Hauptkomponentenanalyse

Volker Tresp

Mathematische Vorbereitung: Singulärwertzerlegung

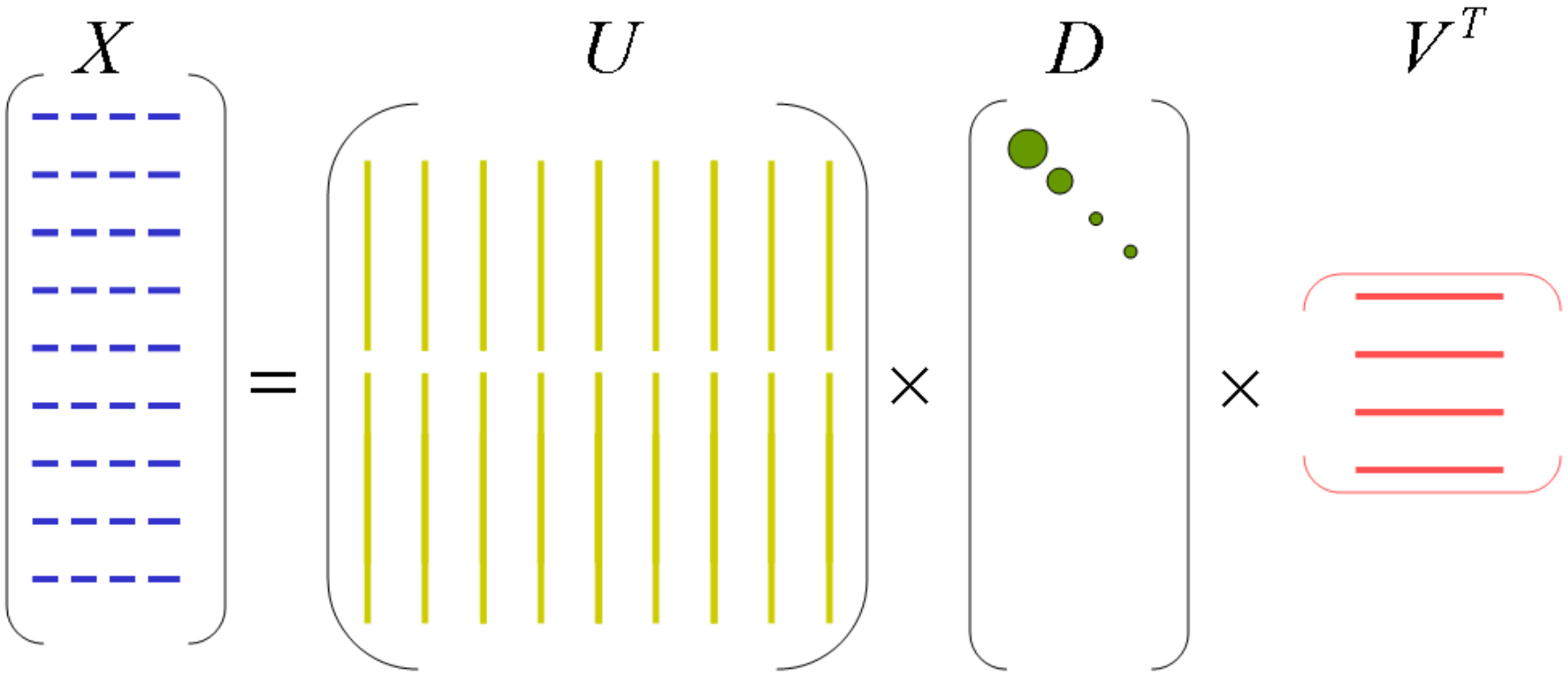
Singulärwertzerlegung

- Singulärwertzerlegung, auch SVD von *Singular Value Decomposition*
- Eine beliebige (rechteckige, quadratische, ...) $N \times M$ Matrix X läßt sich zerlegen in

$$X = UDV^T$$

  wobei U und V beide **orthonormale** Matrizen sind. U ist eine $N \times N$ Matrix und und V ist $M \times M$ Matrix.

- D ist eine $N \times M$ **Diagonalmatrix** mit diagonalen Einträgen (Singulärwerten) $d_i \geq 0, i = 1, \ldots, \tilde{r}$, mit $\tilde{r} = \min(M, N)$
- Die $\mathbf{u}_j$ sind die linken Singulärvektoren
- Die $\mathbf{v}_j$ sind die rechten Singulärvektoren
- d_j sind die Singulärwerte (singular values)


X
U
D
V^T

Kovarianzmatrix und Kernmatrix

- Es ergibt sich für die empirische Kovarianzmatrix

$$N \times \Sigma = X^T X = V D^T U^T U D V^T = V D^T D V^T = V D_V V^T$$

- Und für die empirische Kernmatrix

$$M \times K = X X^T = U D V^T V D^T U^T = U D D^T U^T = U D_U U^T$$

- Mit

$$N \times \Sigma V = V D_V \quad M \times K U = U D_K$$

sieht man, dass die Spalten V die Eigenvektoren zu Σ sind und die Spalten U die Eigenvektoren zu K sind. Die Eigenwerte sind die Diagonaleinträge von D_V, bzw D_U.

Umformungen

- Die Singulärwertzerlegung ist

$$X = UDV^T$$

  woraus man ableiten kann

$$X = UU^T X$$

$$X = XVV^T$$

Reduzierter Rang

- In der SVD sind die d_i sind monoton geordnet: $d_1 \geq d_2 \geq d_3 ... \geq d_{\tilde{r}}$. In vielen Fällen kann man $d_i, i > r$ vernachlässigen und identisch Null setzen und man erhält eine Rang-r Approximation. Sei D_r eine Diagonalmatrix mit den entsprechenden Einträgen. Dann ist die Approximation

$$\hat{X} = U_r D_r V_r^T$$

$$\hat{X} = U_r U_r^T X$$

$$\hat{X} = X V_r V_r^T$$

wobei U_r die ersten r Spalten von U enthält. Entsprechend V_r.

Beste Approximation

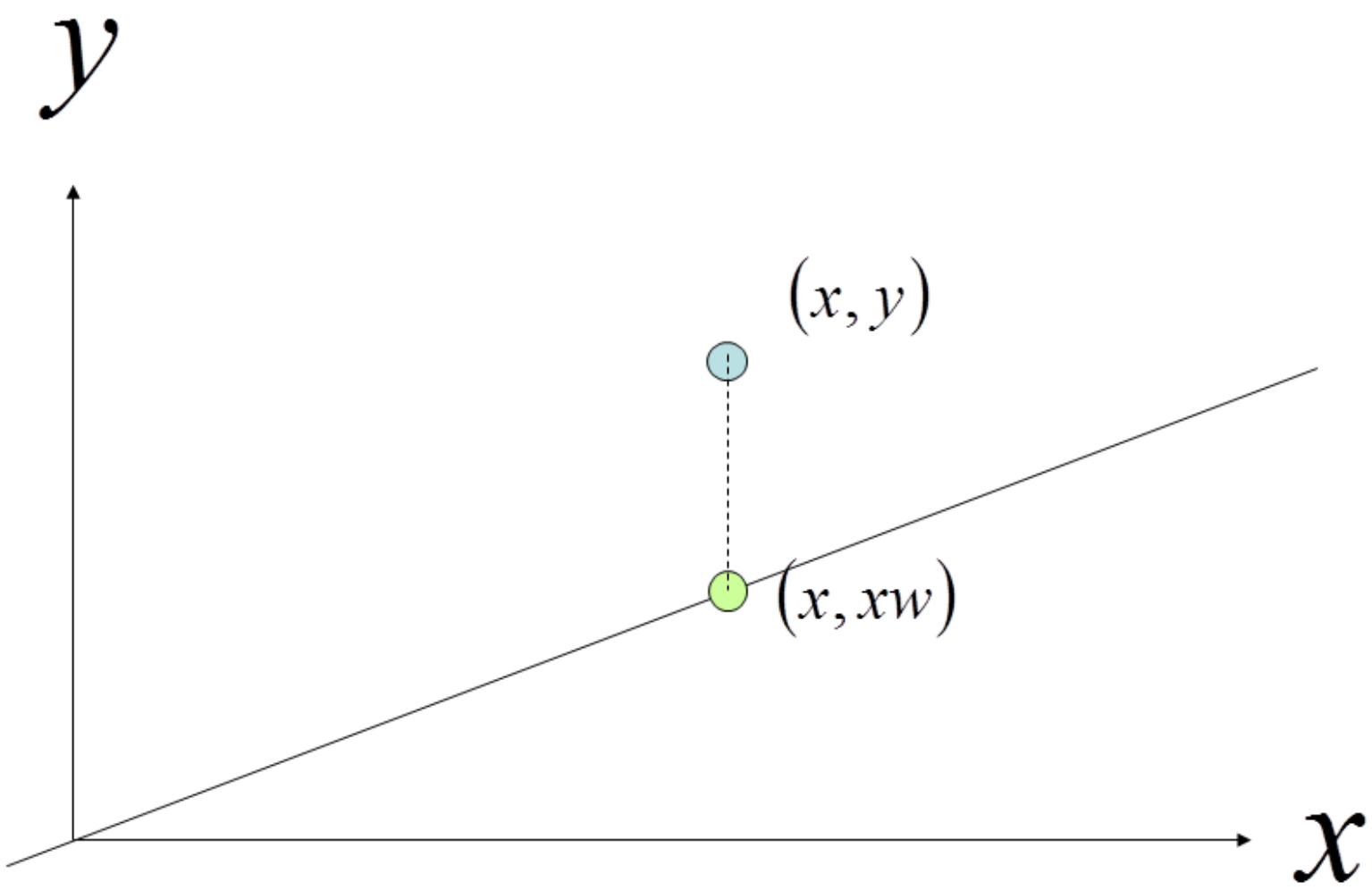
- Die obige Approximation ist die beste rank-r Approximation in Bezug auf den quadratischen Fehler (Frobenius Norm). Der Approximationsfehler wird

$$\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{j,i}-\hat{x}_{j,i})^2=\sum_{j=r+1}^{\tilde{r}}d_j^2$$

