

PCA based feature fusion

Seminar Inhaltsbasierte Bildsuche

WS 04/05

Übersicht

- Motivation: PCA an einem Beispiel
- PCA in der Bildsuche
- Tests
- Zusammenfassung / Ausblick
- Diskussion / Demo

Motivation: PCA

PCA = Principal Component Analysis

Samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$

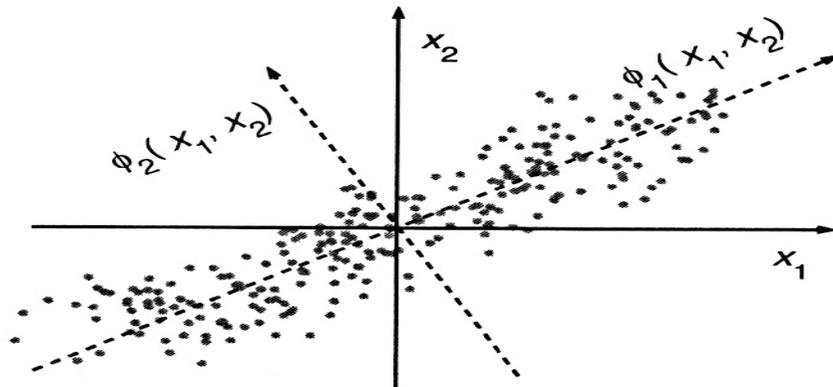
Gesucht: $\Phi^T : X \rightarrow Y; \mathbf{y} = \Phi^T \mathbf{x}$

sodass $\Sigma_y = \Lambda = \Phi^T \Sigma_x \Phi$

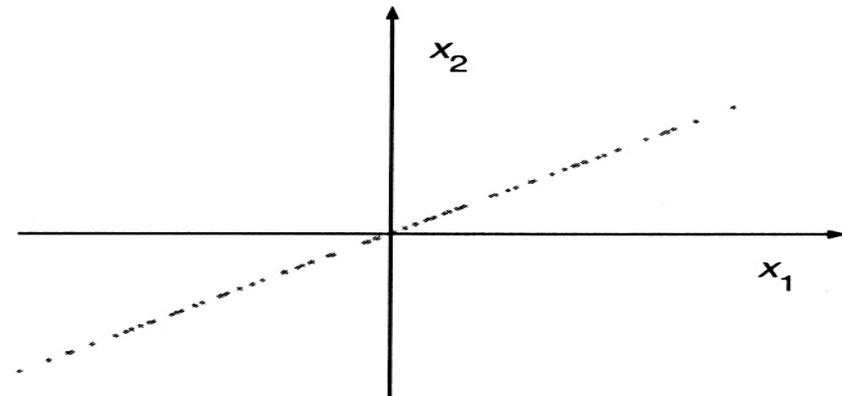
wobei Σ_y Diagonalmatrix

Φ_k : Matrix mit k ersten EV

$\mathbf{y} = \Phi_k^T \mathbf{x}$ Projektion auf Unterraum



(a) PCA basis



(b) PCA reduction to 1D

Motivation: Face Recognition



PCA
→

Eigenvektoren (Eigenfaces) zu
7 max. Eigenwerten



Turk, M., and Pentland, A., Eigenfaces for Face Recognition,
Journal of Cognitive Neuroscience, 3(1),pp.71-86, 1991

Motivation: Face Recognition

Projektion auf Facespace (EV der 7 max. EW)

Original



Projektion

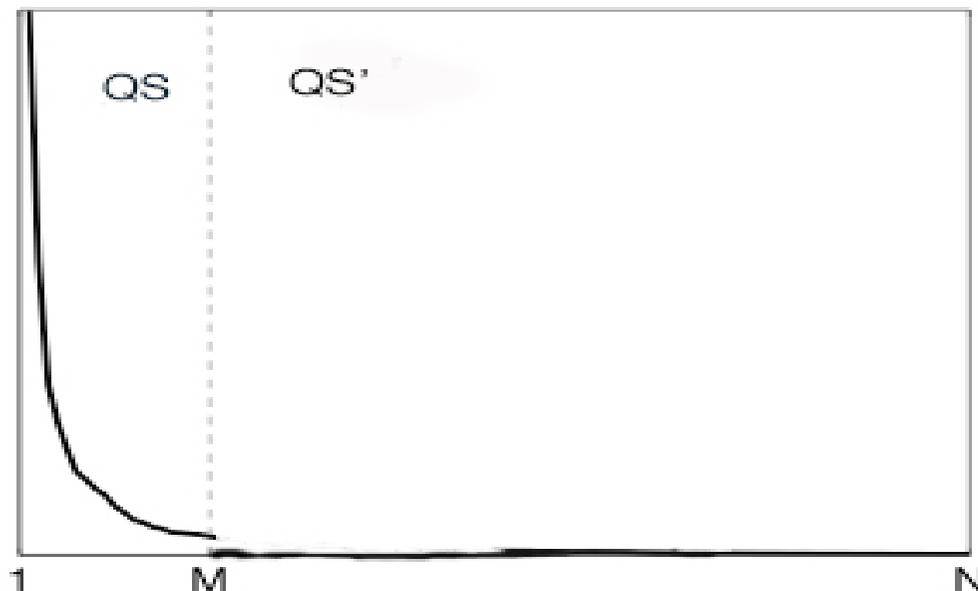


PCA in der Bildsuche

- Faces \Rightarrow Features
- Facespace \Rightarrow Query Space
- Klassifizierung \Rightarrow 1 Klasse relevanter Bilder
- Gesucht: Ähnlichkeit zu Query-Bildern
- Fusion ist eine Art von Relevance Feedback
- Relevante Features sollen durch PCA *automatisch* bestimmt werden

