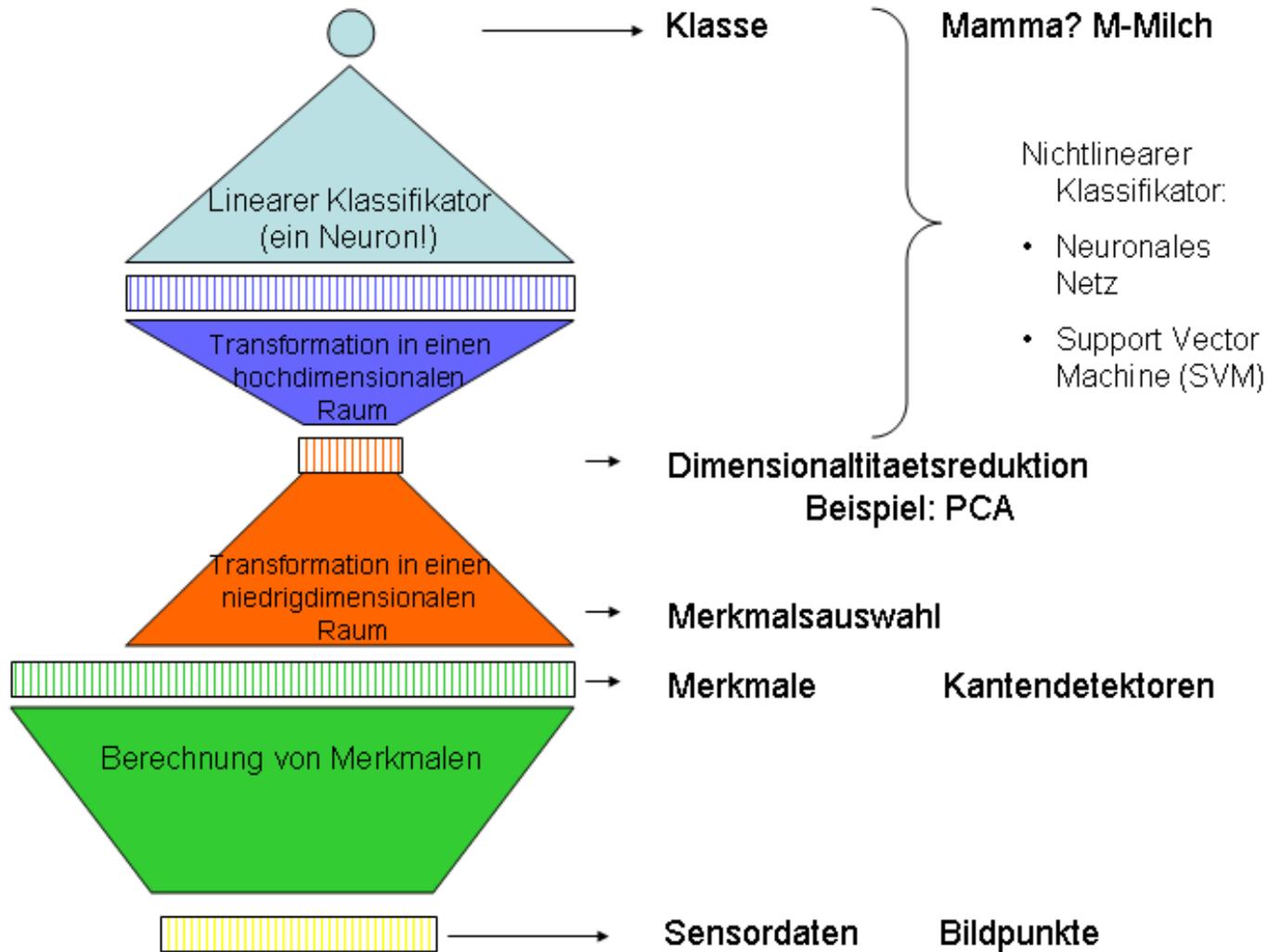


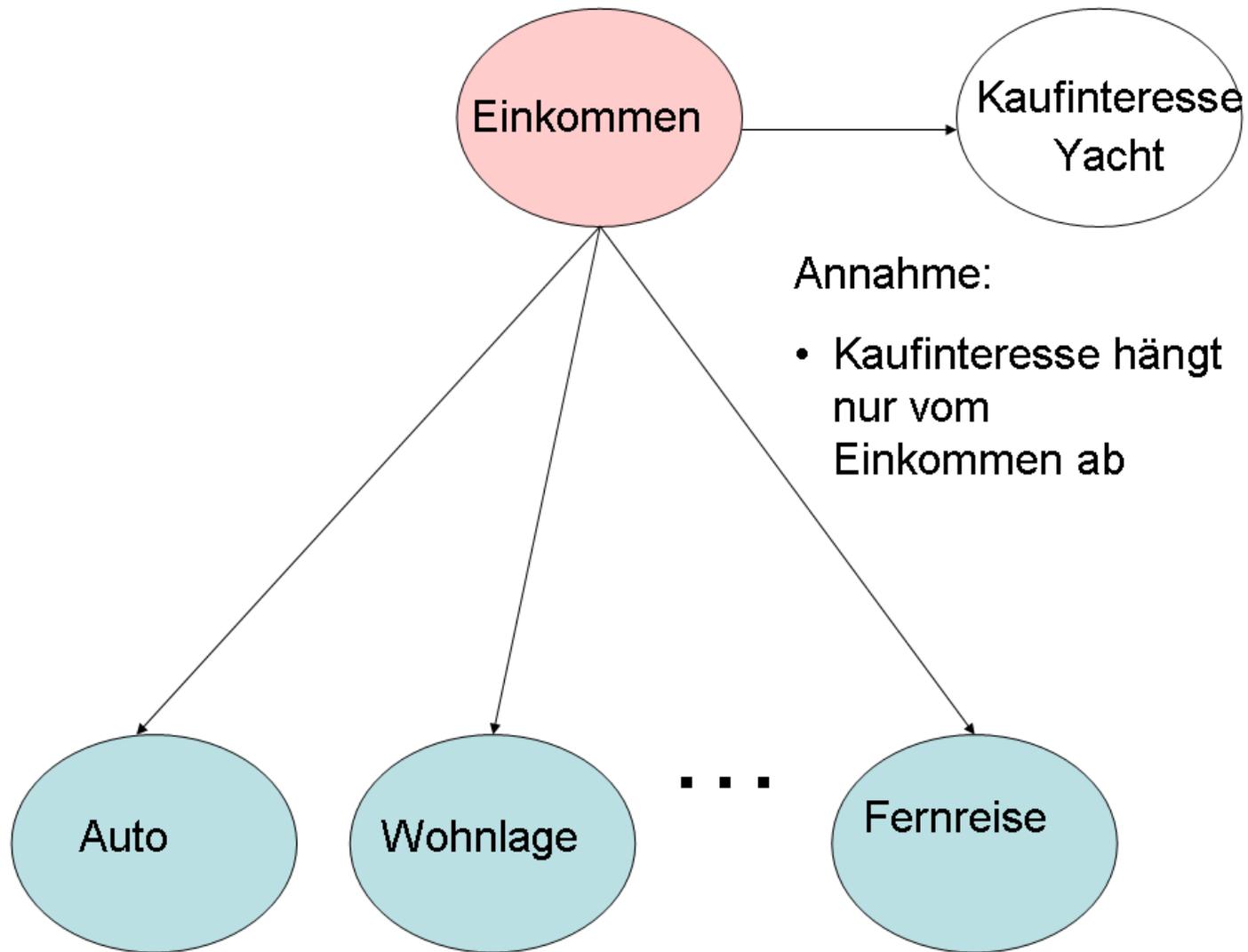
# Anwendungen der Hauptkomponentenanalyse