Hauptkomponentenanalyse

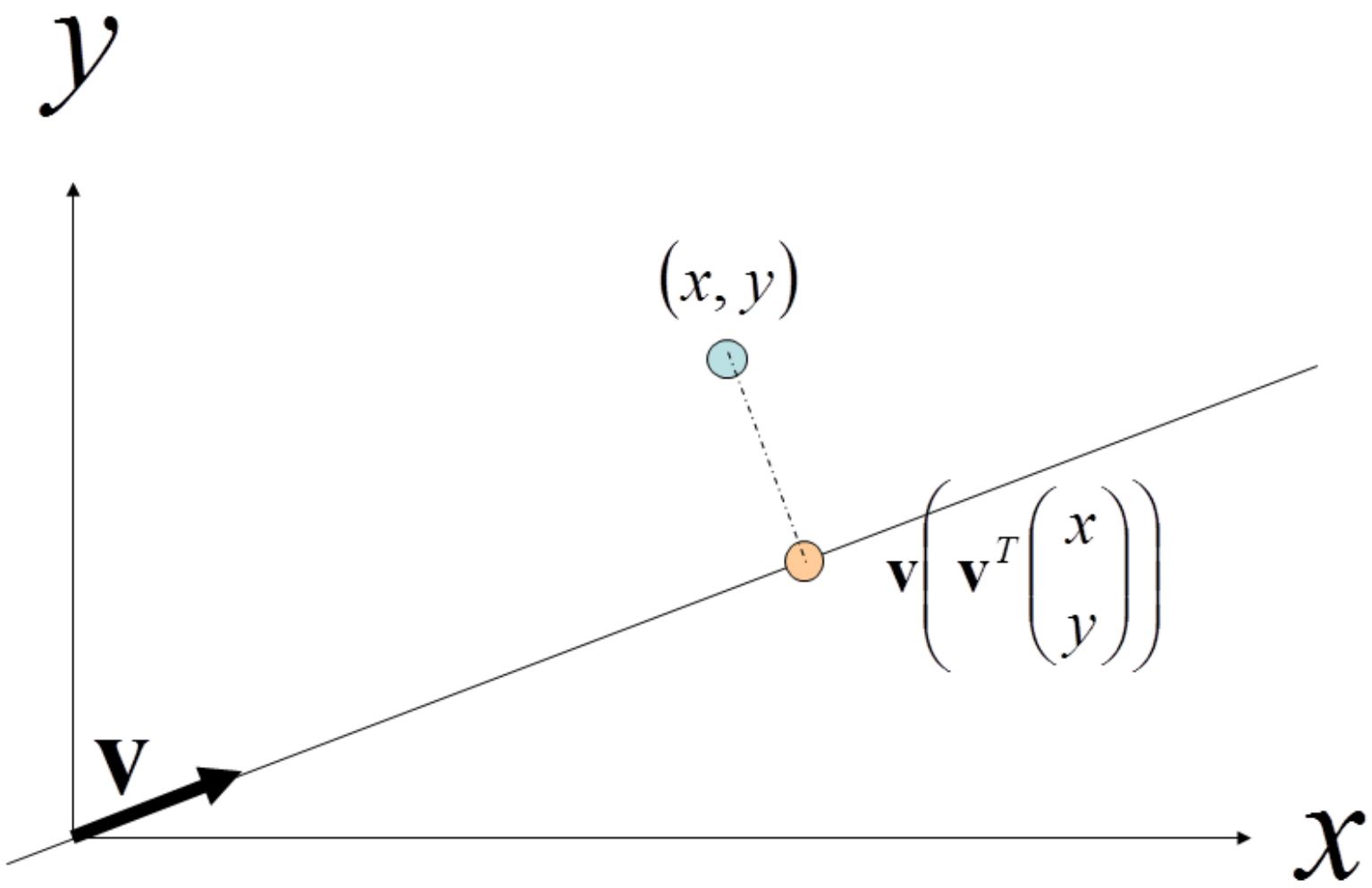
Review: Regression

- Betrachte 1-D Regression
- Ein Datenpunkt $(x, y)^T$ wird geschätzt als $(x, xw)^T$
- Nur y wird "bereinigt"


y
(x, y)
(x, xw)
x

Projektion

- Ein Datenpunkt $(x, y)^T$ wird auf die Gerade projiziert und geschätzt als $\mathbf{v}\mathbf{v}^T(x, y)^T$
- Sowohl y als auch x werden "bereinigt"


y
(x, y)
v(v^T(x y))
v
x

Berechnung des Optimalen Projektionsvektors

- Die Frage stellt sich nun, welches die optimale Projektionsebene sein soll und welche Dimension diese haben sollte

Berechnung des Optimalen Projektionsvektors

- Wir suchen den Vektor $\mathbf{v}$ der Länge 1, der den quadratischen Rekonstruktionsfehler über alle Datenpunkte minimiert (wir unterscheiden jetzt nicht mehr zwischen $\mathbf{x}$ und y)

$$\sum_{i=1}^{N} (\mathbf{v}\mathbf{v}^T\mathbf{x}_i - \mathbf{x}_i)^T(\mathbf{v}\mathbf{v}^T\mathbf{x}_i - \mathbf{x}_i) + \lambda(\mathbf{v}^T\mathbf{v} - 1)$$

$$= \sum_{i=1}^{N} \mathbf{x}_i^T\mathbf{v}\mathbf{v}^T\mathbf{v}\mathbf{v}^T\mathbf{x}_i + \mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T\mathbf{v}\mathbf{v}^T\mathbf{x}_i - \mathbf{x}_i^T\mathbf{v}\mathbf{v}^T\mathbf{x}_i + \lambda(\mathbf{v}^T\mathbf{v} - 1)$$

$$= \sum_{i=1}^{N} \mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T\mathbf{v}\mathbf{v}^T\mathbf{x}_i + \lambda(\mathbf{v}^T\mathbf{v} - 1)$$

Hauptkomponenten

- Die Ableitung nach $\mathbf{v}$ ergibt

$$\sum_{i=1}^{N} \mathbf{x}_i^T \mathbf{x}_i \mathbf{v} = \lambda \mathbf{v}$$

  oder in Matrix form

$$\Sigma \mathbf{v} = \lambda \mathbf{v}$$

- Somit ist der optimale Projektionsvektor der erste Eigenvektor von Σ
- Eine verbesserte Approximation erhält man wenn man auf r Projektionsvektoren projiziert; man kann zeigen, dass sich die optimalen Projektionsvektoren sich aus den ersten r rechten Singulärvektoren ergeben und man erhält

$$\hat{\mathbf{x}}_i = V_r V_r^T \mathbf{x}_i$$

Bereinigte Datenmatrix

- Somit stellt

$$\widehat{X} = UD_rV^T = U_rU_r^TX = XV_rV_r^T$$

eine bereinigte (rauschreduzierte) Datenmatrix dar.

- Betrachten wir

$$\mathbf{x}_i \to \mathbf{z}_i = V^T x_i \to \widehat{\mathbf{x}}_i = V\mathbf{z}_i$$

dann ist $\widehat{\mathbf{x}}_i$ die bereinigte Version von $\mathbf{x}_i$. Man kann auch sagen, dass $\widehat{\mathbf{x}}_i$ durch die **Hauptkomponenten** $\mathbf{z}_i$ dargestellt wird

Zentrierte Hauptkomponentenanalyse

- In manchen Anwendungen ist es vorteilhaft, zunächst den Mittelwert in jeder Dimension abzuziehen; der Mittelwert an sich liefert ja keine Information über einen spezifischen Datenvektor

$$\tilde{x}_{i,j} = x_{i,j} - m_j$$

wobei

$$m_j = \frac{1}{N} \sum_{i=1}^{N} x_{j,i}$$

- Entsprechend enthält $\tilde{X}$ die skalierten Datenvektoren
- Zentrieren ist empfohlen, wenn die Daten (annähernd) Gauß-verteilt sind