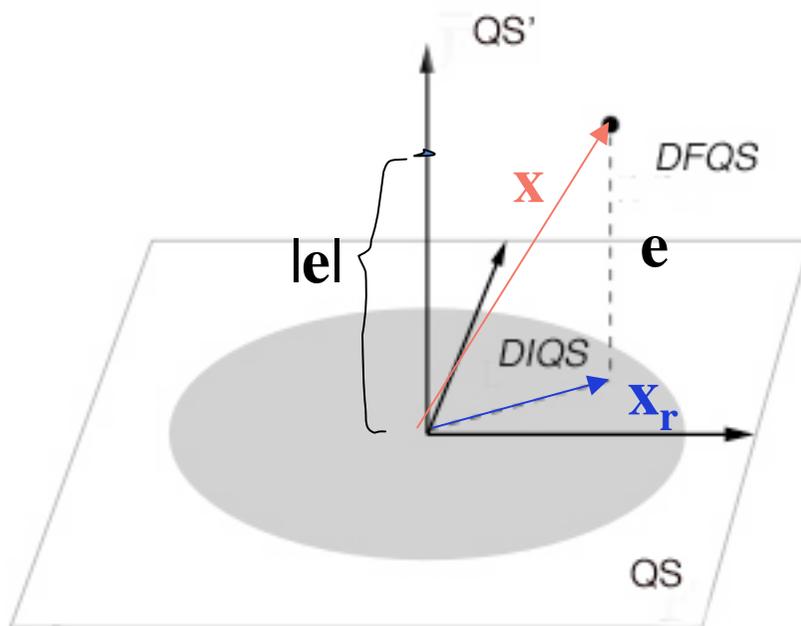
PCA in der Bildsuche

- Sei $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_M\}$ Menge von Query-Bildern (User)
- $\dim(\mathbf{q}_i) = N$, $M \ll N$
- OBdA: Mittelpunkt der \mathbf{q}_i im Nullpunkt
- Sei Φ die Matrix der Eigenvektoren zur Kovarianzmatrix Σ_q
- $\dim(\Phi) = N \times M$ (EV mit EW = 0 werden nicht benutzt)



Projektion auf Nullraum

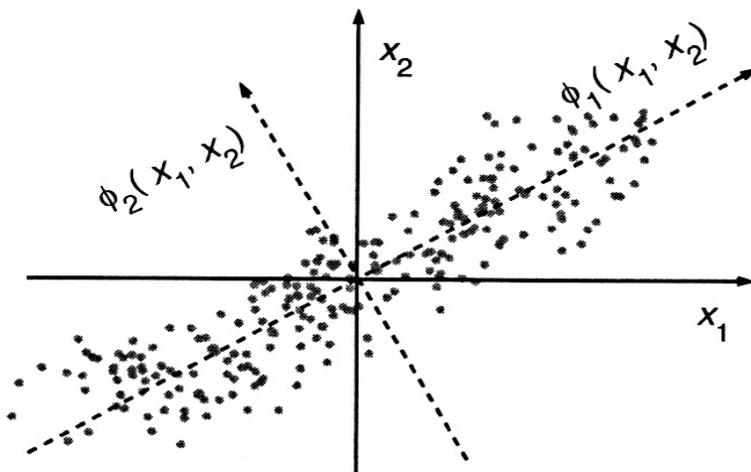
QS' : Nullraum der Abbildung Φ^T



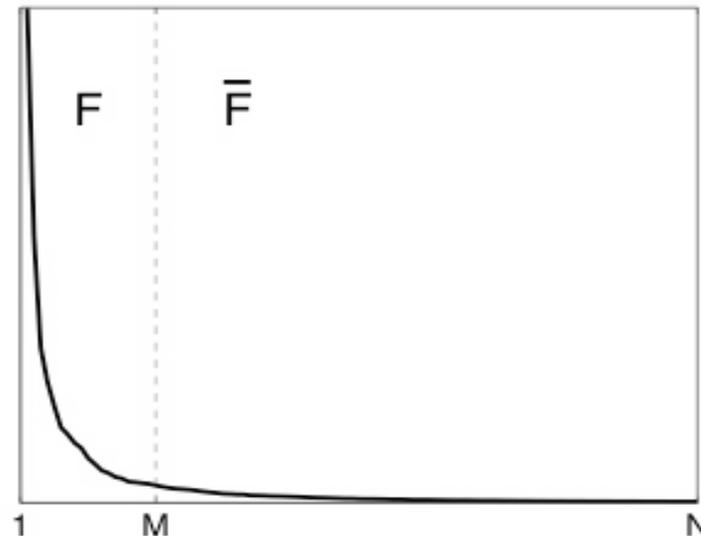
- $\mathbf{y} = \Phi^T \mathbf{x}$ Projektion von \mathbf{x} auf QS
- $\mathbf{x}_r = \Phi \mathbf{y}$ Rekonstruktion
- $|\mathbf{e}| = |\mathbf{x} - \mathbf{x}_r|$ Reconstruction Error
- \mathbf{e} entspricht Projektion auf Nullraum QS'

Warum Nullraum?

- Gesucht: Ähnlichkeit zwischen den Query-Bildern
- Genauer: Feature, das diese Ähnlichkeit bewirkt
- Korrekt: Unterraum, der die Ähnlichkeit beschreibt
- Projektion auf Nullraum entspricht Nutzung der Features, die der User (unbewusst) als Ursache der Ähnlichkeit empfindet



(a) PCA basis



Kombination der Features

- Vor der Fusion: verschiedene Features zu einem langen Vektor kombinieren (seriell oder parallel)
- Hier: Features seriell aneinandergesetzt
- 4 Methoden wurden untersucht:

$$\begin{aligned} \text{raw: } f &= [f_1; f_2; \dots; f_n] \\ \text{norm: } f &= \left[\frac{f_1}{|f_1|}; \frac{f_2}{|f_2|}; \dots; \frac{f_n}{|f_n|} \right] \\ \text{range: } f &= \left[\frac{f_1}{\max_1}; \frac{f_2}{\max_2}; \dots; \frac{f_n}{\max_n} \right] \\ \text{sum: } f &= \left[\frac{f_1}{\text{sum}(f_1)}; \frac{f_2}{\text{sum}(f_2)}; \dots; \frac{f_n}{\text{sum}(f_n)} \right] \end{aligned}$$

