \dots, RT_1, \dots, RT_{NRT}];$$

---

Die  $NRT$  ähnlichsten Objekte gemäß der Objektähnlichkeit  $S(r_{ij})$ :

$$RT^{ij} = [RT^{ij}_1, \dots, RT^{ij}_1, \dots, RT^{ij}_{NRT}];$$

# Update von $W_{ij}$ :

- Initialisierung:  $W_{ij} = 0$ ;
- $W_{ij} = W_{ij} + \text{Score}_l$ , falls  $RT^{ij}_l$  ist in RT;  
=  $W_{ij} + 0$ , sonst;

$$l = 0, \dots, N_{RT};$$

- Falls  $W_{ij} < 0$ , auf 0 setzen;
- $W_{Tij} = \sum W_{ij}$ ;
- Normalisierung:  $W_{ij} = W_{ij} / W_{Tij}$ ;

# intra-weight:

- $M'$  = mit „highly relevant“ oder „relevant“ markierte Bilder
- Zusammen mit den Repräsentanten  $r_{ij}$  bilden diese  $M'$  Bilder eine  $M' \times K$  Matrix.
- Spalten:  $r_{ijk}$  Sequenz der Länge  $M'$ .

# Update von $W_{ijk}$ :

- Inverse der Standardabweichung der  $r_{ijk}$  Sequenz

$$W_{ijk} = 1 / \sigma_{ijk}$$

- $W_{Tijk} = \sum W_{ijk}$ ;
- Normalisierung:  $W_{ijk} = W_{ijk} / W_{Tijk}$ ;

# Gliederung

- Motivation
- Theoretische Ansätze
  - Wechselndes Anfrageobjekt
  - Verwendung von Klassifikatoren
  - Gewichtungsmethode
- Implementierung
- Ergebnisse
- Diskussion

# Implementierung

RFGui

query image    calculate    run    round

				
choose:	choose:	choose:	choose:	choose:
no opinion	no opinion	no opinion	no opinion	no opinion
similarity    image1	no opinion highly relevant relevant non-relevant highly non-relevant	similarity	similarity	similarity
				
choose:	choose:	choose:	choose:	choose:
no opinion	no opinion	no opinion	no opinion	no opinion
similarity	similarity	similarity	similarity	similarity

# Gliederung

- Motivation
- Theoretische Ansätze
  - Wechselndes Anfrageobjekt
  - Verwendung von Klassifikatoren
  - Gewichtungsmethode
- Implementierung
- Ergebnisse
- Diskussion

# Ergebnisse

RFGui

676 calculate run 0

				
choose:	choose:	choose:	choose:	choose:
highly relevant	highly relevant	highly relevant	highly non-relevant	highly non-relevant
0.0107713 676	0.3056 666	0.305714 675	0.321759 769	0.324598 773
				
choose:	choose:	choose:	choose:	choose:
highly relevant	highly relevant	highly non-relevant	highly relevant	highly relevant
0.328334 689	0.331134 670	0.331675 929	0.33882 686	0.340359 672