Hauptkomponentenanalyse mit zentrierten Daten

- Sei die SVD:

$$\tilde{X} = \tilde{U}\tilde{D}\tilde{V}^T$$

dann ist

$$\hat{\mathbf{x}}_i = \mathbf{m} + \sum_{l=1}^{r} \tilde{\mathbf{v}}_l \tilde{z}_{i,l}$$

mit $m = (m_1, \ldots, m_M)^T$

$$\tilde{z}_{i,l} = \tilde{\mathbf{v}}_l^T \tilde{\mathbf{x}}_i$$

Anwendungen der Hauptkomponentenanalyse

Volker Tresp

Die SVD liefert verbesserte Merkmale

- Anstatt mit den originalen Merkmalen $\mathbf{x}_i$ arbeiten wir mit den bereinigten Merkmalen $\widehat{\mathbf{x}}_i$
- Anstatt mit den originalen Merkmalen $\mathbf{x}_i$ arbeiten wir mit den Hauptkomponenten $\widehat{\mathbf{z}}_i$. Wir erhalten eine Dimensionsreduktion. Dieses Verfahren wird auch Hauptkomponenten-Regression genannt
- Wir berechnen euklidische Abstände nicht im Originalraum $\mathsf{dist}_x(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ sondern im rekonstruierten Raum

$$\mathsf{dist}_{\widehat{x}}(i,j) = \|\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_j\|$$

oder im Raum der Hauptkomponenten

$$\mathsf{dist}_z(i,j) = \|\widehat{\mathbf{z}}_i - \widehat{\mathbf{z}}_j\|$$

Äquivalenz

- Es gilt: $\text{dist}_{\hat{x}}(i,j) = \text{dist}_z(i,j)$
- Beweis:

$$\begin{aligned}\text{dist}_{\hat{x}} &= (U_r\mathbf{z}_i - U_r\mathbf{z}_j)^T(U_r\mathbf{z}_i - U_r\mathbf{z}_j) \\ &= (\mathbf{z}_i - \mathbf{z}_j)^T U_r^T U_r(\mathbf{z}_i - \mathbf{z}_j) \\ &= (\mathbf{z}_i - \mathbf{z}_j)^T(\mathbf{z}_i - \mathbf{z}_j) = \text{dist}_z\end{aligned}$$

Detektion von Neuheit

- Angenommen, ich habe einen "normalen" Datensatz mit der Hauptkomponentenanalyse analysiert
- Stammt ein neuer $\mathbf{x}$ aus der gleichen Klasse von "normalen" Datenvektoren, sollte der Rekonstuktionsfehler oder die Anomalie

$$\mathsf{AN}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \sum_{j=1}^{M} (\hat{x}_j - x_j)^2$$

  klein sein.
- Ist der Abstand groß, so ist $\mathbf{x}$ "anomal", oder "novel"

Singulärwertzerlegung handgeschriebener Ziffern

Datensatz

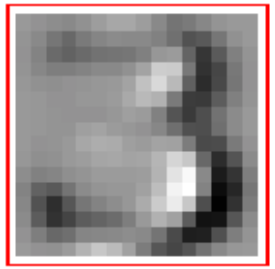
- Dimensionsreduktion und Kompression
- 130 handgeschriebene Ziffern “ 3 ” (insgesamt: 658): beträchtliche Unterschiede in der Schreibweise
- 16×16 grauwertiges Bild: jedes Bild ist ein Punkt im 256-dimensionalen Raum
- $\mathbf{x}_i$ ist ein 256-dimensionaler Vektor aus den Pixelwerten des i-ten Bildes

Visualisierung

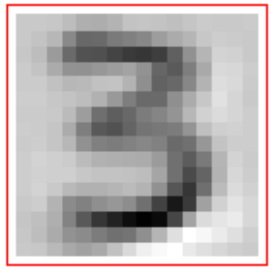
- Die Singulärvektoren $\mathbf{v}_1$, $\mathbf{v}_2$ werden mit Hilfe der SVD berechnet,
- $\mathbf{v}_1$ verlängert den unteren Teil der “3”
- $\mathbf{v}_2$ steht für die Dicke der Ziffer

$$\hat{\mathbf{x}}_i =$$

$$+ z_{i,1} \cdot$$



$$+ z_{i,2} \cdot$$



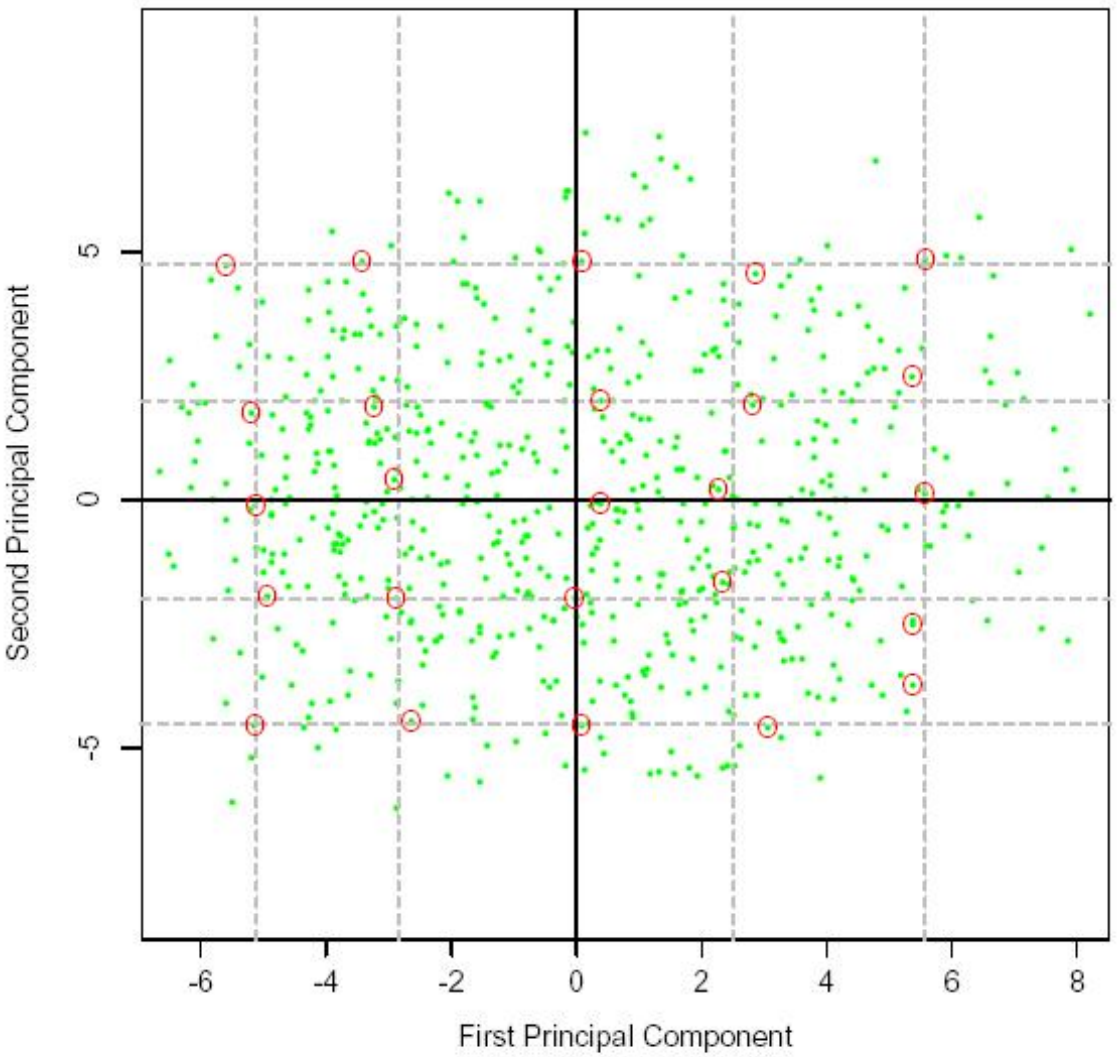
$$\cdot$$

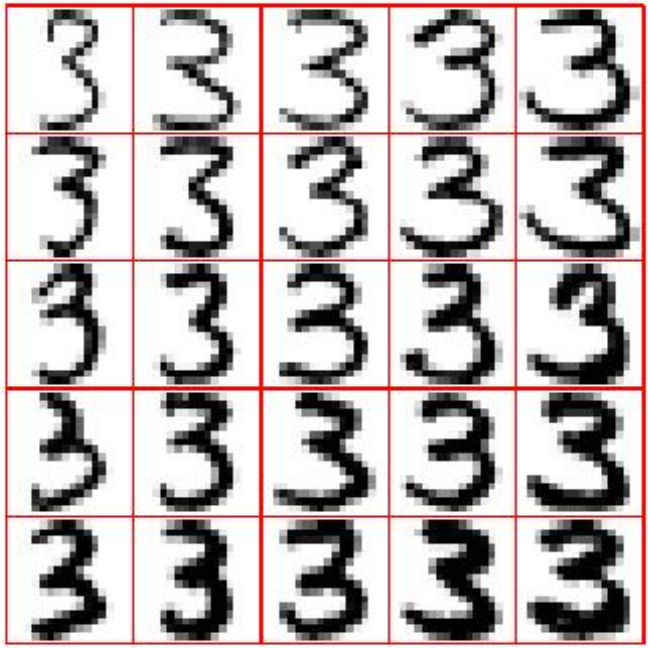
$\mathbf{m}$ $\mathbf{v}_1$ $\mathbf{v}_2$

Visualisierung: Rekonstruktion

- Für verschiedene Werte der Hauptkomponenten $z_{1,i}$ und $z_{2,i}$ wird das rekonstruierte Bild gezeigt