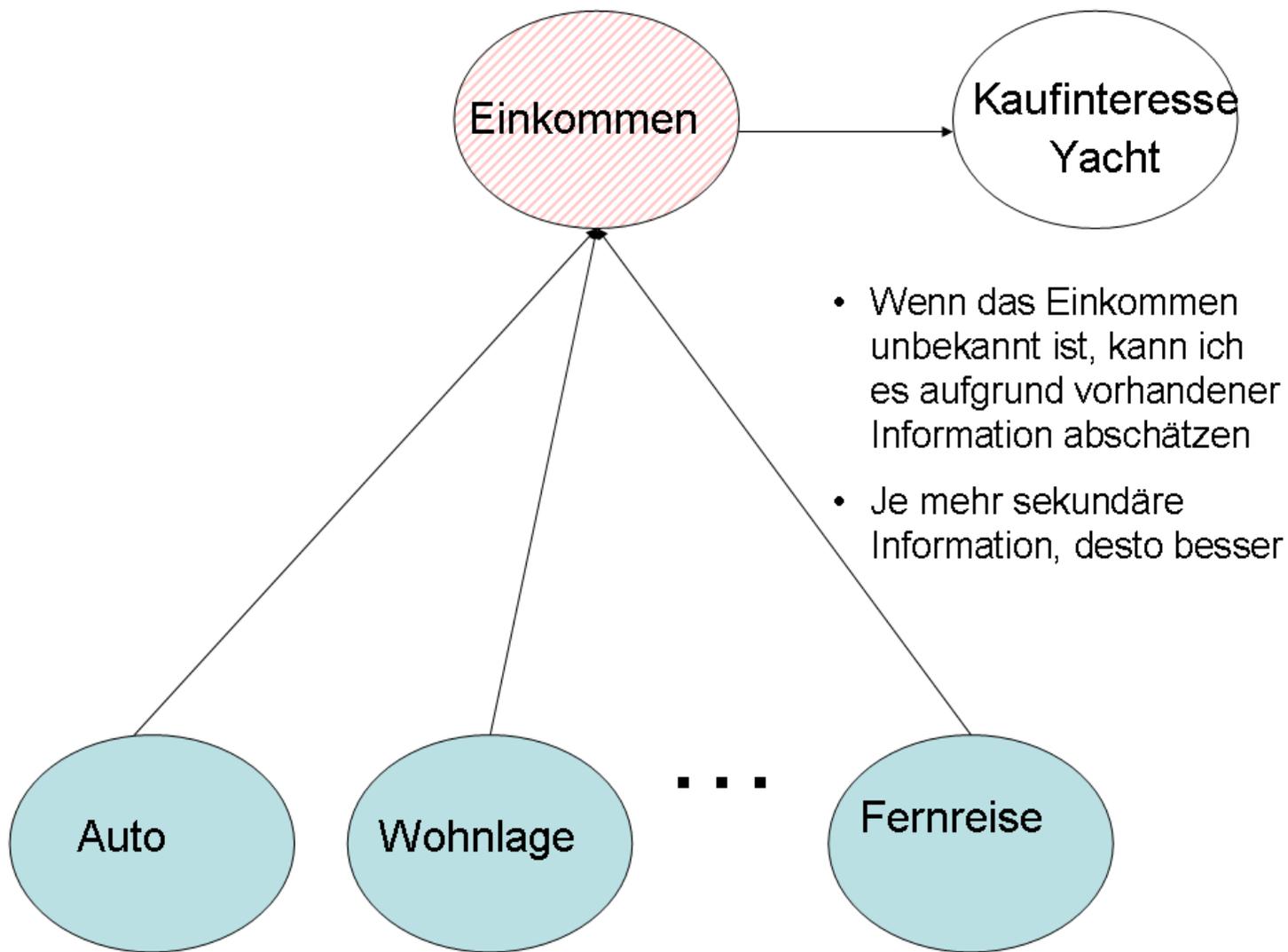
Volker Tresp  
vertreten durch Florian Steinke