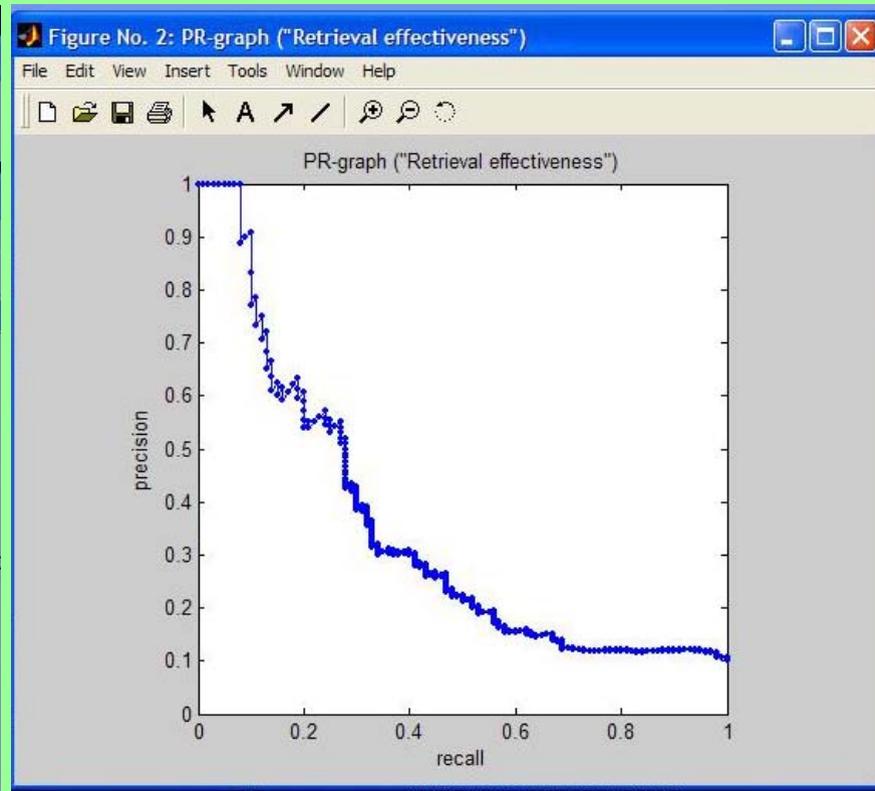
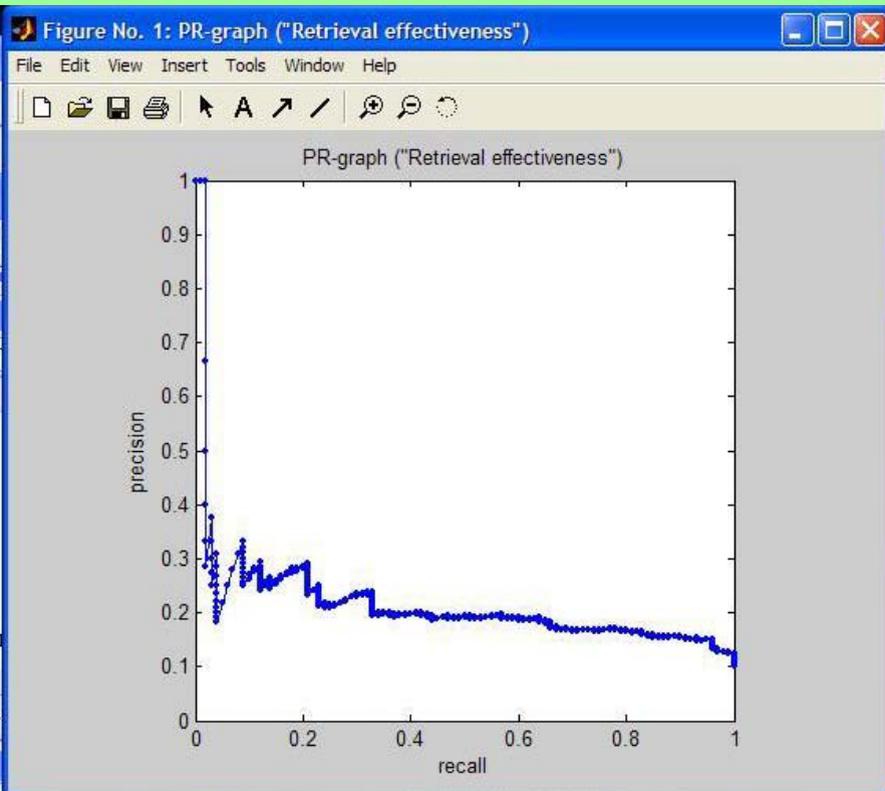
# Ergebnisse

RFGui

676 calculate run 1

				
choose: highly relevant	choose: highly relevant	choose: highly relevant	choose: highly relevant	choose: highly relevant
0.00121019   676	0.0963994   675	0.0989375   666	0.0994115   689	0.101291   699
				
choose: highly relevant	choose: highly relevant	choose: highly relevant	choose: highly non-relevant	choose: highly relevant
0.101814   697	0.102674   686	0.103182   683	0.103308   870	0.103656   673

# PR-Graphen



# Ergebnisse

RFGui

61 calculate run 0

				
choose: highly relevant	choose: highly relevant	choose: highly non-relevant	choose: highly non-relevant	choose: no opinion
0.157659 61	0.291405 47	0.306733 206	0.307494 929	0.309256 512
				
choose: no opinion	choose: no opinion	choose: highly relevant	choose: no opinion	choose: no opinion
0.316528 516	0.323581 526	0.324807 88	0.327383 531	0.328515 522

# Ergebnisse

RFGui

61 calculate run 3

				
choose: highly relevant	choose: highly relevant	choose: no opinion	choose: non-relevant	choose: no opinion
0.0568319   61	0.102358   47	0.106274   206	0.107885   929	0.109112   512
				
choose: no opinion	choose: no opinion	choose: highly relevant	choose: no opinion	choose: no opinion
0.111213   522	0.111264   526	0.111912   88	0.11221   212	0.113815   761

# Gliederung

- Motivation
- Theoretische Ansätze
  - Wechselndes Anfrageobjekt
  - Verwendung von Klassifikatoren
  - Gewichtungsmethode
- Implementierung
- Ergebnisse
- Diskussion

Vielen Dank für die  
Aufmerksamkeit!