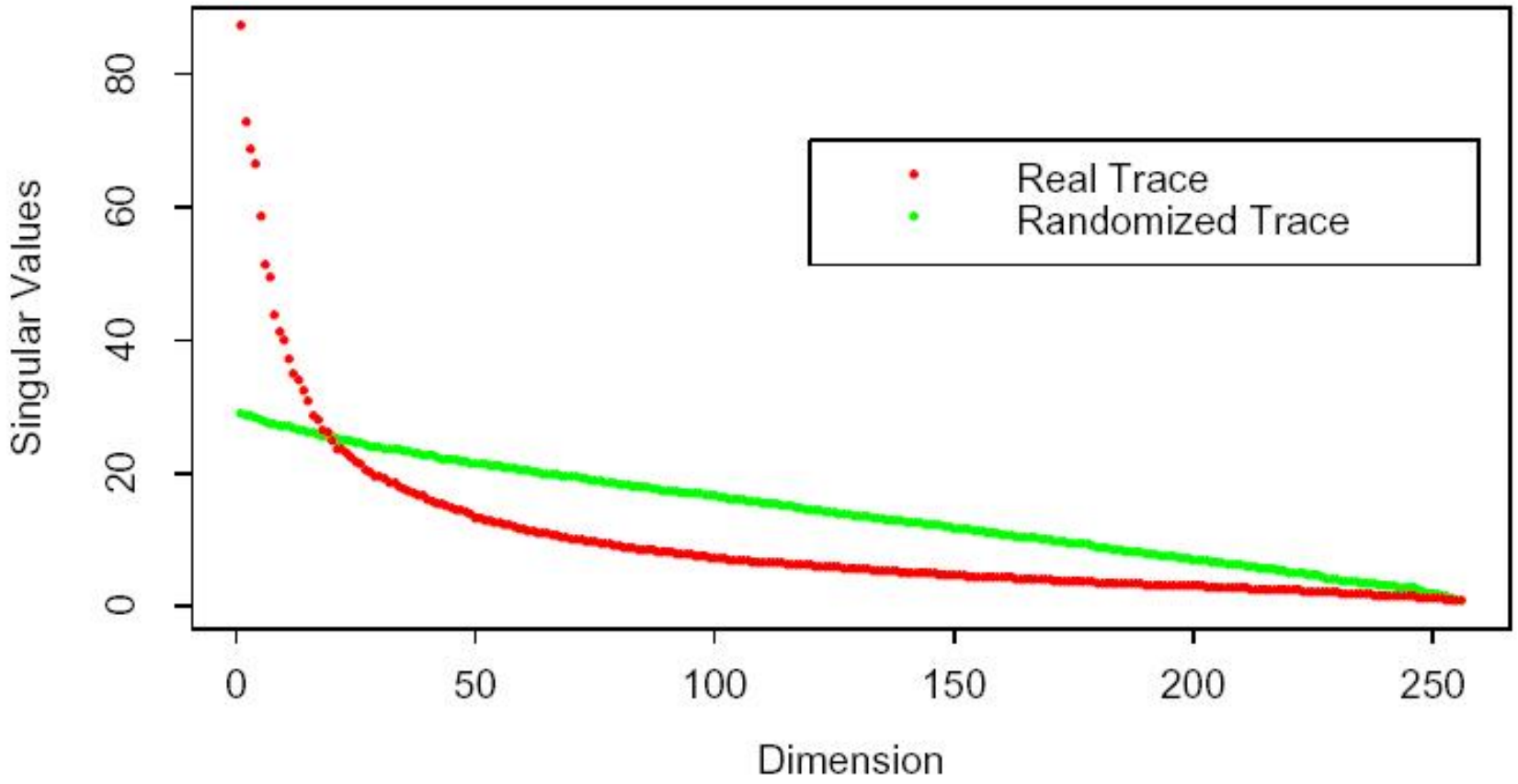
$$\hat{\mathbf{x}}_i = \mathbf{m} + z_{i,1}\mathbf{v}_1 + z_{i,2}\mathbf{v}_2$$


Second Principal Component
5
0
-5
-6
-4
-2
0
2
4
6
8
First Principal Component



Rang der Approximation

- Welchen Rang r sollte die Approximation besitzen?
- Geplottet sind d_i als Funktion von i (rot) für X und für eine X-Matrix, in der die Elemente von $\mathbf{x}_j$ zufällig vertauscht wurden (grün).

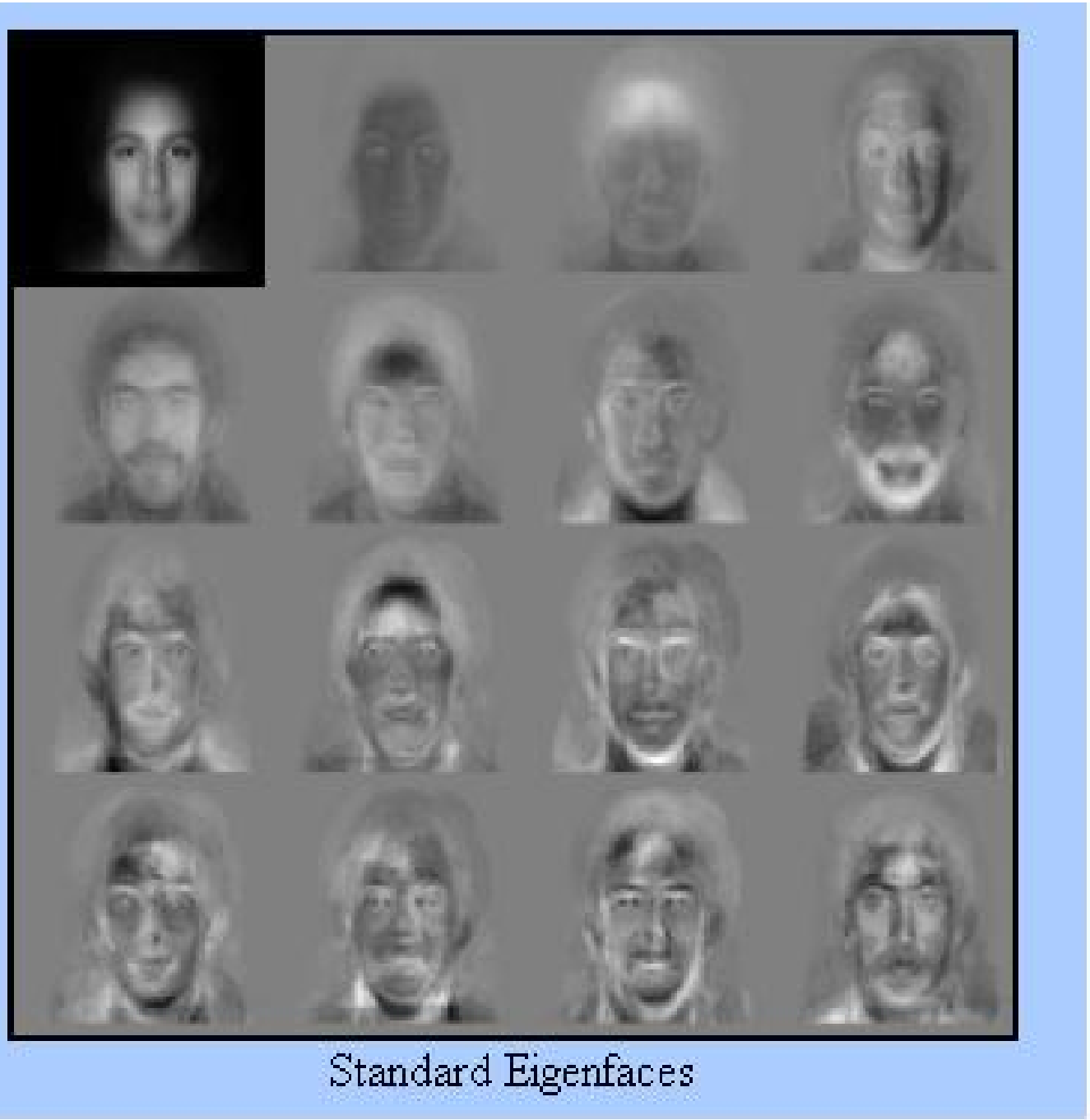

Singular Values
0
20
40
60
80
0
50
100
150
200
250
Dimension
Real Trace
Randomized Trace

Eigengesichter (Eigenfaces)

Eigengesichter (Eigenfaces)

Datensatz

- Hauptkomponentenanalyse 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, ...