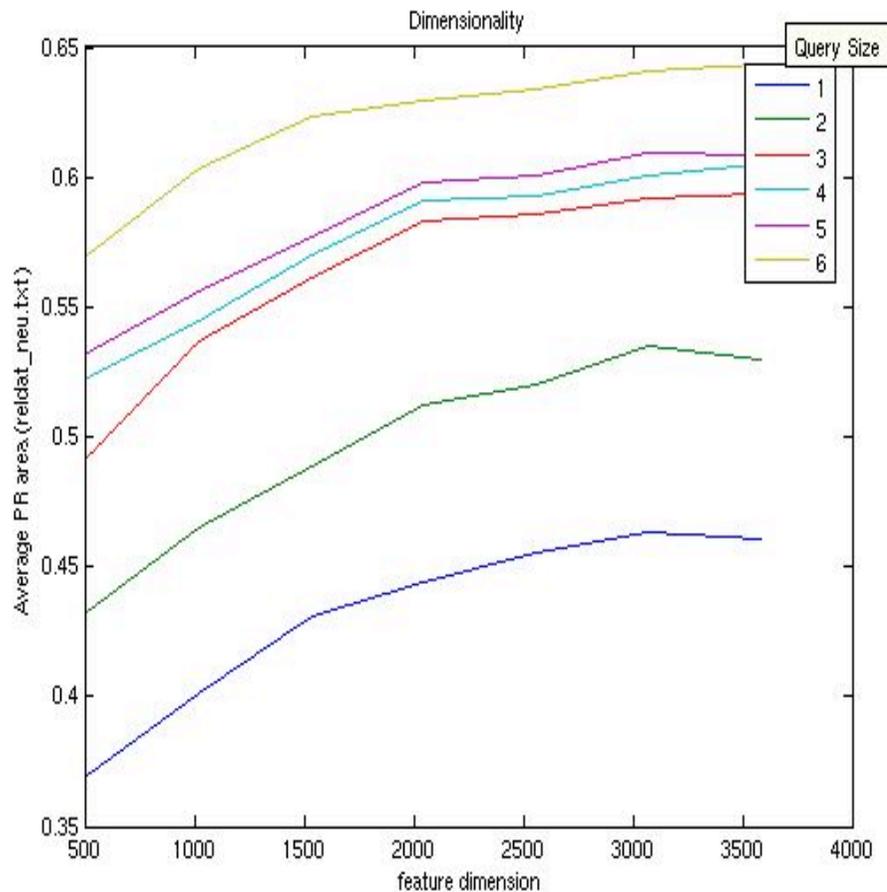
Algorithmus

1. Features zu Super-Vektor kombinieren
2. User bestimmt ähnliche Bilder Q
3. Translation um Mittelwert der q_i
4. Berechnung des QS \Rightarrow Abb. Φ
5. Projektion aller Feature auf QS
6. Berechnung der DFQS für jedes Bild
7. Sortierung nach DFQS

Tests

- Tuning:
 - ◆ Dimensionalität des Featurevektors
 - ◆ Kombination der Features
 - ◆ Anzahl der Anfragebilder
- Vergleich mit anderen Methoden:
 - ◆ Fusion vs. Single Feature
 - ◆ (Fusion vs. Relevance Feedback)

Tests: Dimensionalität

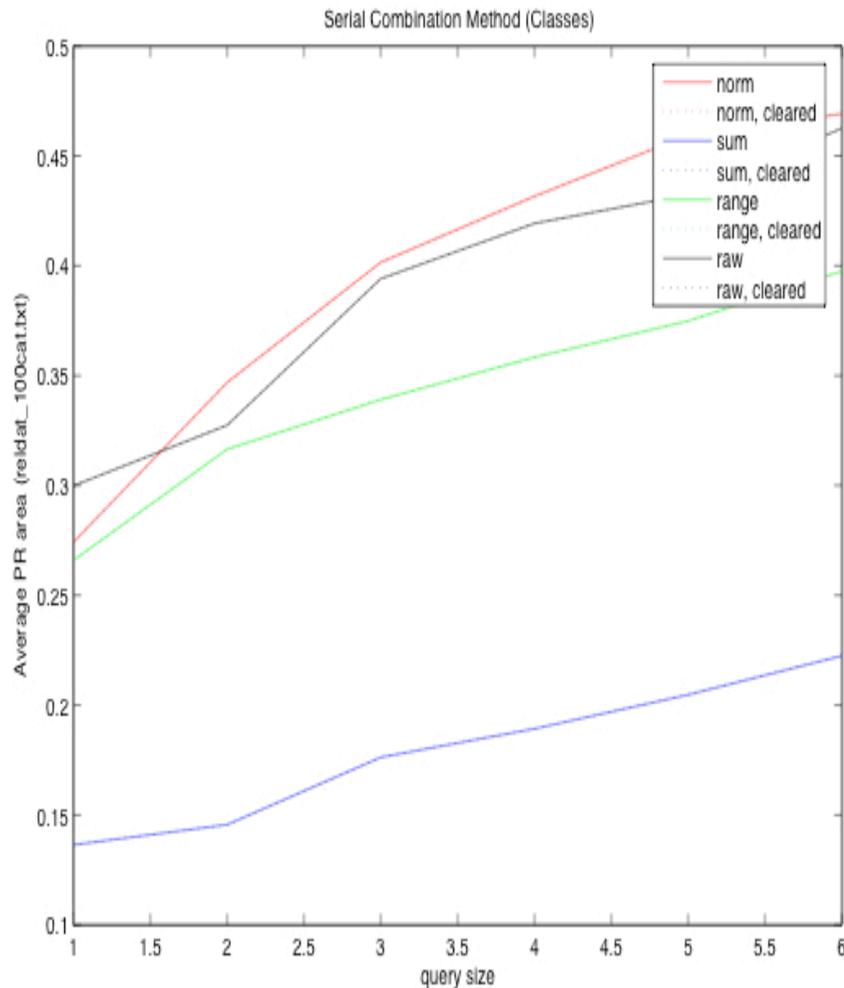


- 3 zufällige Sequenzen von 7 Features f_1, \dots, f_7
- $\dim(f_i) = 512 \quad \forall i=1 \dots 7$
- Länge der Features F: 512, 1024, 1536, 2048, 2560, 3072, 3584
- Berechnung des *average PR-Graphen* für alle Längen und alle Sequenzen
- Mittelung über Sequenzen
- Auswertung für verschiedene Größen der Anfrage Q_n

Erkenntnisse:

1. PCA optimal $\dim(F) \approx 3000$
2. GT `reldat_neu.txt` ungeeignet für $Q_n > 1$

Tests: Feature Kombination



- Seminar Features (dim ≈ 1000)
- sehr unterschiedliche Dimensionen und Wertebereiche
- 4 verschiedene Methoden
- cleared: Dimensionen mit Standardabweichung = 0 gelöscht
- durchschnittliche Fläche des PR-Graphen (17 Bilder)

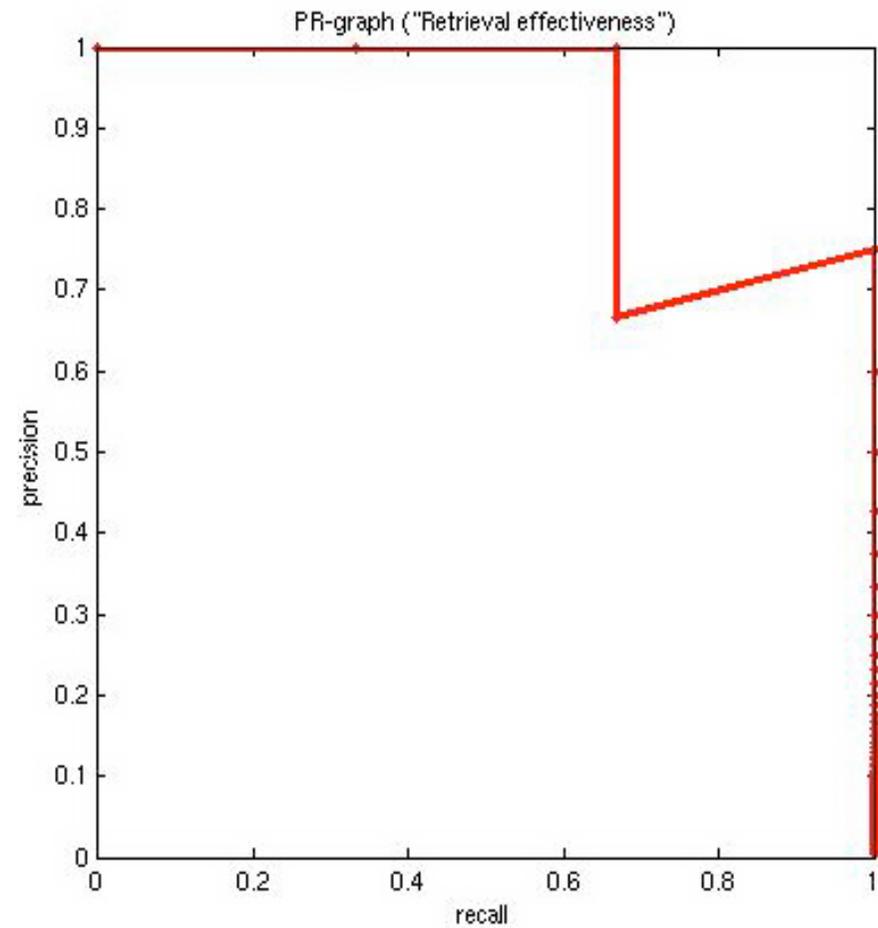
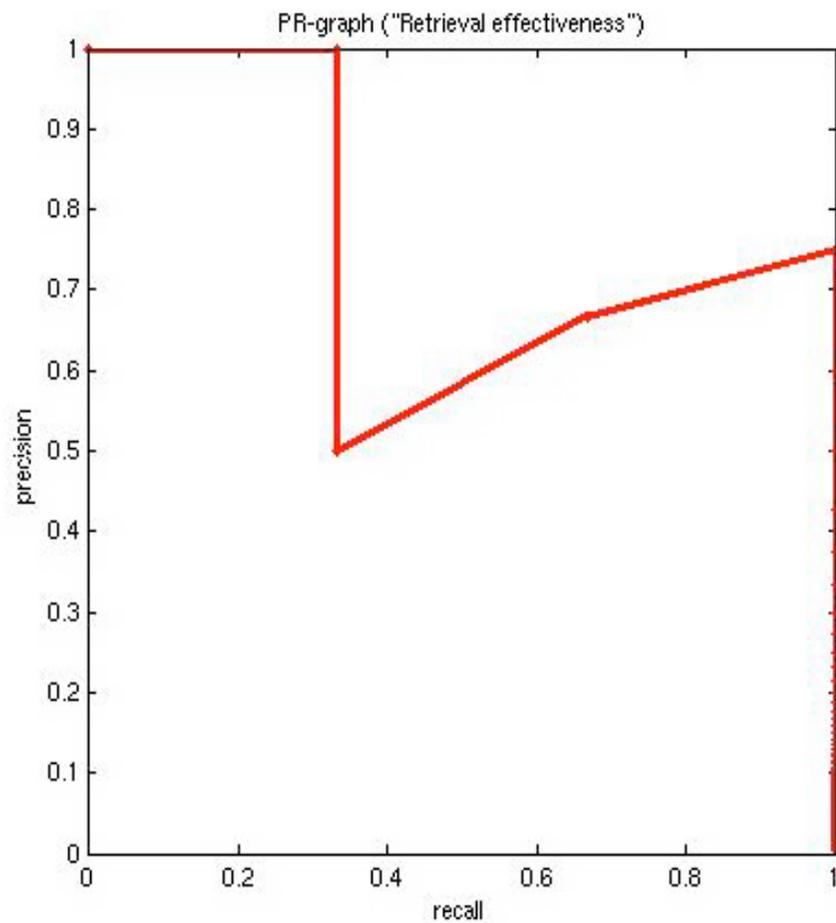
Erkenntnisse:

1. einfaches serielles Zusammensetzen genügt, Normierung übersichtlicher
2. Löschen von informationslosen Dimensionen bringt keine bessere Qualität, aber schlechte Skalierbarkeit