# Dimensionsreduktion für Supervised Learning



## Beispiel: Kaufentscheidung

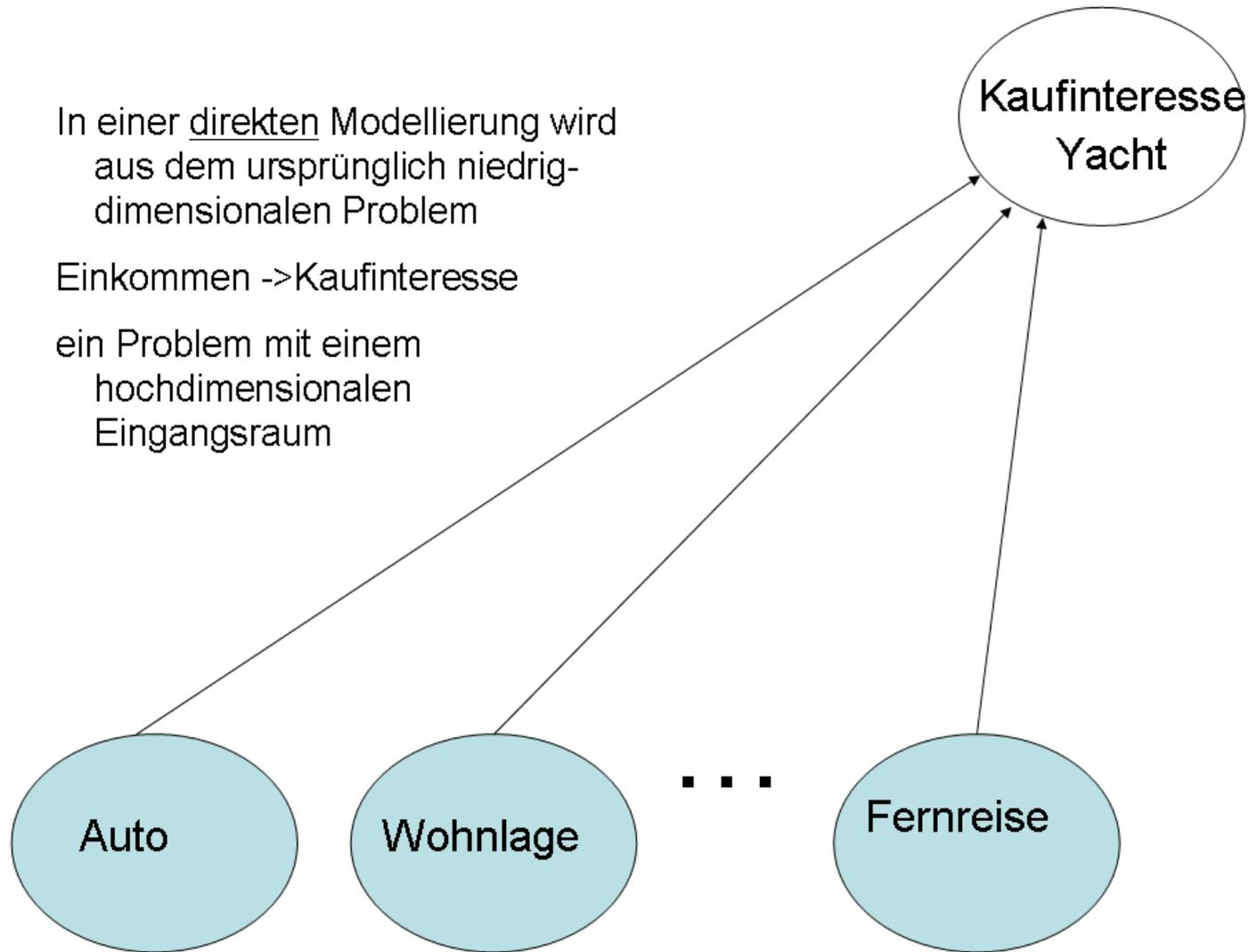


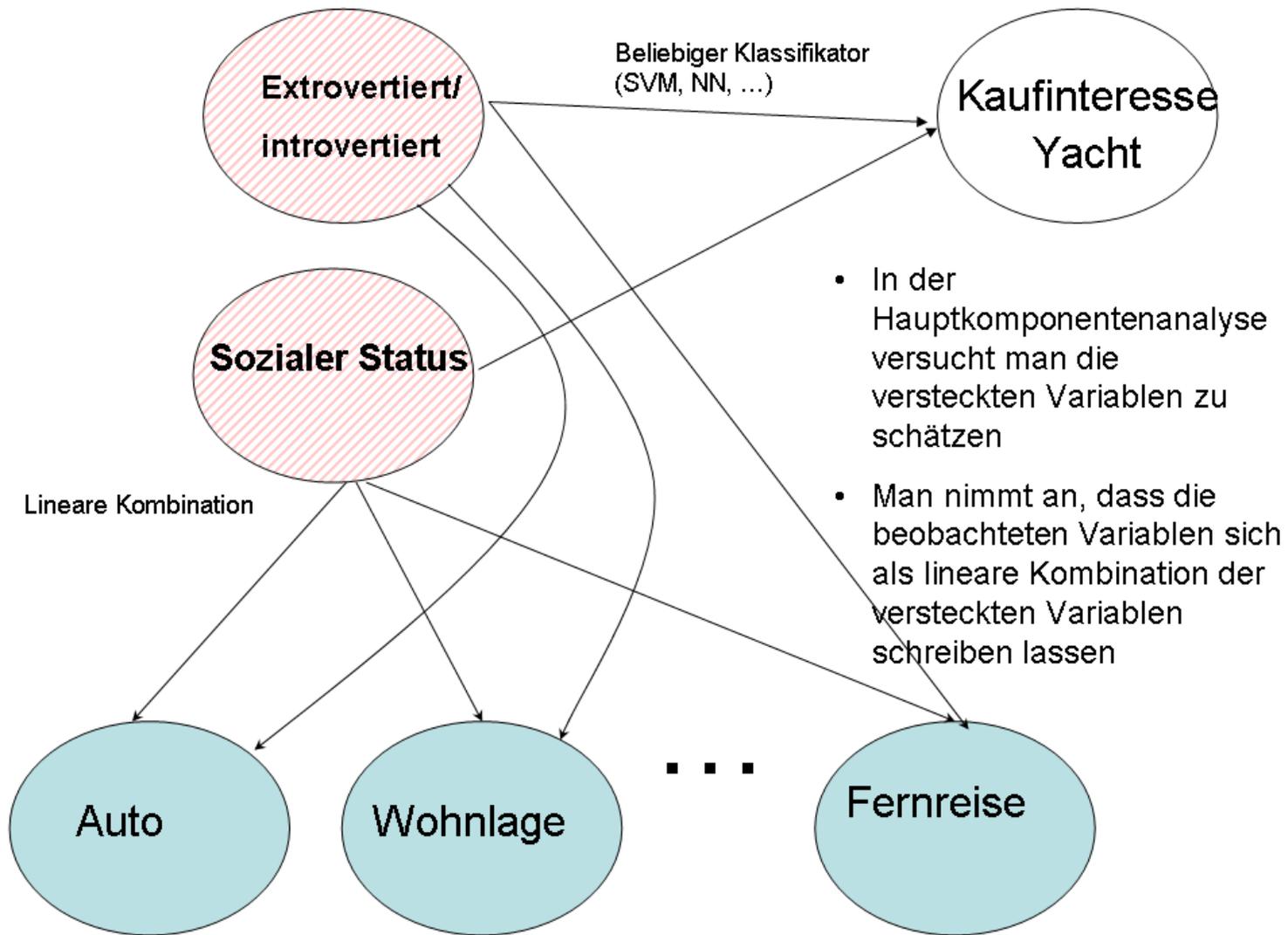


In einer direkten Modellierung wird  
aus dem ursprünglich niedrig-  
dimensionalen Problem

Einkommen -> Kaufinteresse

ein Problem mit einem  
hochdimensionalen  
Eingangsraum





# Verbesserte Abstandsmaße durch Hauptkomponentenanalyse

- Der euklidische Abstand eines Testvektors  $\mathbf{z}$  zu einem Datenvektor  $\mathbf{x}_i$  ist

$$dist_{Euklid} = \sum_{j=1}^n (z_j - x_{j,i})^2$$

- Ein bedeutungsvolleres Abstandsmaß ergibt sich durch die Hauptkomponenten

$$dist_{Euklid,PCA} = \sum_{j=1}^r (w(z)_j - w_{j,i})^2$$

wobei  $w_{j,i} = \mathbf{u}_j^T \mathbf{x}_i$     $w(z)_j = \mathbf{u}_j^T \mathbf{z}$

- Man kann den Abstand auch in Bezug auf den rekonstruierten Vektor berechnen

$$dist_{Euklid,rec} = \sum_{j=1}^n (\hat{z}_j - \hat{x}_{j,i})^2$$

wobei  $\hat{x}_{j,i} = \sum_{j=1}^r u_{j,i} w_{j,i}$     $\hat{z}_j = \sum_{j=1}^r u_{j,i} w(z)_j$

- Mahalanobis Distanz:

$$dist_{Mahalanobis} = \sum_{j=1}^r \left( \frac{w(z)_j - w_{j,i}}{\sigma_j} \right)^2$$

- Wie man leicht zeigen kann

$$dist_{Euklid,PCA} = dist_{Euklid,rec} \neq dist_{Euklid} \neq dist_{Mahalanobis}$$

Beweis:

$$\begin{aligned} dist_{Euklid,rec} &= (U_r \mathbf{w}(z) - U_r \mathbf{w}_i)^T (U_r \mathbf{w}(z) - U_r \mathbf{w}_i) \\ &= (\mathbf{w}(z) - \mathbf{w}_i)^T U_r^T U_r (\mathbf{w}(z) - \mathbf{w}_i) = (\mathbf{w}(z) - \mathbf{w}_i)^T (\mathbf{w}(z) - \mathbf{w}_i) \\ &= dist_{Euklid,PCA} \end{aligned}$$

## Anomaliedetektion durch Hauptkomponentenanalyse

- Angenommen, ich habe einen “normalen” Datensatz mit der Hauptkomponentenanalyse analysiert
- Stammt  $\mathbf{z}$  aus der gleichen Klasse von “normalen” Datenvektoren, sollte der Rekonstruktionsfehler

$$DFFS(\mathbf{z}, \hat{\mathbf{z}}) = \sum_{j=1}^M (\hat{z}_j - z_j)^2$$

(DFFS = distance from feature space) klein sein.

- Ist jedoch  $DFFS(\mathbf{z}, \hat{\mathbf{z}})$  gross, so ist  $\mathbf{z}$  “anomal”, oder “novel”

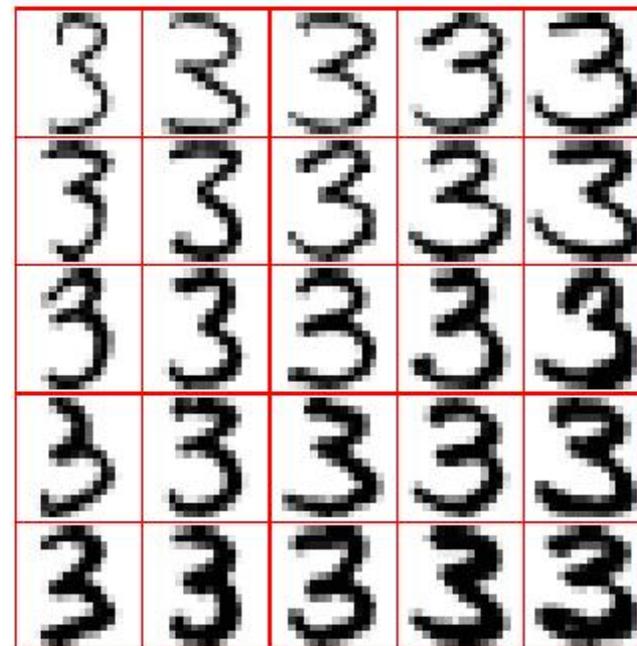
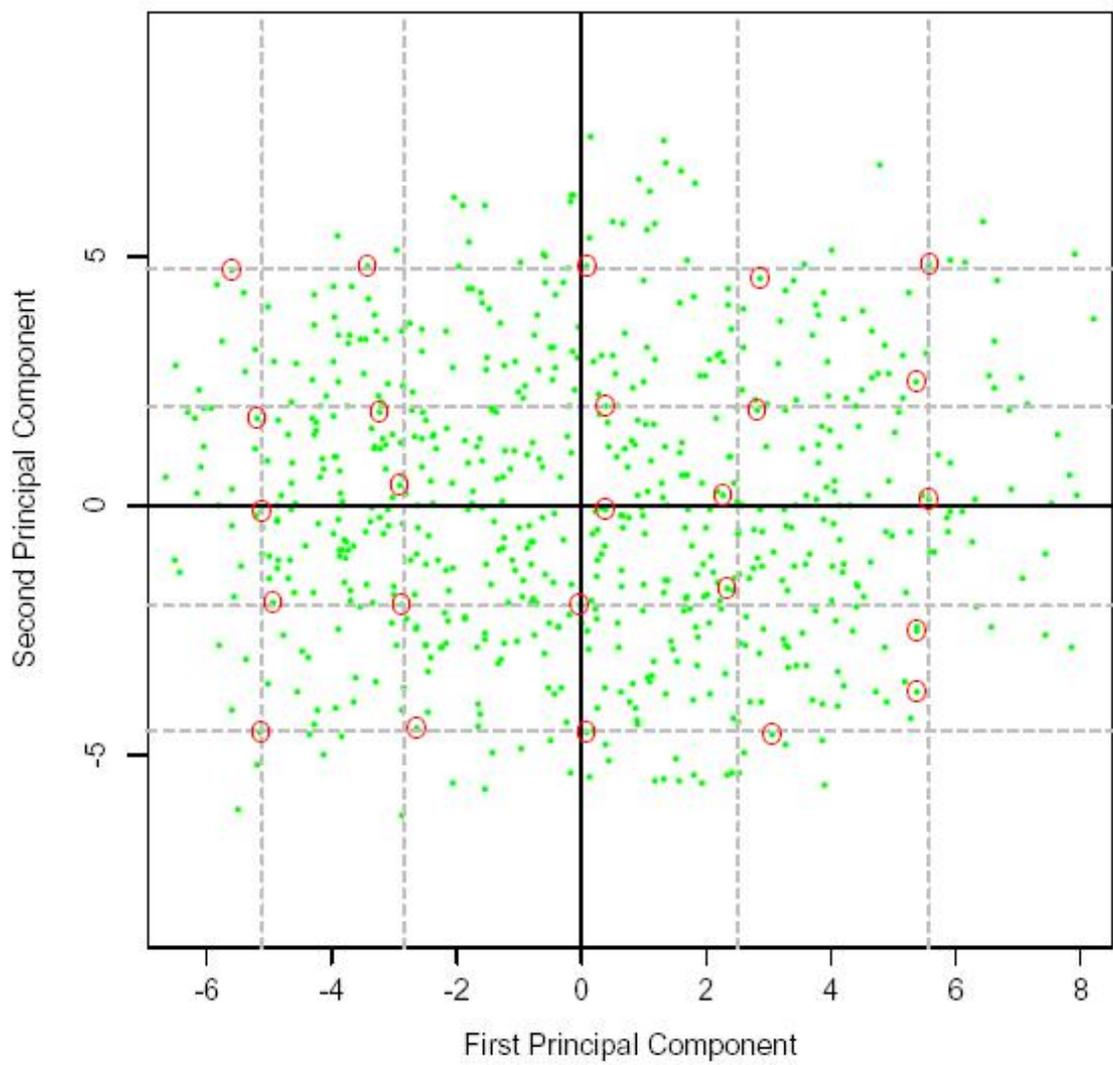
## Erkennung handgeschriebener Ziffern