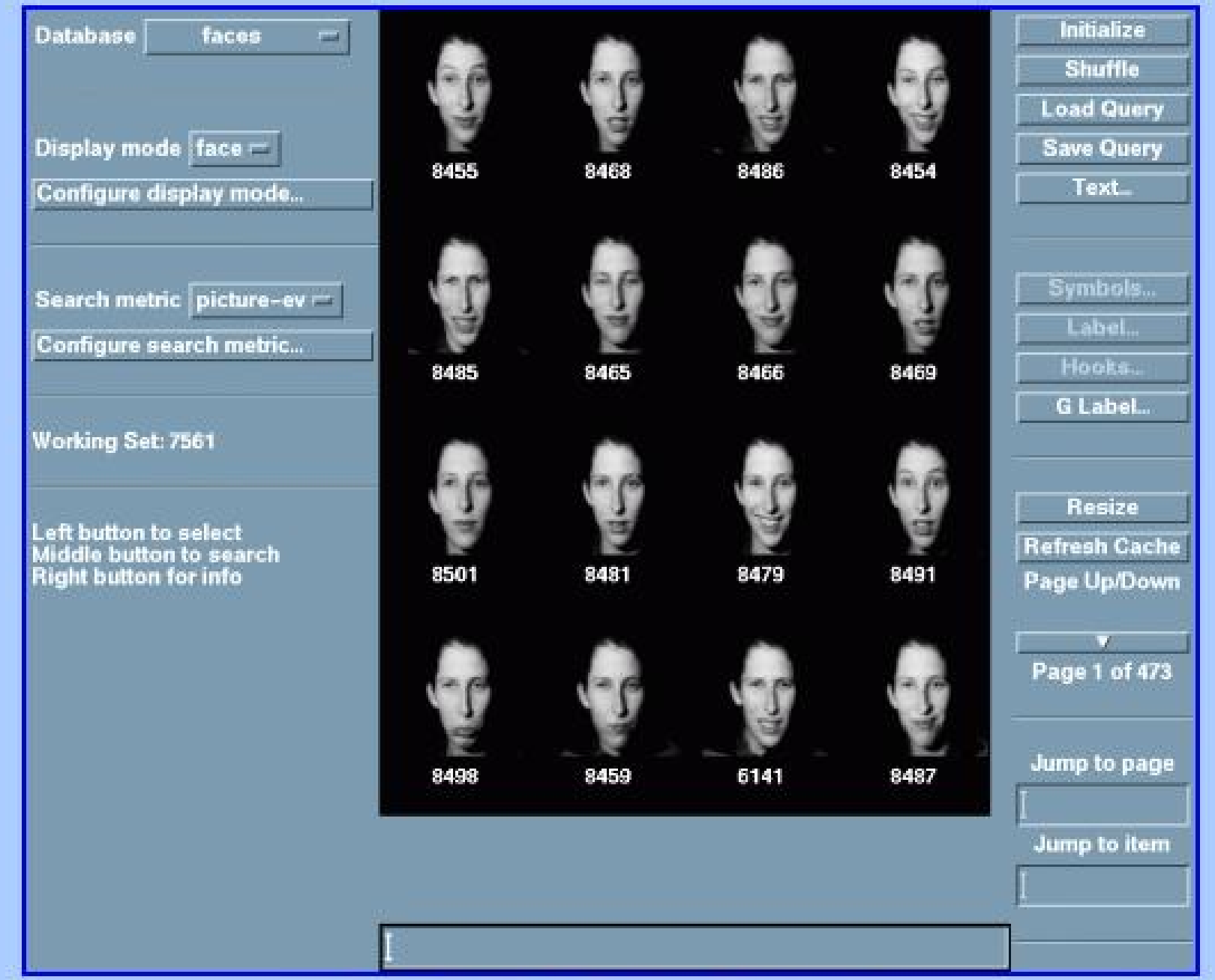


Standard Eigenfaces

Ähnlichkeitssuche basierend auf den Hauptkomponenten

- In folgenden 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

$$\|\mathbf{z} - \mathbf{z}_i\|$$



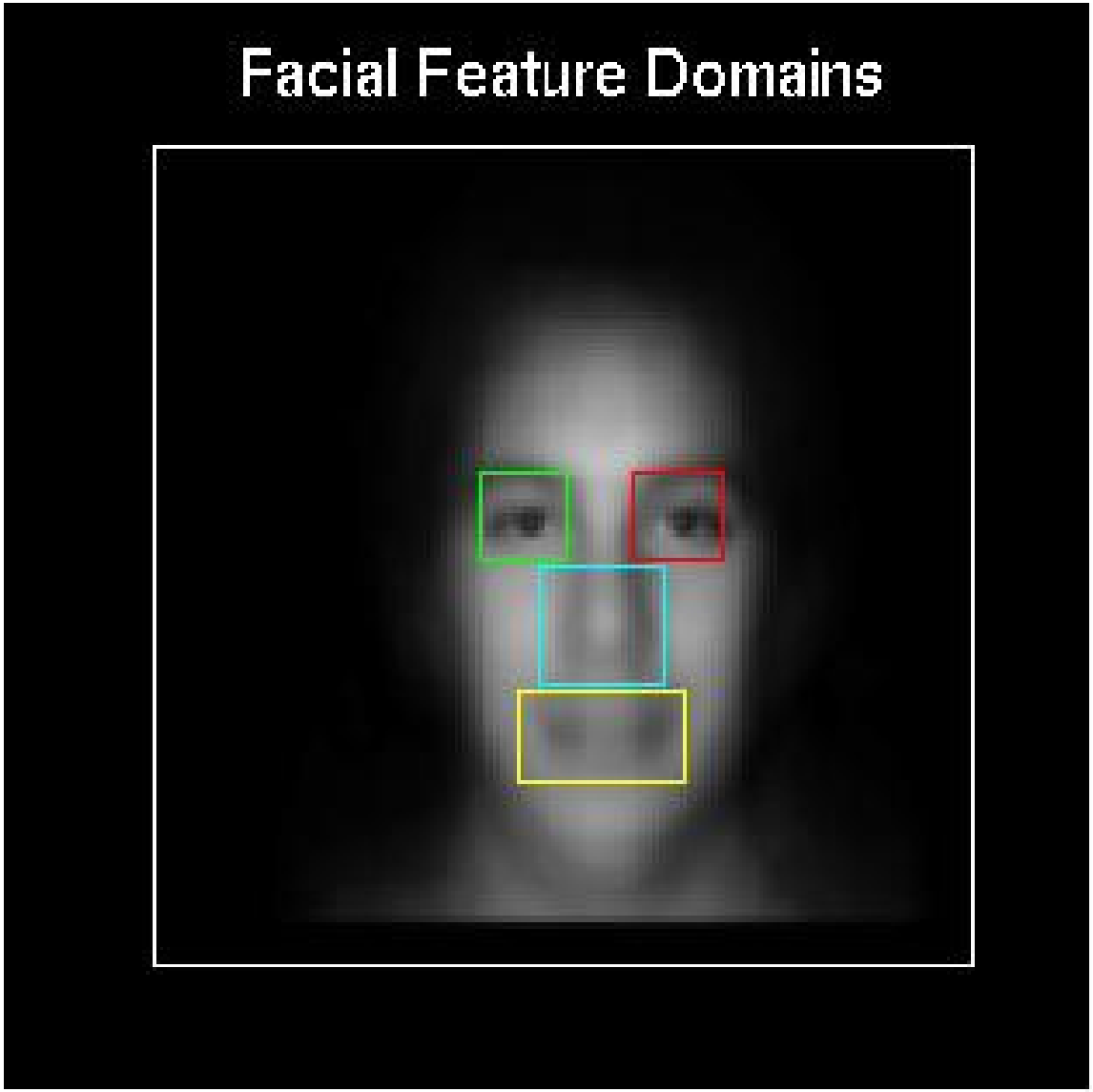
MIT Media Lab Database Photobook

Erkennungsrate

- 200 Bilder wurden zufällig ausgewählt und dem nächsten Nachbarn im Raum der Hauptkomponenten 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 Eigenmünder 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.