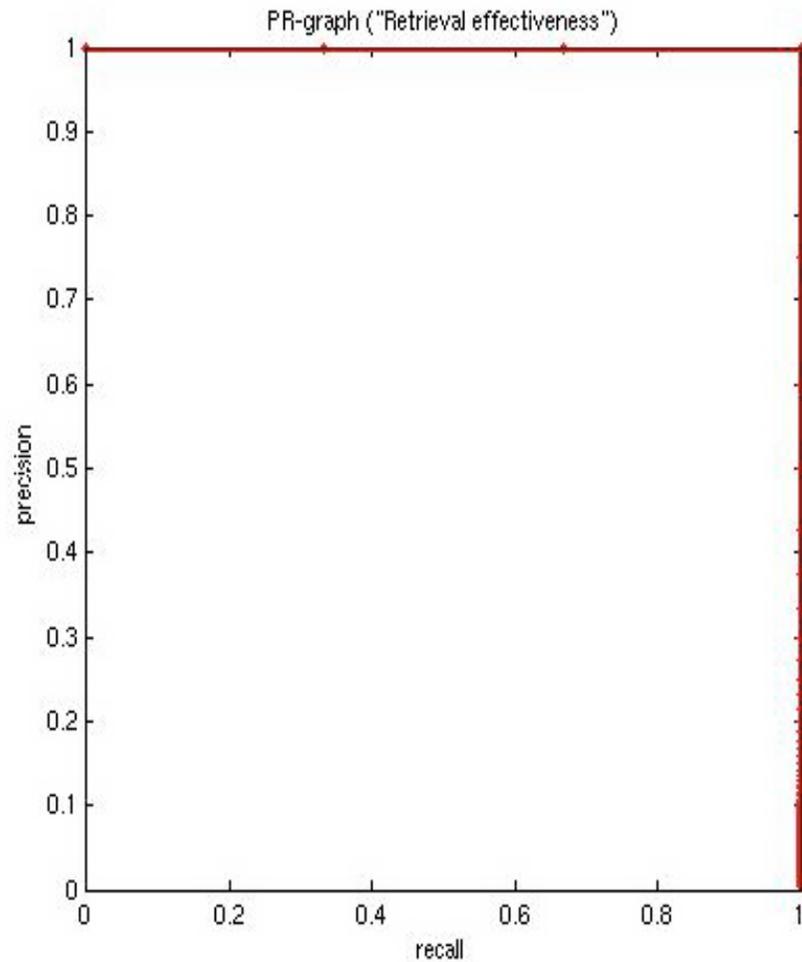
Tests: Query Size

- Anfrage: Alle 17 Bilder in `reldat_neu.txt`
- Pro Bild 20 zufällige Bilder aus gleicher Kategorie
- Berechnung der durchschnittlichen PR-Graph Fläche in Abhängigkeit der Anfragegröße
- Auswertung für 3 verschiedene Feature DBs
- Probleme:
 - ◆ Bilder innerhalb einer Kategorie nicht unbedingt ähnlich
 - ◆ Nur `reldat_100cat.txt` als Ground Truths sinnvoll
 - ◆ PR-Graph wird durch größere Anfragen verbessert obwohl gefundene Bilder nicht unbedingt ähnlich sind

Problem mit kleinen Klassen (1)

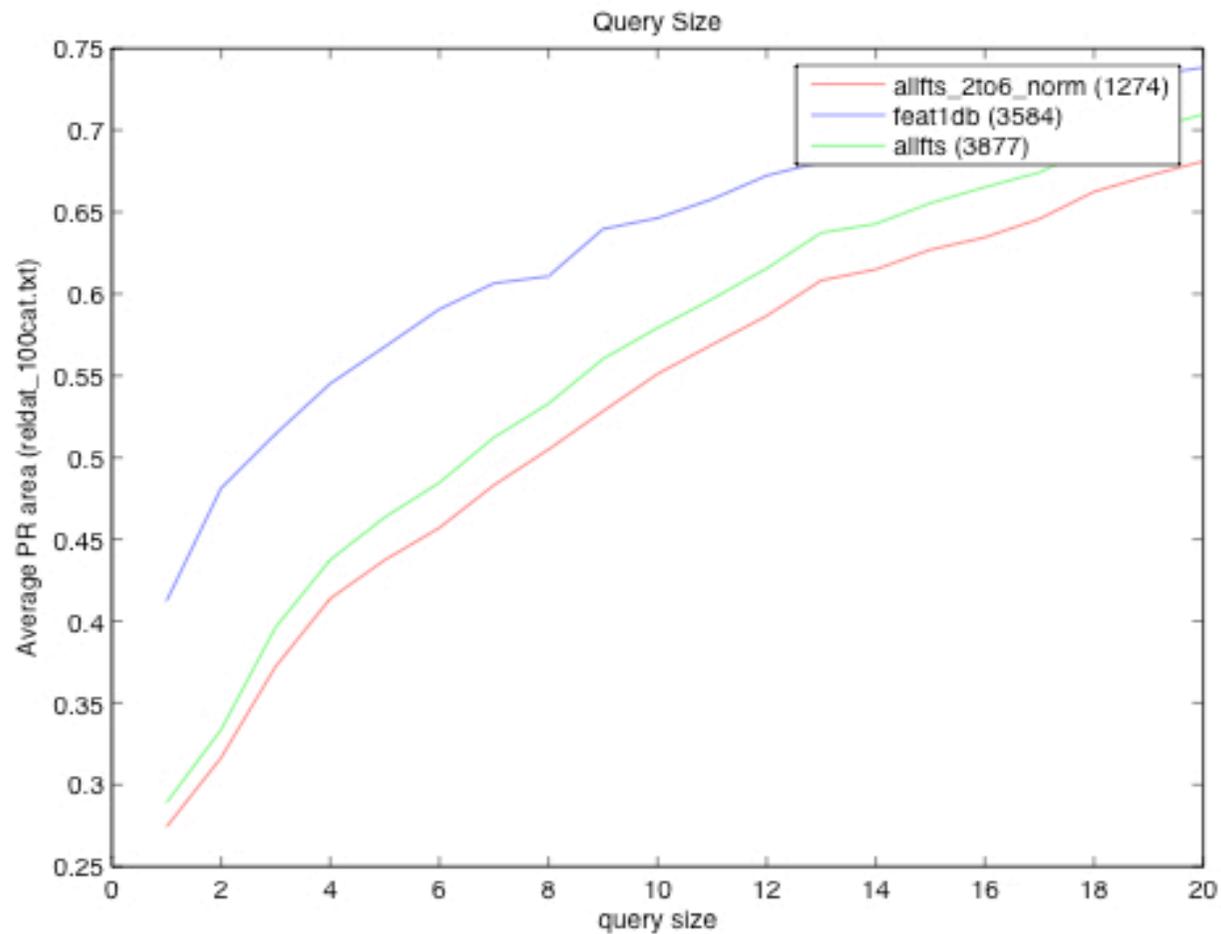


Problem mit kleinen Klassen (2)