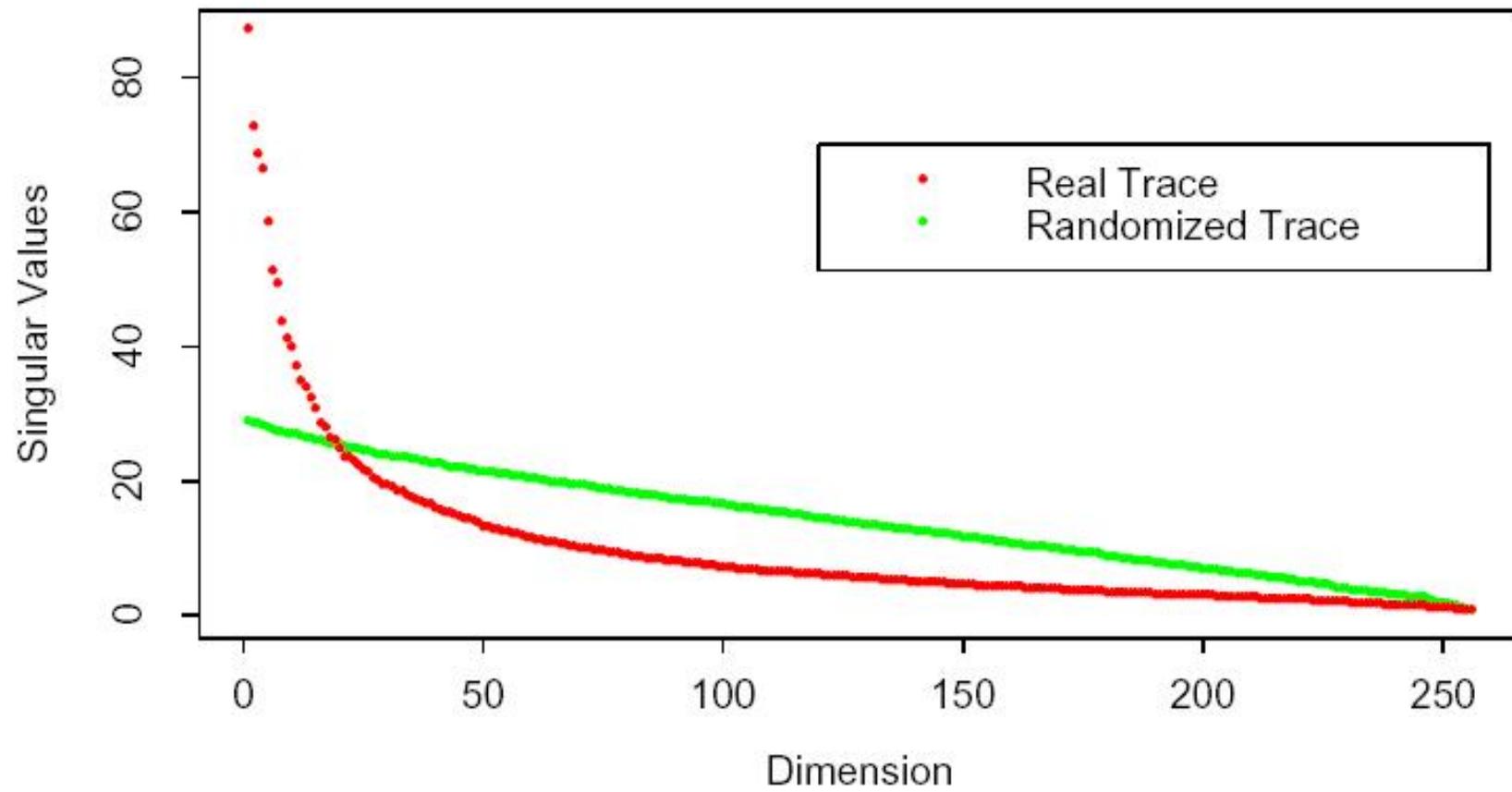
- Dimensionsreduktion und Kompression
- 130 handgeschriebene Ziffern “ 3 ” (insgesamt: 658): beträchtliche Unterschiede in der Schreibweise
- $16 \times 16$  grauwertiges Bild: jedes Bild ist ein Punkt im 256-dimensionalen Raum
- $\mathbf{x}_i$  ist ein 256-dimensionaler Vektor aus den Pixelwerten des  $i$ -ten Bildes



$$\hat{\mathbf{x}}_i = \mathbf{m} + w_{1,i} \cdot \mathbf{u}_1 + w_{2,i} \cdot \mathbf{u}_2.$$

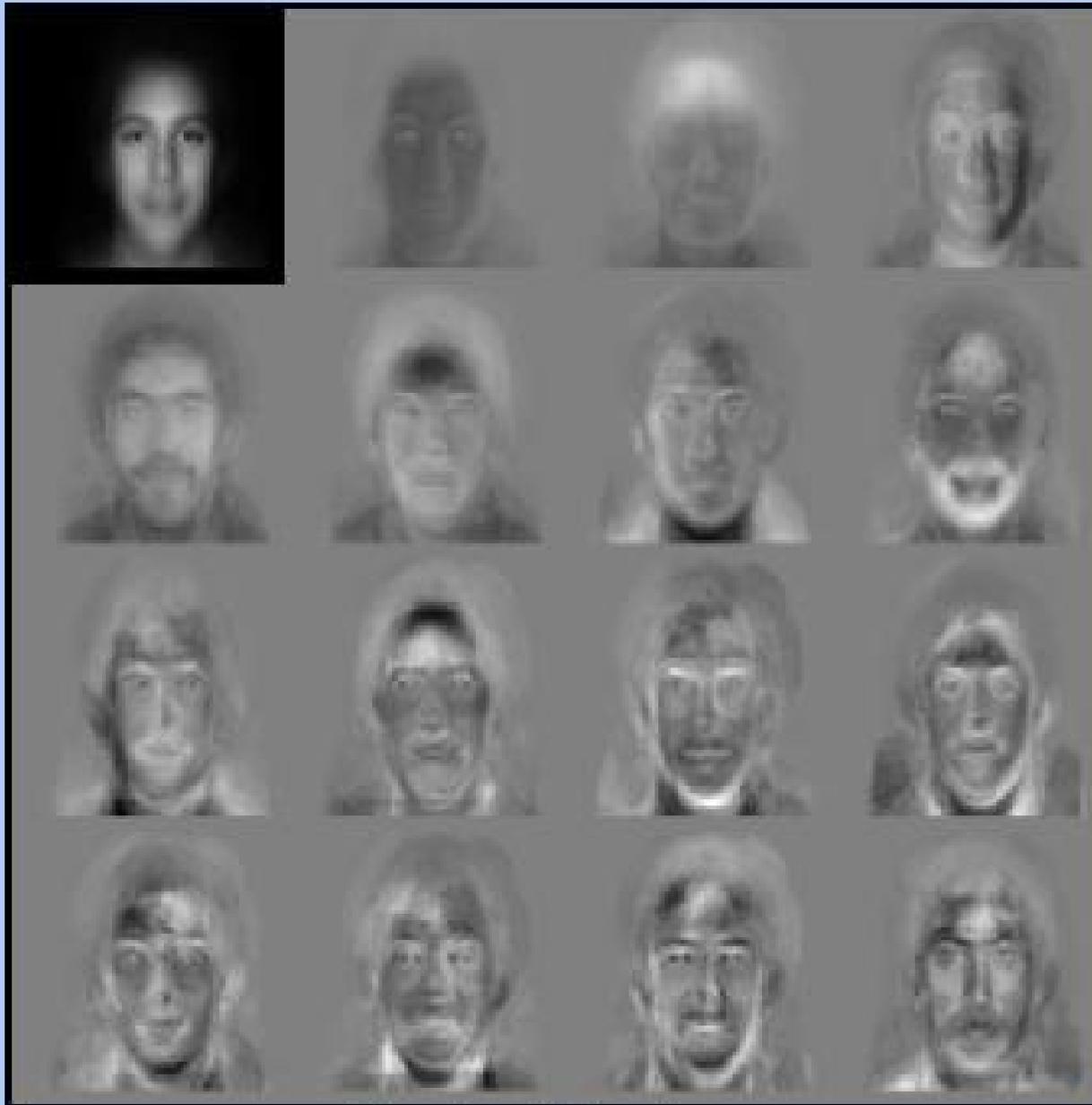
The equation shows the reconstruction of a handwritten digit  $\hat{\mathbf{x}}_i$  as a linear combination of a mean vector  $\mathbf{m}$  and two principal components  $\mathbf{u}_1$  and  $\mathbf{u}_2$ . Each term is represented by a grayscale image of the digit '3' enclosed in a red box. The mean vector  $\mathbf{m}$  is the most prominent digit. The principal components  $\mathbf{u}_1$  and  $\mathbf{u}_2$  represent variations in the digit's appearance, such as slant and thickness. The weights  $w_{1,i}$  and  $w_{2,i}$  are scalar values that determine the contribution of each component to the final reconstruction.