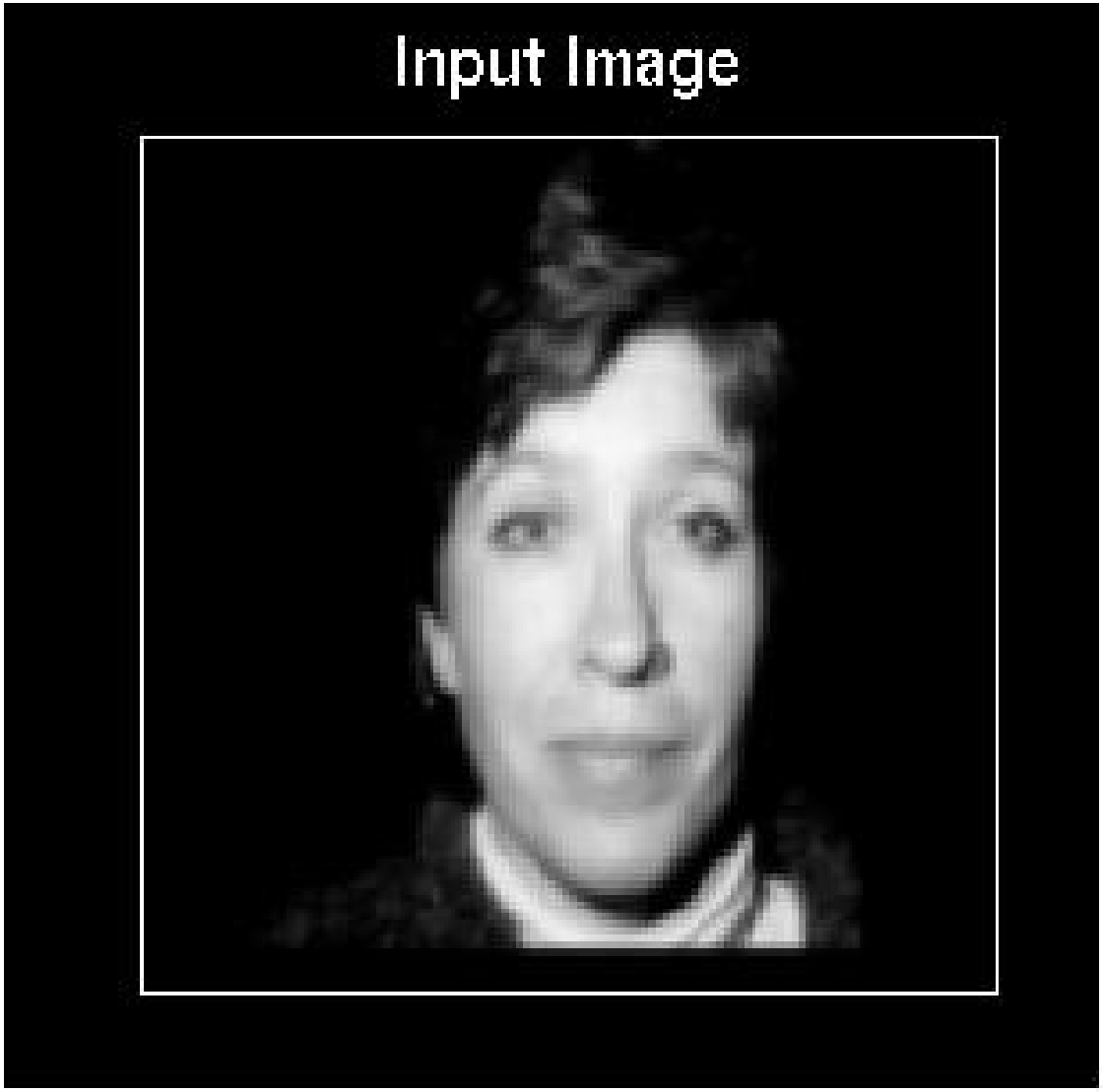

Facial Feature Domains

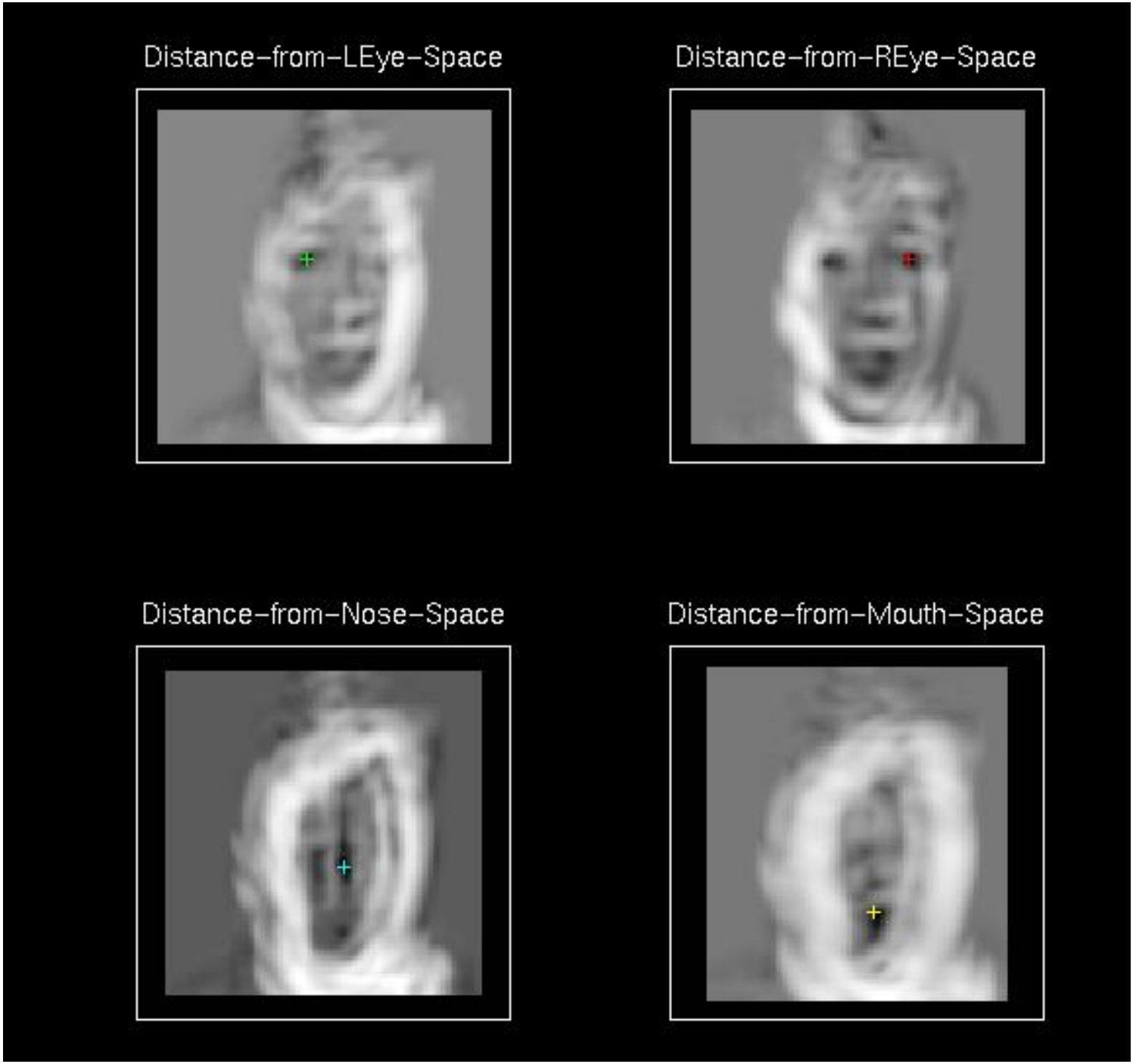
Automatische Detektion der Merkmale

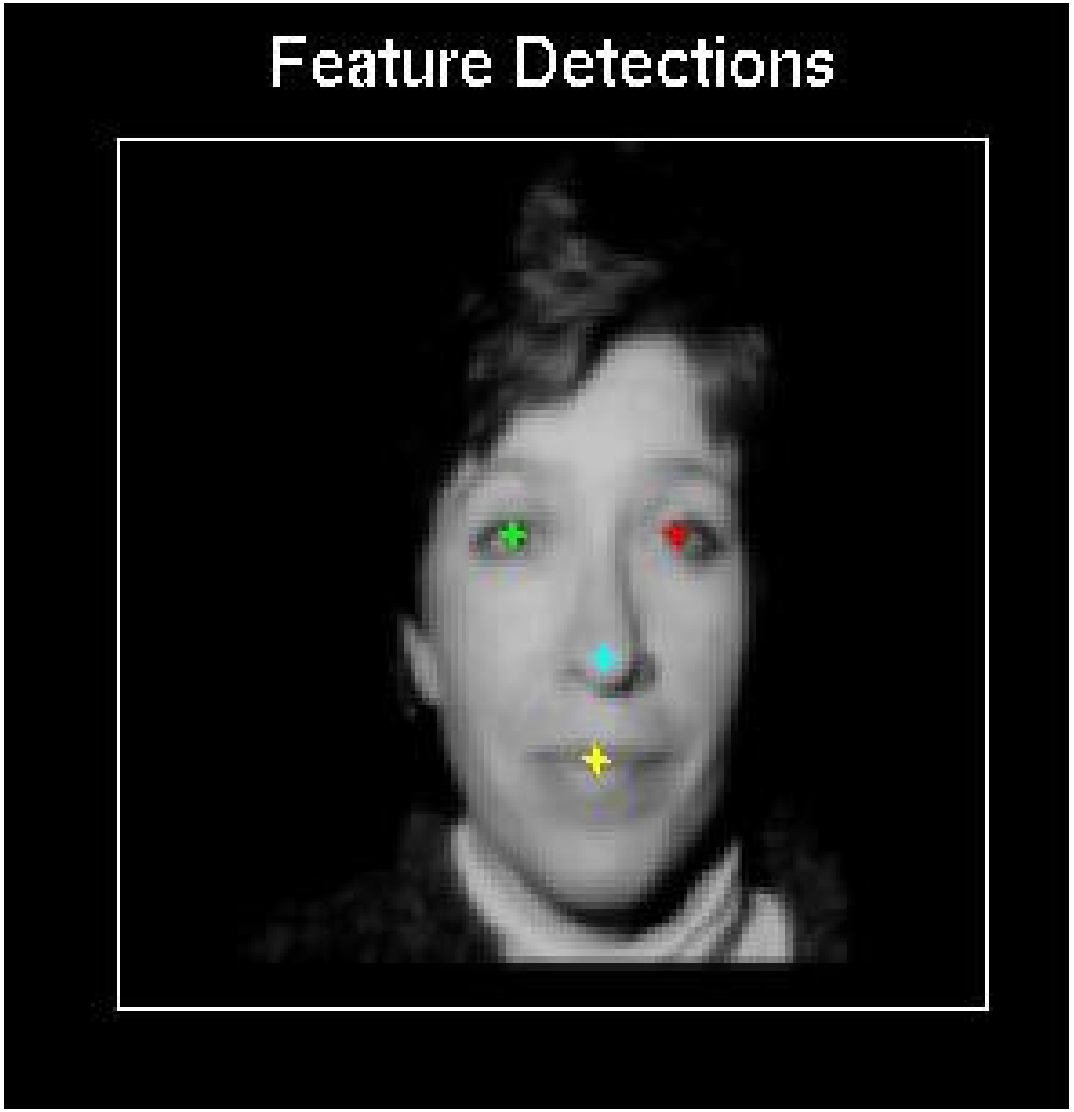
- Die modularen Methoden benötigen eine automatische Detektion der Merkmale (Augen, Nase, Mund)
- Man definiert ein rechteckiges Fenster, welches durch das zentrale Pixel im Rechteck indiziert wird
- Man berechnet im Fenster den quadratischen Abstand zwischen rekonstruiertem Bild und den roh-Pixelwerten im Fenster, basierend auf einer Hauptkomponentenanalyse die z.b. auf der Bildklasse linkes *Augen berechnet* wurde (AN: Anomalie)