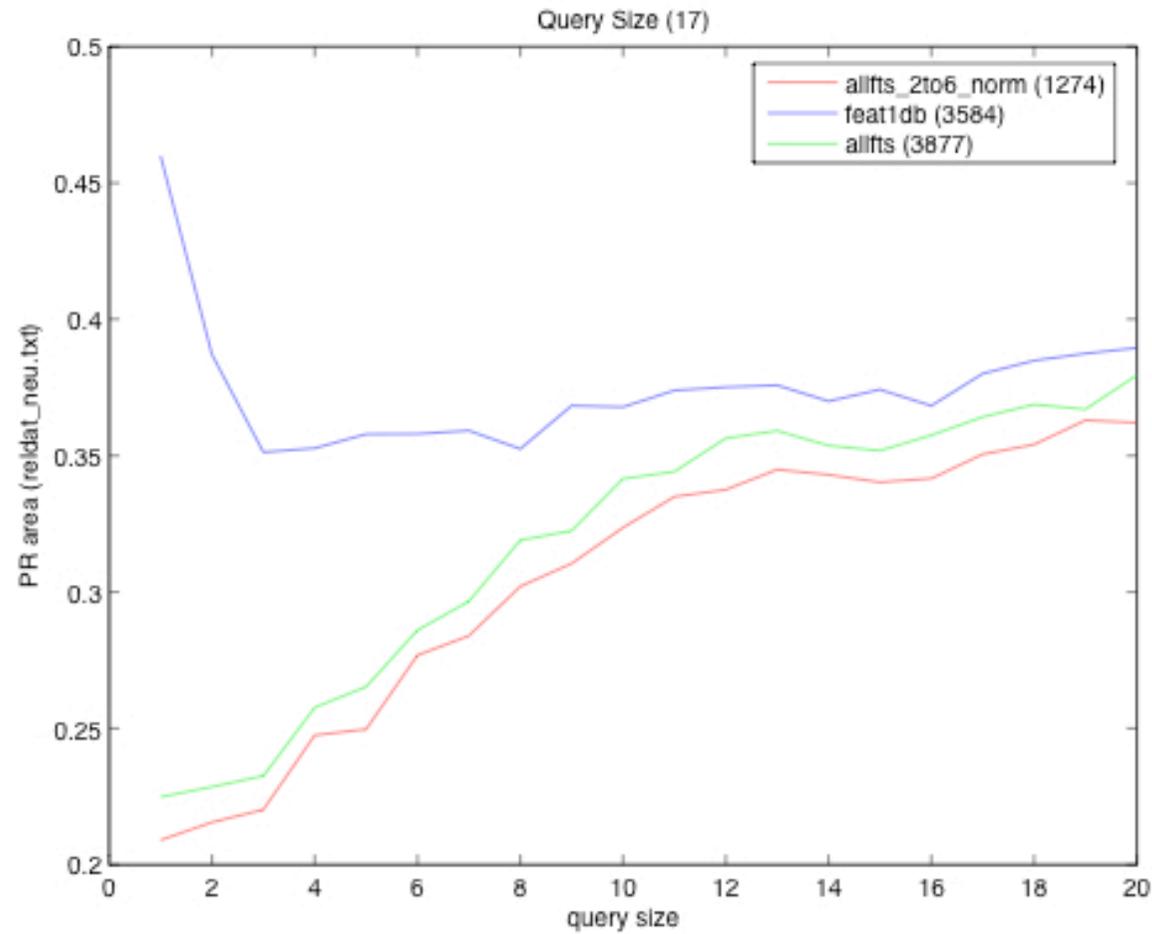


31.jpg 1 6.2714e-12	88.jpg 2 8.6825e-12	51.jpg 3 1.8259e-11
260.jpg 4 16138.0115	269.jpg 5 24752.0648	212.jpg 6 25604.1495
84.jpg 7 26664.7743	160.jpg 8 26829.1889	240.jpg 9 26900.3735
<input type="button" value="previous"/>		<input type="button" value="next"/>

Tests: Query Size (1)



Tests: Query Size (2)