## Eigengesichter (Eigenfaces)

- PCA zur Gesichtserkennung
- <http://vismod.media.mit.edu/vismod/demos/facerec/basic.html>
- 7562 Bilder von etwa 3000 Personen
- $\mathbf{x}_i$  enthält die Pixelwerte des  $i$ -ten Bildes
- Eigengesichter sind basierend auf 128 Bildern von Gesichtern berechnet worden
- Zur Erkennung wurden die ersten 20 Eigenvektoren (=Eigengesichter) ( $r = 20$ ) benutzt
- Fast jede Person hat mindestens 2 Bilder; viele Personen haben variierende Bilder mit unterschiedlichem Gesichtsausdruck, unterschiedlicher Haartracht, Barttracht, ...



Standard Eigenfaces

## Ähnlichkeitssuche basierend auf den Hauptkomponenten

In folgendem Bild ist das linke Bild oben das Testbild. Basierend auf dem euklidischen Abstand im Raum der Hauptkomponenten wurden die folgenden 15 Bilder als nächste Nachbarn klassifiziert. Interessanterweise stammen alle 15 Bilder von der korrekten Person, obwohl die Datenbank aus insgesamt 7562 Bildern von unterschiedlichen Personen bestand!

Der Abstand wird berechnet nach

$$dist_{Euklid,PCA} = \sum_{j=1}^r (w(z)_j - w_{j,i})^2$$

Database:

Display mode:

Search metric:

Working Set: 7561

Left button to select  
Middle button to search  
Right button for info

8455 8468 8486 8454

8485 8465 8466 8469

8501 8481 8479 8491

8498 8459 6141 8487

## Eigenfaces (4)

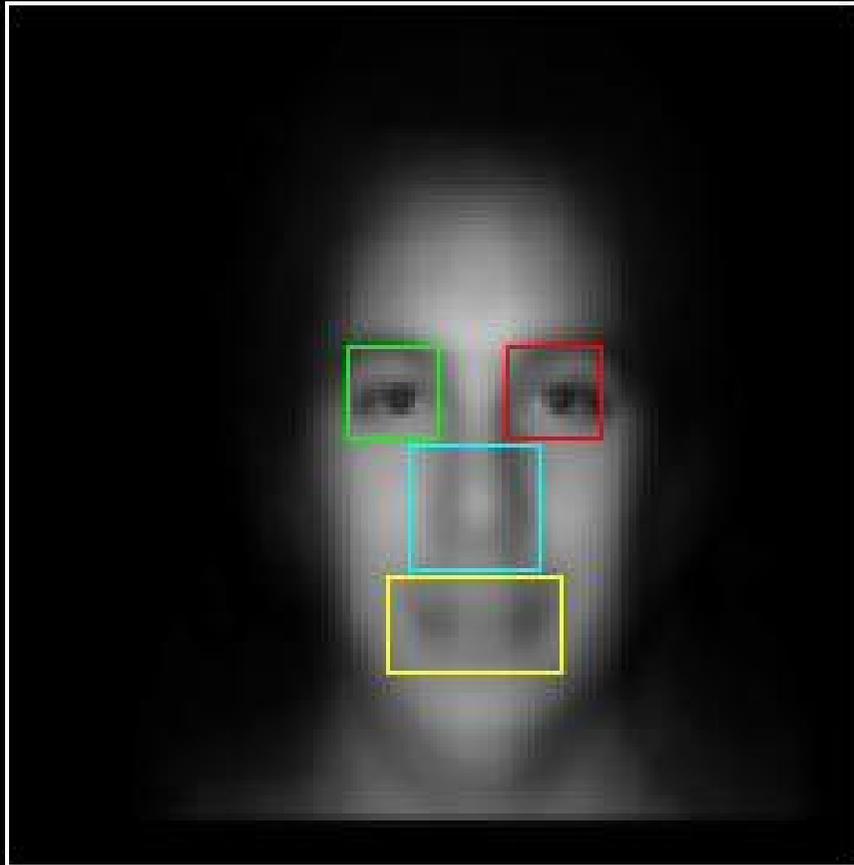
200 Bilder wurden zufällig ausgewählt und dem nächsten Nachbarn im Hauptkomponentenraum zugeordnet. Die Eigenvektor basierte Erkennung war zu 95% richtig.

## Modulare Eigengräume: Detektion, Kodierung and Erkennung

Die Eigengesichtsmethode lässt sich leicht auch auf Gesichtsmerkmale anwenden, was zu Eigenaugen, Eigennasen und Eigenmänder führt. Untersuchungen der menschlichen Augenbewegungen bestätigen, dass sich der Mensch ebenfalls auf diese Merkmale in der Erkennung konzentriert.

Die niedrigaufgelöste Darstellung des Gesamtgesichtes wird ergänzt durch genauere Einzelheiten der lokalen Gesichtsmerkmale.

# Facial Feature Domains



## Modulare Eigengräume: Detektion, Kodierung and Erkennung (2)

Die modularen Methoden benötigen eine automatische Detektion der Merkmale (Augen, Nase, Mund)

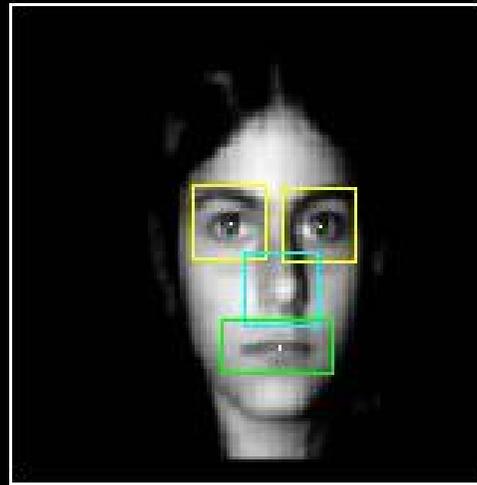
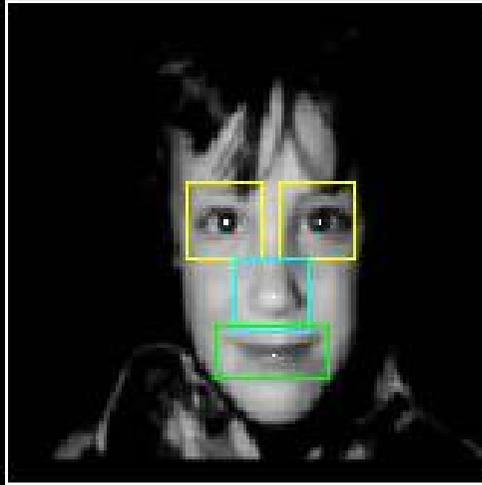
Man definiert ein rechteckiges Fenster, welches durch das zentrale Pixel im Rechteck indiziert wird.