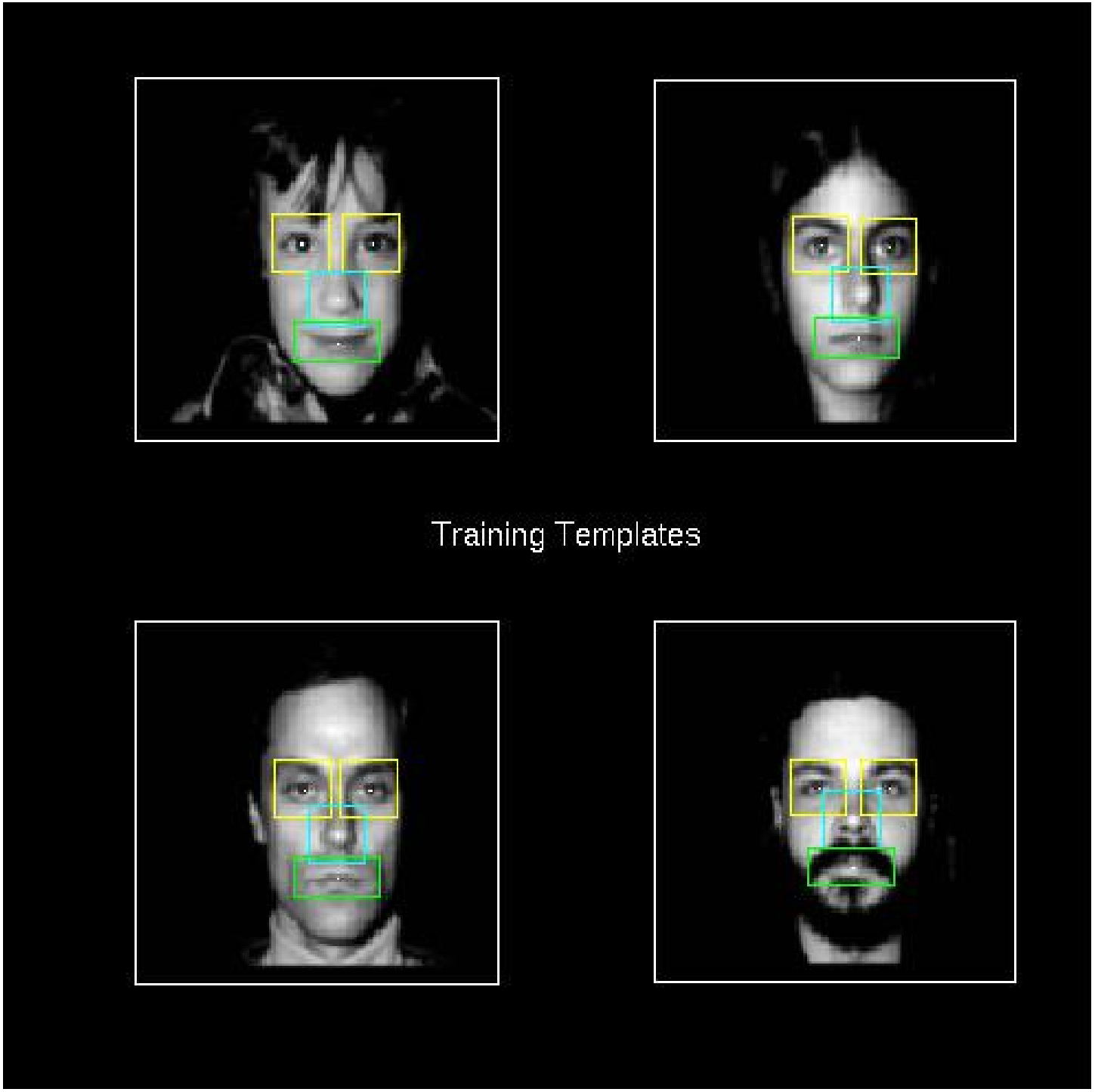
$$\mathrm{AN}_{\text{linkes Auge}}(\mathbf{z}_{pos_k}) = \|\mathbf{z}_{pos_k} - \hat{\mathbf{z}}_{pos_k}\|^2$$

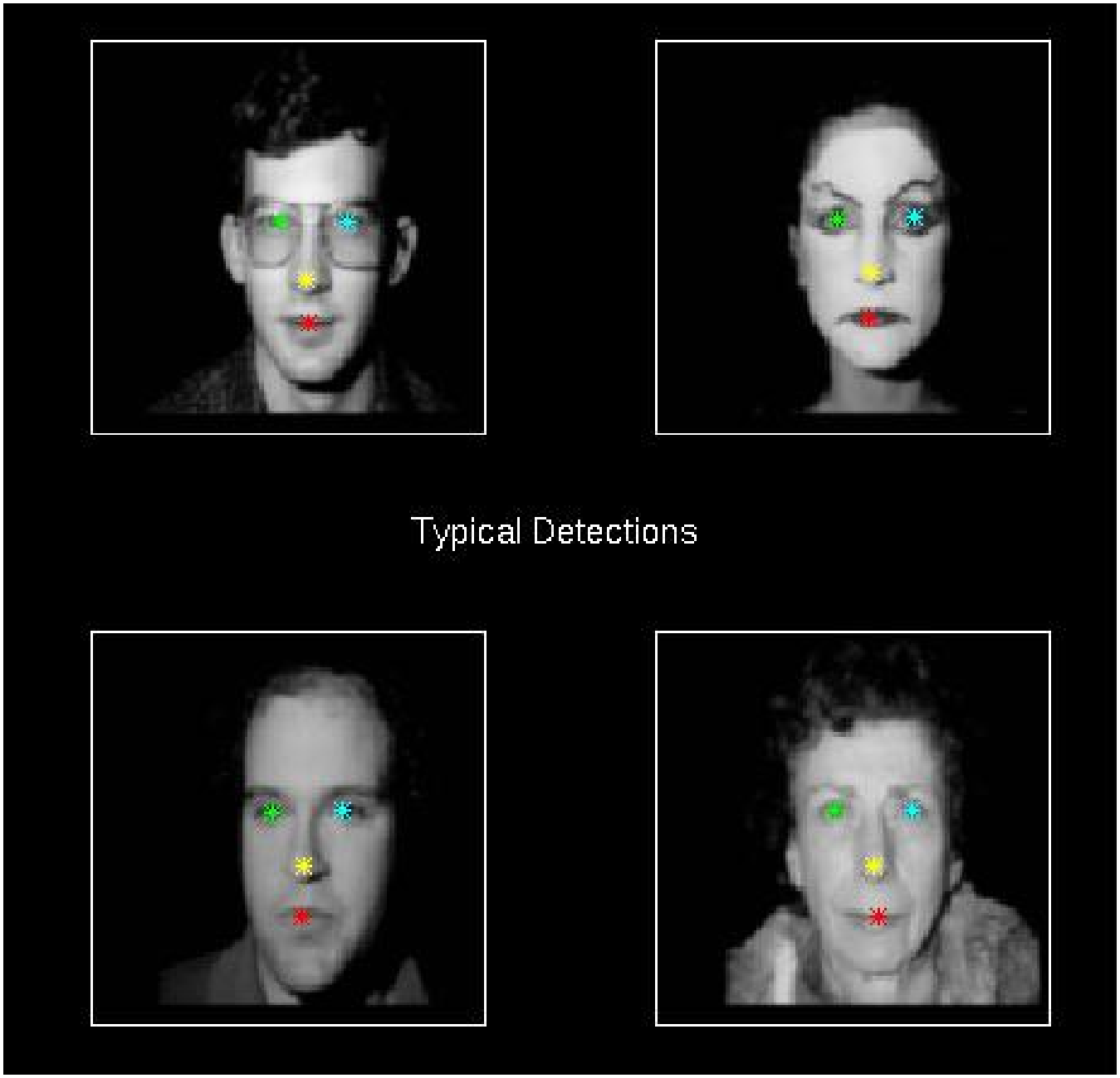
- $\mathrm{AN}_{\text{linkes Auge}}(\mathbf{z}_{pos_k})$ bewertet also, wie anomal der Bildausschnitt im Rechteck ist, falls es sich im Bildausschnitt um ein (beliebiges) linkes Auge handeln würde (Anomaliedetektion)
- Dei Anomalie wird für jedes mögliche Fenster berechnet; in den folgenden Bildern entspricht die Helligkeit der Anomalie; diese wird ebenso berechnet für rechtes Auge, Nase, Mund.