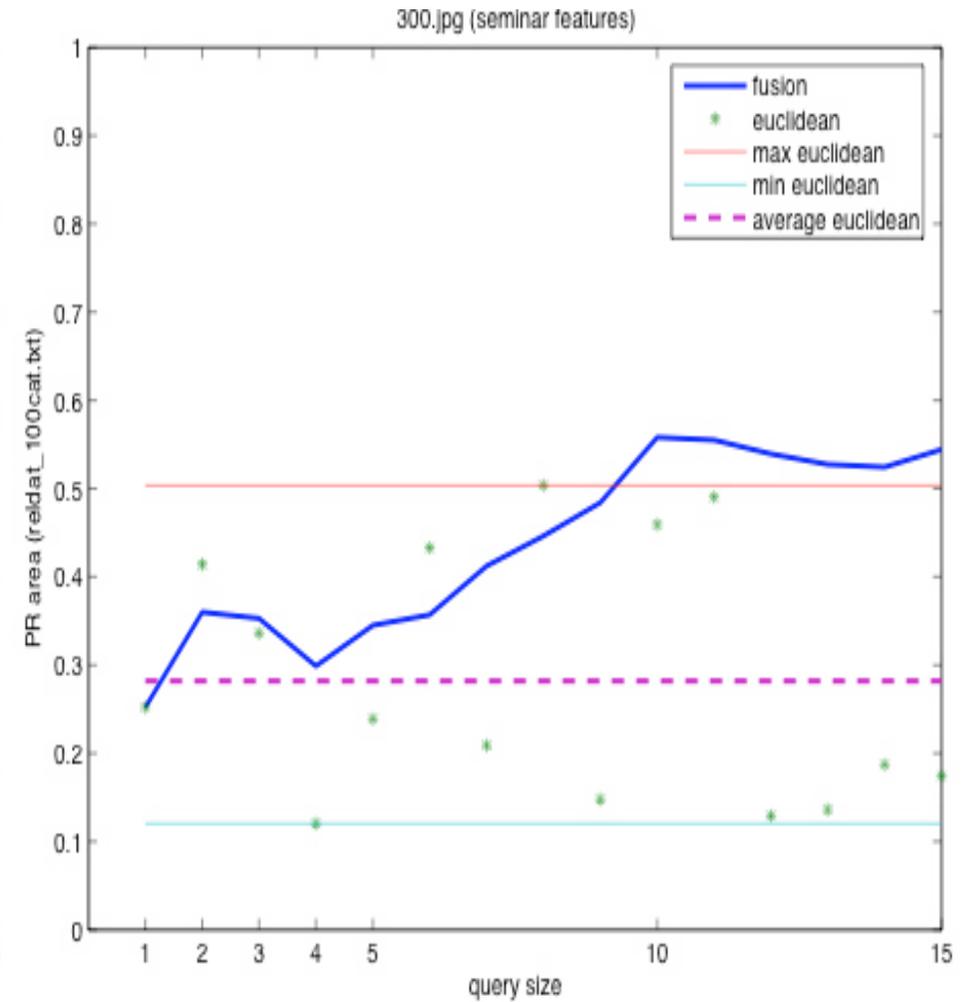
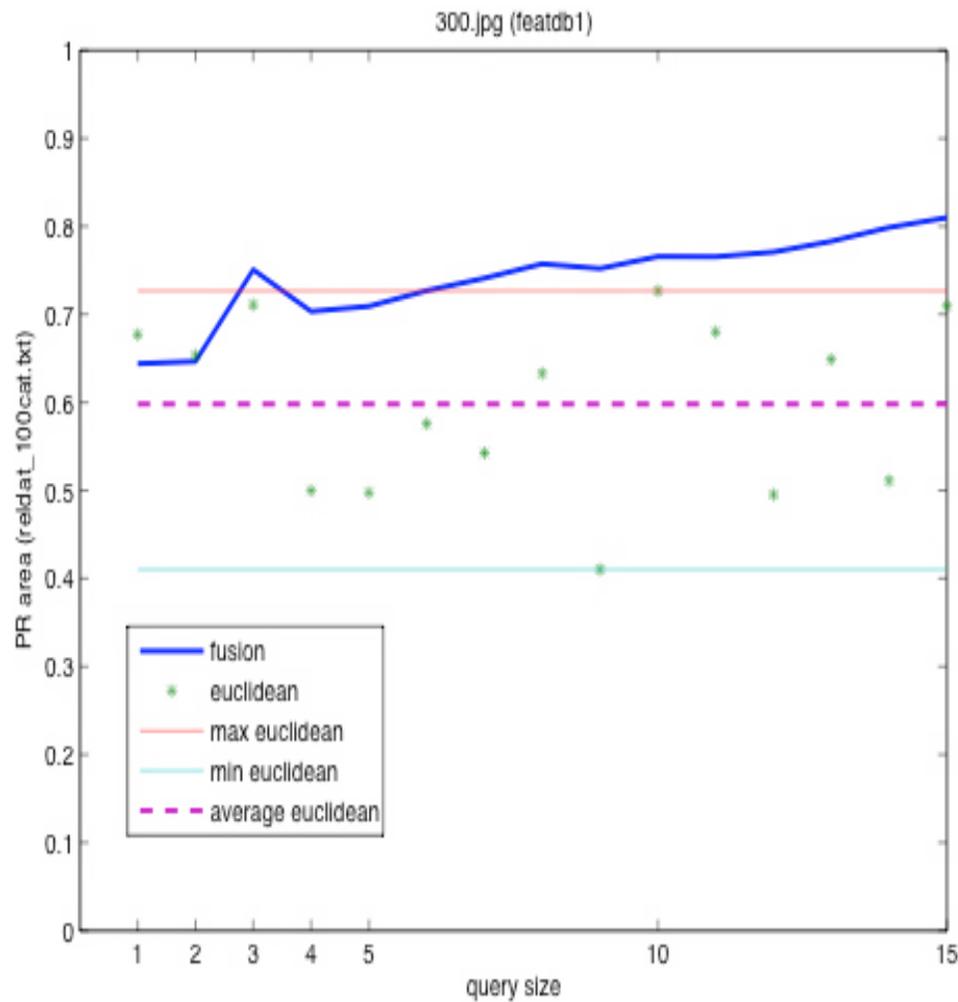
Tests: Query Size (3)

- Ergebnis:
 - ◆ Größere Anfragen führen zu besseren Suchergebnissen
 - ◆ ab ca. 10 Bildern scheint Zuwachs nur noch durch die Größe (nicht durch Qualität) bestimmt
 - ◆ Man muss sich Gedanken über die Ground Truths machen
 - ◆ Für quantitative Aussagen benötigt man von Anfragegröße unabhängige Evaluationsmethode