Man berechnet im Fenster den quadratischen Abstand zwischen rekonstruiertem Bild und den roh-Pixelwerten im Fenster, basierend auf einer Hauptkomponentenanalyse die z.b. auf der Bildklasse linkes *Augen berechnet* wurde.

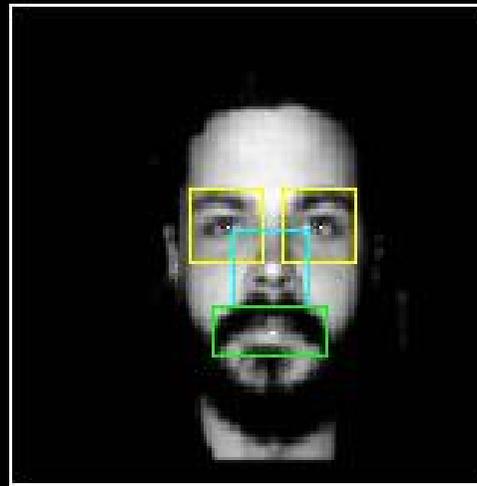
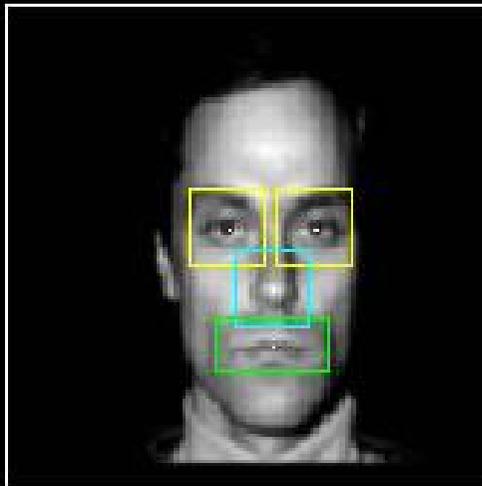
$$DFFS(\text{linkes Auge, Fenster}) = \sum_{\text{pixel } j \text{ im Fenster}} (\hat{z}_j - z_j)^2$$

DFFS(linkes Auge) (distance from feature space) bewertet also, wie anomal der Bildausschnitt im Rechteck ist, falls es sich im Bildausschnitt um ein (beliebiges) linkes Auge handeln würde (Anomaliedetektion)

DFFS(linkes Auge) wird für jedes mögliche Fenster berechnet; in den folgenden Bildern entspricht die Helligkeit dem Restfehler; ebenso für rechtes Auge, Nase, Mund.



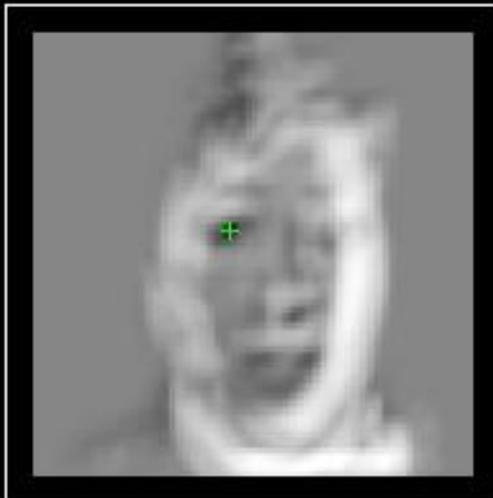
Training Templates



Input Image



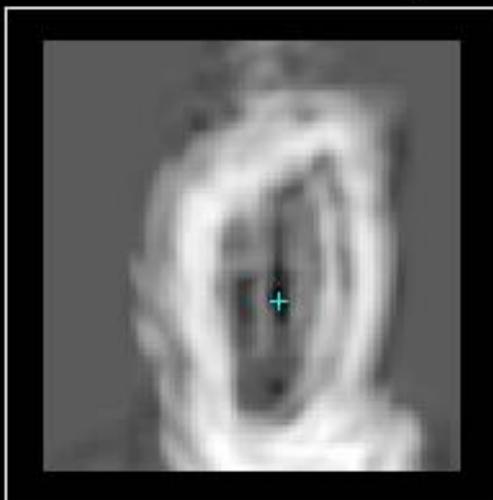
Distance-from-LEye-Space



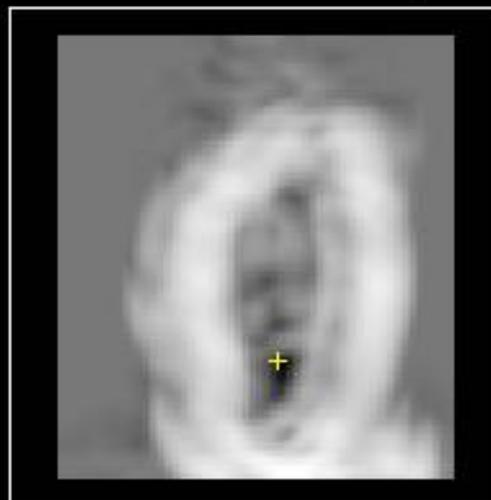
Distance-from-REye-Space



Distance-from-Nose-Space

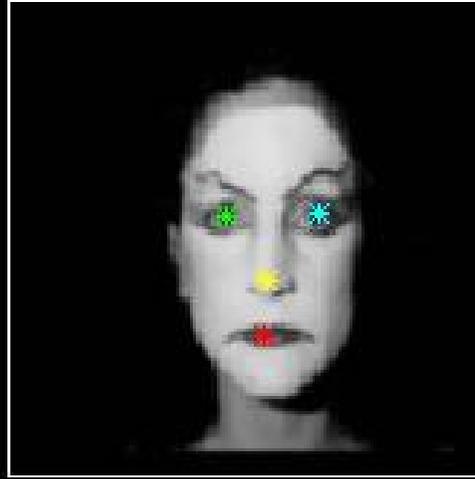
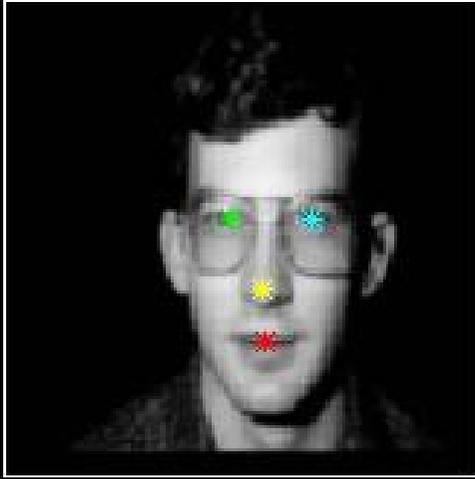


Distance-from-Mouth-Space

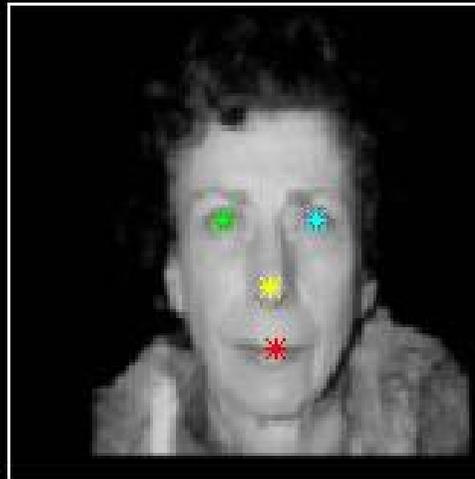
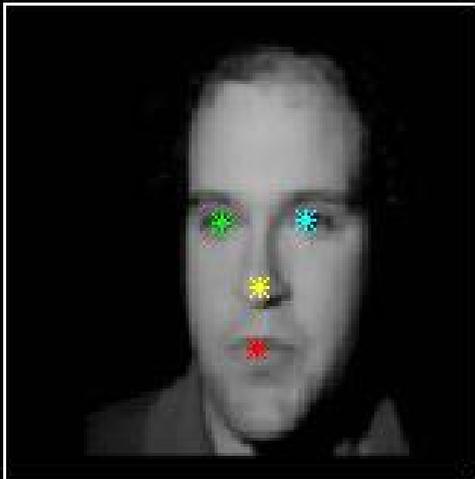


## Feature Detections





Typical Detections



## Detektionsrate

Das Nächste Bild zeigt die Performanz der “linke-Auge-Detektion” basierend auf DFFS mit einem und 10 Eigenvektoren. Gezeigt ist ebenfalls die Performanz für ein einfaches Template matching (Abstand des Fensters zum Mittelwertbild eines Auges = SSD).