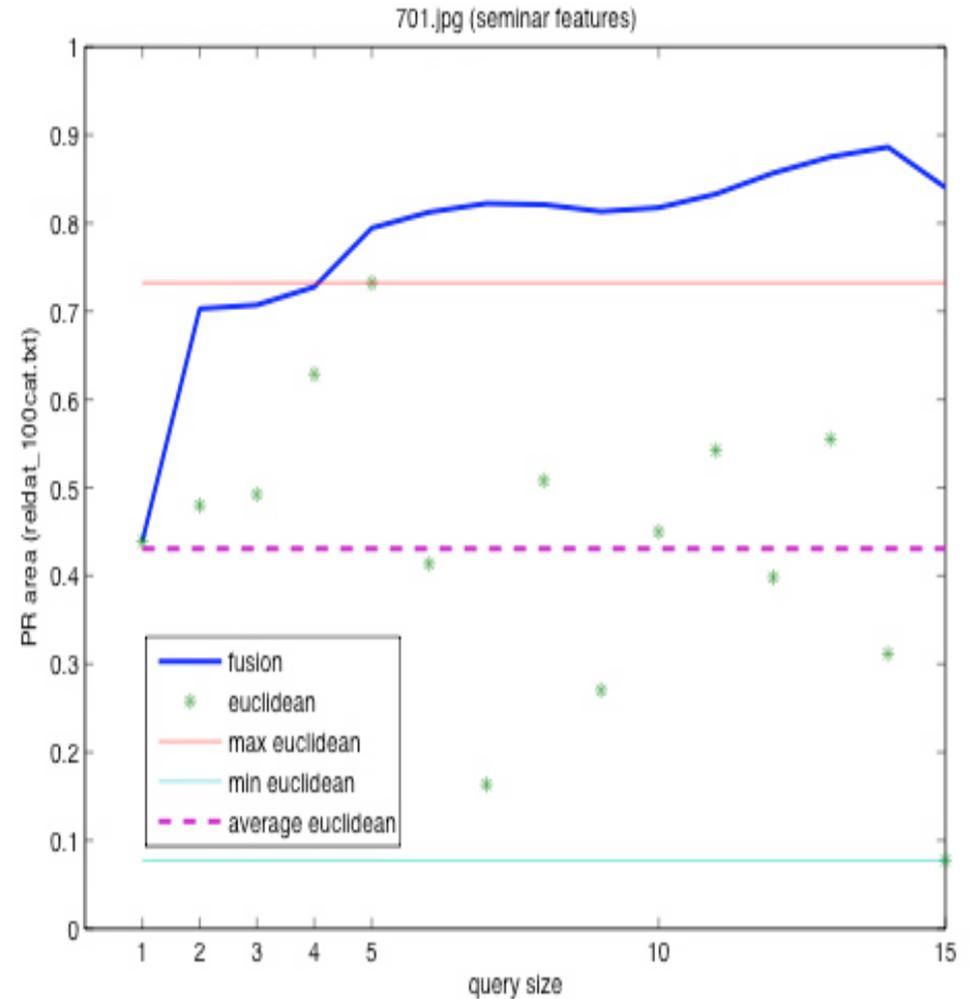
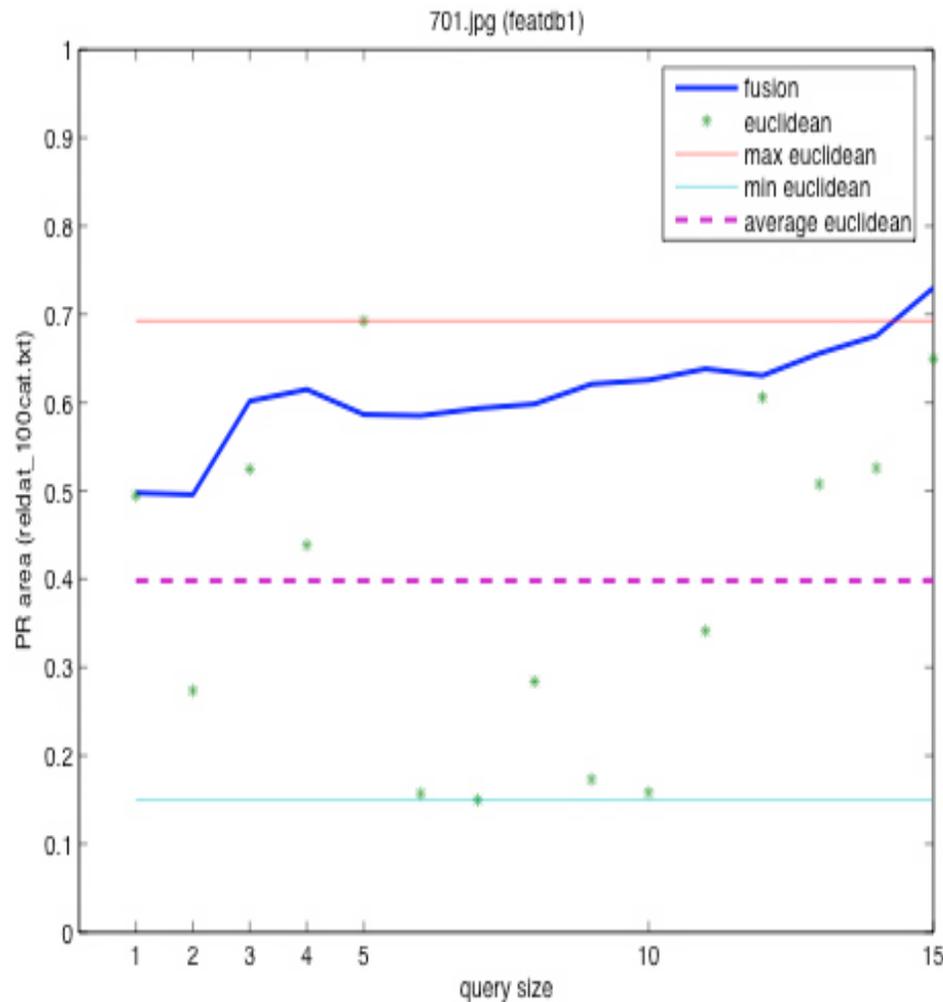
Tests: Fusion vs. Single Feature

- Vergleich zwischen Fusion und kombinierten Featurevektor (Distanz im Feature-Raum)
- 2 Bilder (300.jpg, 701.jpg) aus großen Klassen
- Variieren der Anfragegröße von 1 bis 15 bei der Fusion
- Suche mit euklidischen Abstand für die einzelnen Bilder der Anfrage
- Auswertung für 2 Feature DBs

Tests: Fusion vs. Single Feature (1)



Tests: Fusion vs. Single Feature (2)



Tests: Fusion vs. Single Feature (3)

- Ergebnis:
 - ◆ Fusion bringt Verbesserung gegenüber Distanzmessung im Feature-Raum
 - ◆ Die ersten 2-4 zusätzlichen Anfragebilder scheinen größte Qualitätssteigerung zu bringen

Zusammenfassung

- Fusion besser als Distanzmessung im Feature-Raum
- Voraussetzung: Gute Features
- Geeignete Kombinationsmethoden:
 - ◆ Längennormierung
 - ◆ Einfach hintereinander hängen
- Dimension des Featurevektors hat Einfluss auf Suchergebnis

Ausblick

- Quantitative Aussagen:
 - ◆ geeigneter Evaluationsmethode
 - ◆ größere Bild-DB (mehr relevante Bilder/Testbild)
- Bringen unterschiedlichere Typen von Features bessere Ergebnisse?
- Optimum für $|\text{Featurevektor}|/\text{Anfragegröße}$?
- Bei größeren Anfragen weniger EV benutzen?
- Parallele Kombination nutzbar?
- Dadurch Verbesserung der Laufzeit?

Diskussion / Demo