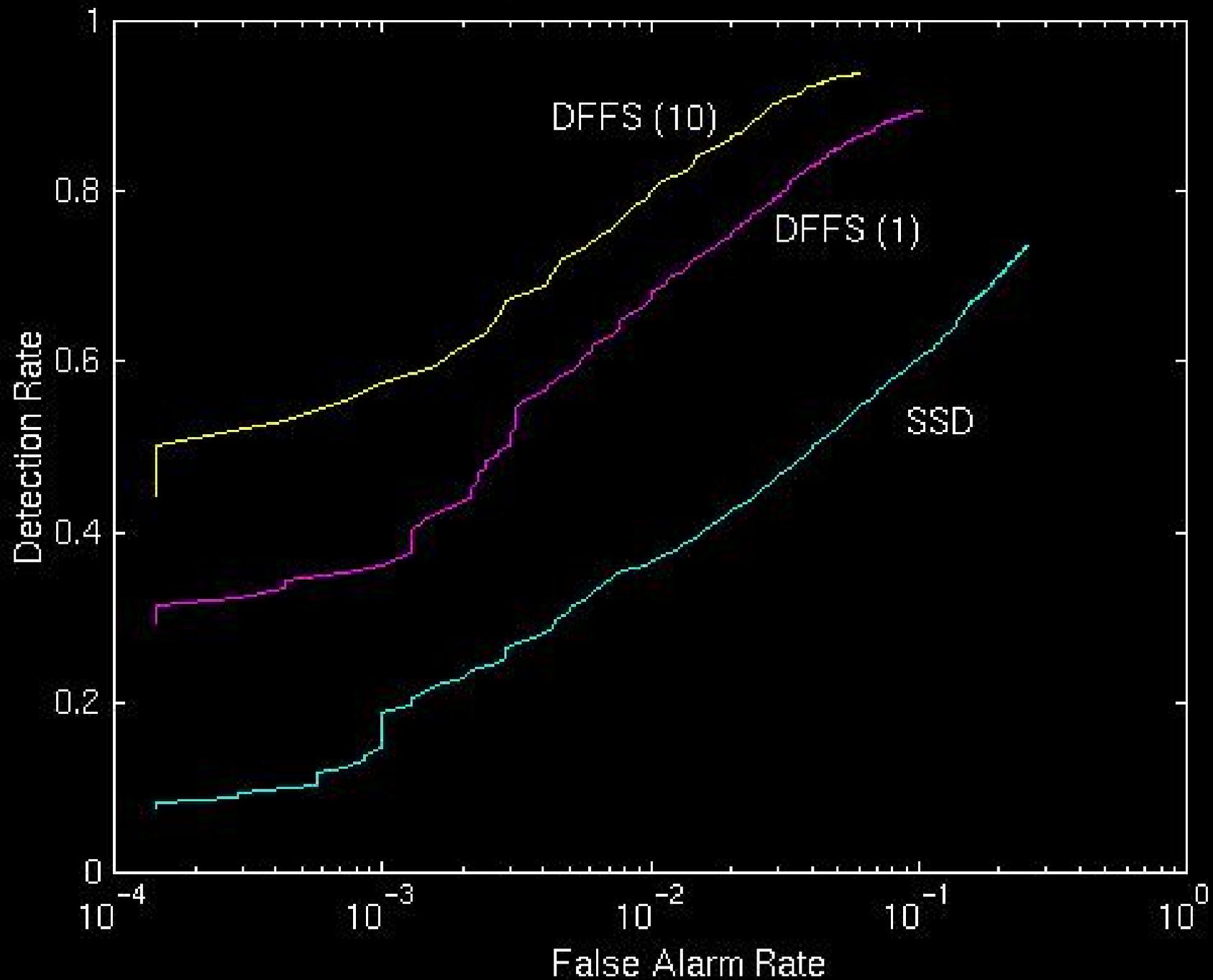
Korrekte Detektion: das globale Minimum ist unter einem Schwellwert  $\alpha$  und ist innerhalb von 5 Pixeln von der richtigen Lokalisierung

Falscher Alarm: das globale Minimum ist unter einem Schwellwert  $\alpha$  und ist außerhalb von 5 Pixeln von der richtigen Lokalisierung

In den Kurven wird der Schwellwert  $\alpha$  variiert

Mit einem entsprechenden Schwellwert erreicht DFFS(10) eine Detektionsrate von 94% und eine Falscher-Alarm-Rate von 6%.

Receiver Operating Characteristics: Left-Eye Detection



## Modulare Erkennung

Es wird getestet, ob der nächste Nachbar in Bezug auf die Hauptkomponenten die gleiche Person wie im Testbild ist.

Die Person im Testbild hat einen anderen Gesichtsausdruck wie die gleiche Person im Trainingsdatensatz

Die Erkennung basierend auf modularen Merkmalen (Augen, Nase, Mund) zeigt grosse Robustheit gegenüber Variationen wie Brille, Schminke, Barttracht, ...

### Novel Test Views



### Eigenface-based Matches



### Eigenfeature-based Matches



## Latent Semantic Analysis (LSA)

- Standard term-frequency (tf) Repräsentation klassifiziert Texte als unähnlich, wenn die genauen Worte nicht in beiden Texten vorkommen
- Ziel: Generalisierung unter Berücksichtigung der Bedeutung der Wörter
- $X$ : tf-Matrix (Wörter mal Dokumente) (mit idf oder ähnlicher Transformation)
- Latent Semantic Analysis (LSA): PCA auf tf-Matrix angewandt
- Notation für SVD:  $X = WSP$
- Rang:  $r = 50 \dots 1500$ ; es gibt eine niedrig-dimensionale Repräsentation, die Texte beschreibt
- Performanz oft ähnlich zu menschlicher Performanz in vielen Aufgaben: Assoziation, semantische Ähnlichkeit
- Beispiel: Bewertung von Term-papers

## Einfaches Beispiel

9 Texte: 5 über Mensch-Maschine Interaktion (c1 - c5), 4 ueber mathematische Graph Theorie (m1 - m4)

12 Schlüsselwörter (kursiv) und tf-Matrix

Example of text data: Titles of Some Technical Memos

- c1: *Human machine interface for ABC computer applications*
- c2: *A survey of user opinion of computer system response time*
- c3: *The EPS user interface management system*
- c4: *System and human system engineering testing of EPS*
- c5: *Relation of user perceived response time to error measurement*
  
- m1: *The generation of random, binary, ordered trees*
- m2: *The intersection graph of paths in trees*
- m3: *Graph minors IV: Widths of trees and well-quasi-ordering*
- m4: *Graph minors: A survey*

$$\{X\} =$$

	<b>c1</b>	<b>c2</b>	<b>c3</b>	<b>c4</b>	<b>c5</b>	<b>m1</b>	<b>m2</b>	<b>m3</b>	<b>m4</b>
<b>human</b>	1	0	0	1	0	0	0	0	0
<b>interface</b>	1	0	1	0	0	0	0	0	0
<b>computer</b>	1	1	0	0	0	0	0	0	0
<b>user</b>	0	1	1	0	1	0	0	0	0
<b>system</b>	0	1	1	2	0	0	0	0	0
<b>response</b>	0	1	0	0	1	0	0	0	0
<b>time</b>	0	1	0	0	1	0	0	0	0
<b>EPS</b>	0	0	1	1	0	0	0	0	0
<b>survey</b>	0	1	0	0	0	0	0	0	1
<b>trees</b>	0	0	0	0	0	1	1	1	0
<b>graph</b>	0	0	0	0	0	0	1	1	1
<b>minors</b>	0	0	0	0	0	0	0	1	1

$$\underline{r}(\text{human.user}) = -.38$$

$$\underline{r}(\text{human.minors}) = -.29$$

$$\{X\} = \{W\}\{S\}\{P\}$$

$$\{W\} =$$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$$\{S\} =$$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$$\{P\} =$$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

## Rekonstruktion

Rekonstruktion  $\hat{X}$  mit  $r = 2$  (Beachte: m4: survey, trees) (Beachte: human, user, minors))

$\{\hat{X}\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

$\underline{r}(\text{human.user}) = .94$

$\underline{r}(\text{human.minors}) = -.83$

## Korrelation im Originalraum und in der Rekonstruktion

Correlations between titles in raw data:

	c1	c2	c3	c4	c5	m1	m2	m3
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

0.02  
-0.30 0.44

Correlations in two dimensional space:

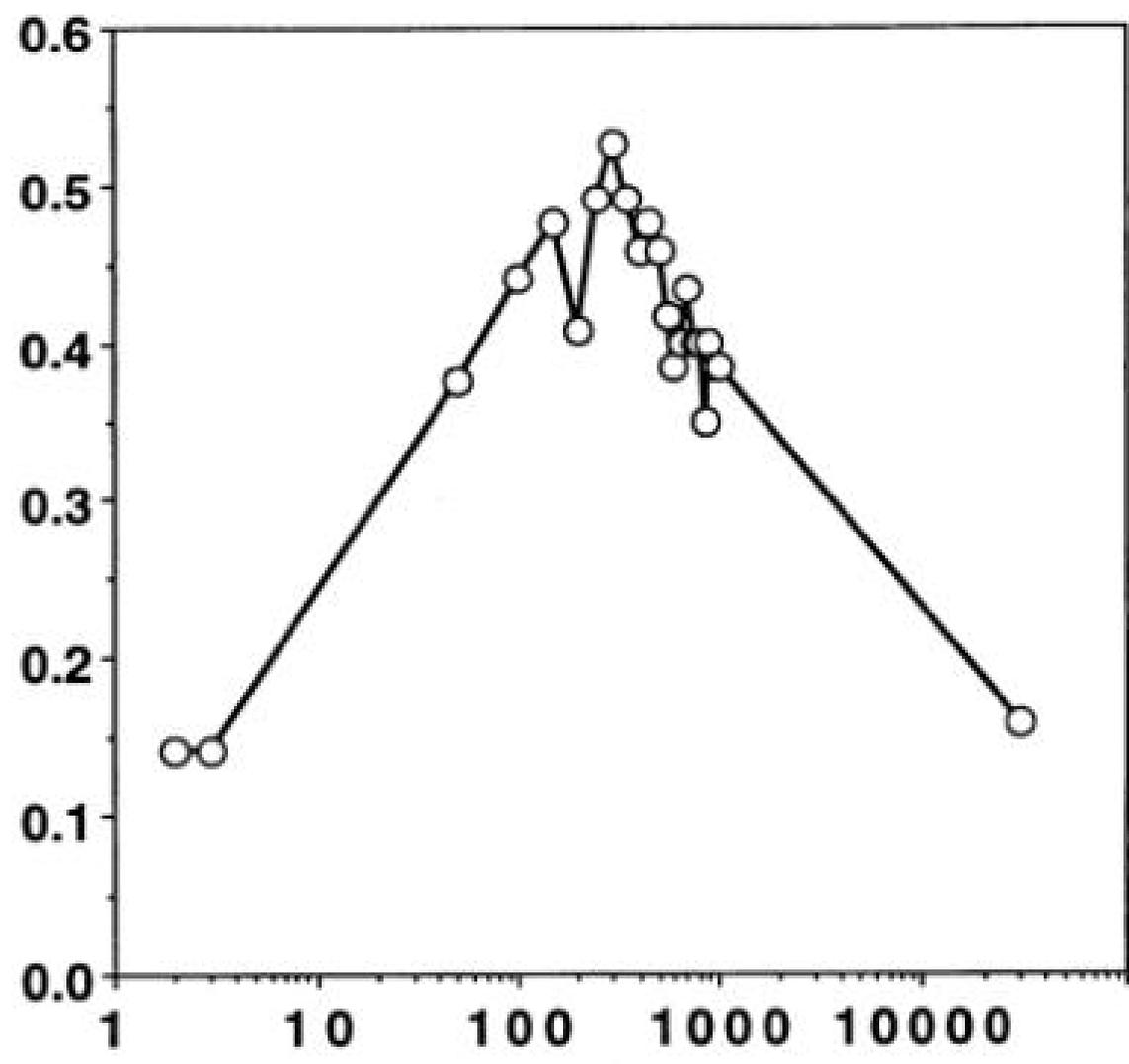
c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

0.92  
-0.72 1.00

## Weitere Eigenschaften von LSA

- LSA und Information Retrieval (Suchmaschinen) (Latent Semantic Indexing (LSI)): finde ähnliche Dokumente
- LSA entdeckt Synonyme
- Beachte: LSA basierte Algorithmen sind sehr populär für *collaborative filtering* mit einer Matrix, bei der Zeilen Benutzern entsprechen, Spalten Objekten (Filmen) entsprechen und der Matrixeintrag der Bewertung des Objektes durch den Benutzer entspricht; fehlende Bewertungen werden durch den Wert Null dargestellt und über LSA berechnet

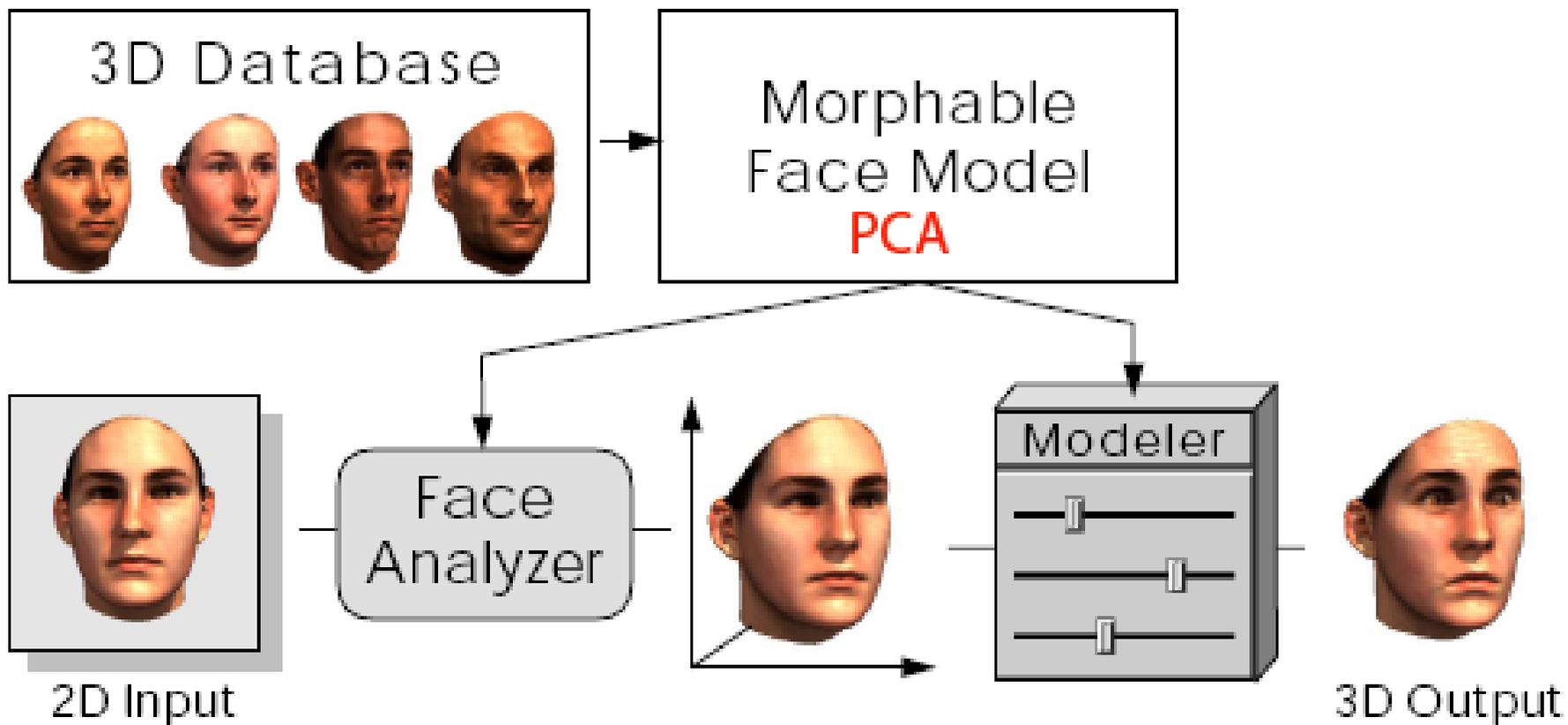
**Proportion Correct on Synonym Test**



**Number of Dimensions in LSA (log)**

## 3D Morphable Head Model

- Volker Blanz & Thomas Vetter: A Morphable Model For The Synthesis Of 3D Faces, SIGGRAPH 99  
`www.mpi-inf.mpg.de/~blanz`
- 3D Scans von Gesichtern werden in Korrespondenz gesetzt.  
Damit kann jedes Gesichts als Vektor  $\mathbf{x}_i \in \mathbb{R}^n$  repräsentiert werden ( $n = 80000$ ).
- PCA hilft bei der Dimensionsreduktion.



ORIGINAL



CARICATURE



MORE MALE



FEMALE



SMILE



FROWN



WEIGHT



HOOKED NOSE