Input Image


Distance-from-LEye-Space
Distance-from-REye-Space
Distance-from-Nose-Space
Distance-from-Mouth-Space

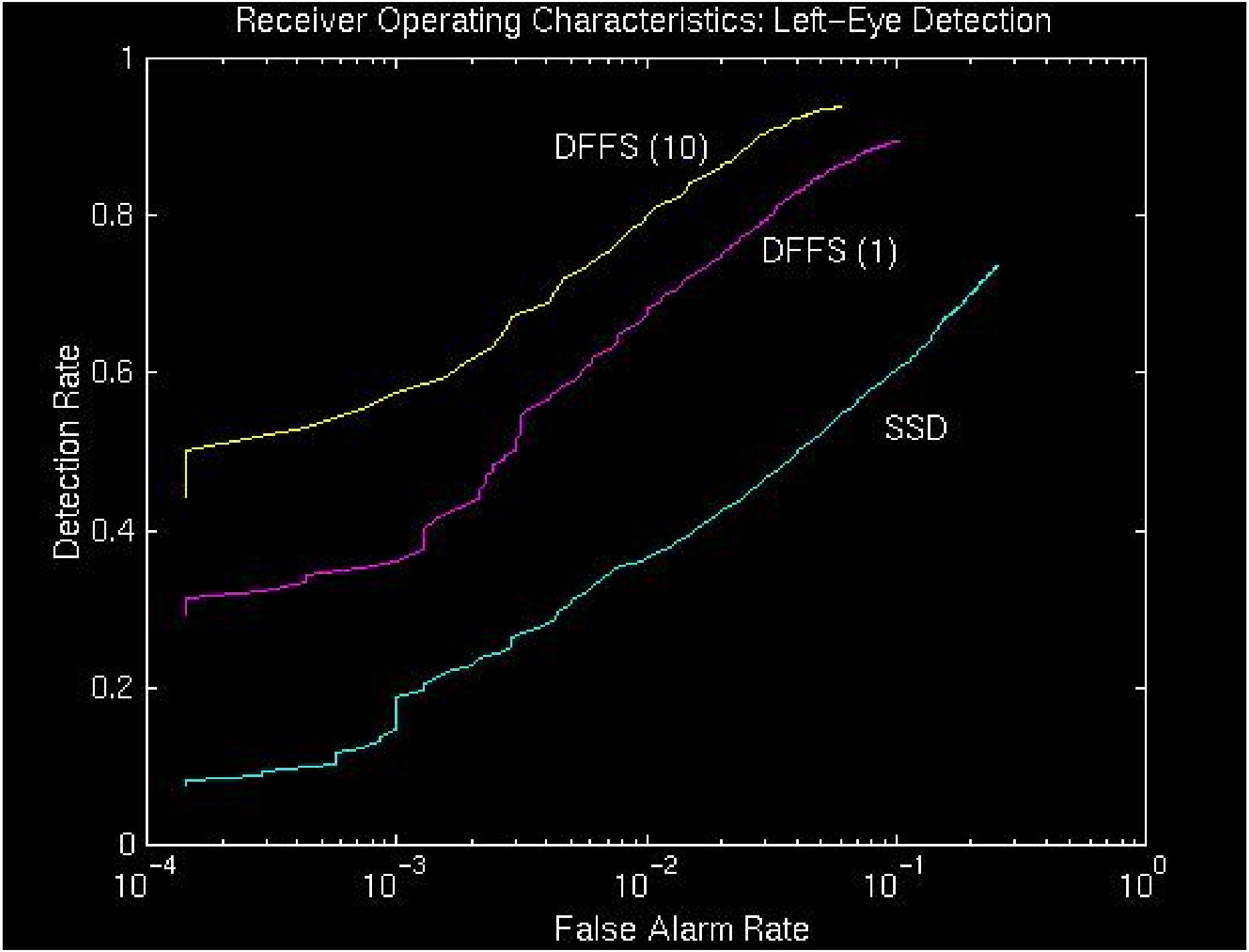

Feature Detections


Training Templates


Typical Detections

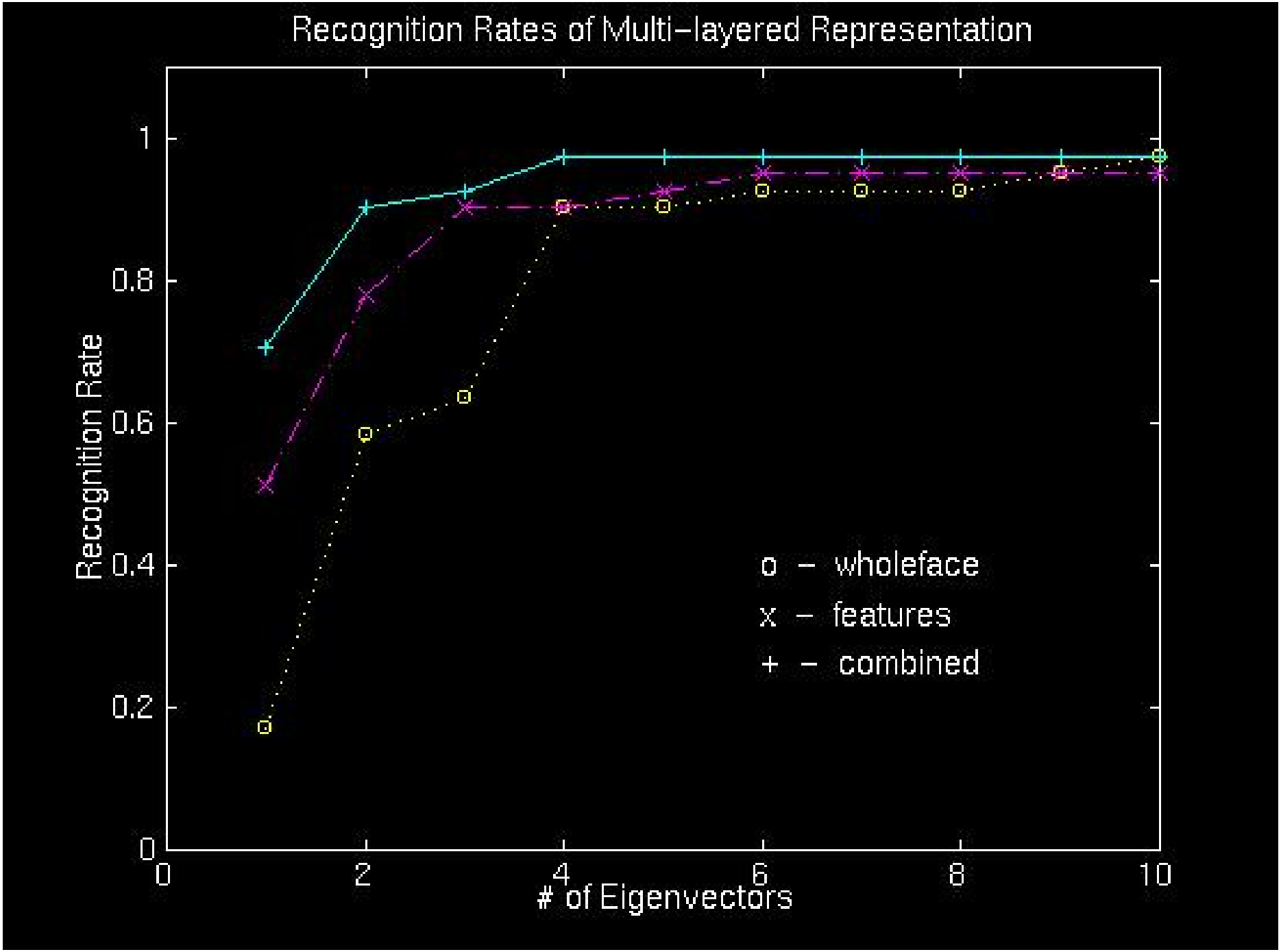
Detektionsrate

- Das Nächste Bild zeigt die Performanz der "linke-Auge-Detektion" basierend auf $\mathrm{AN}_{\text{linkes Auge}}(\mathbf{z}_{pos_k})$ (im Bild als DFFS bezeichnet) mit einem und 10 Eigenvektoren. Gezeigt ist ebenfalls die Performanz für ein einfaches Template matching (Abstand des Fensters zum Mittelwertbild eines Auges).
- **Korrekte Detektion:** das globale Minimum ist unter einem Schwellwert α und ist innerhalb von 5 Pixeln von der richtigen Lokalisiserung
- **Falscher Alarm:** das globale Minimum ist unter einem Schwellwert α und ist außerhalb von 5 Pixeln von der richtigen Lokalisiserung
- In den Kurven wird der Schwellwert α variiert. Mit einem entsprechenden Schwellwert erreicht DFFS(10) eine Detektionsrate von 94% und eine Falscher-Alarm-Rate von 6%. D.h. in 94% der Fälle, wo ein linkes Auge im Bild gefunden wird, ist es tatsächlich an der richtigen Stelle gefunden worden. In 6% der Fälle, wo kein linkes Auge im Bild gefunden wurde, ist dennoch eines vorhanden.


Receiver Operating Characteristics: Left−Eye Detection
Detection Rate
False Alarm Rate
DFFS (10)
DFFS (1)
SSD
1
0.8
0.6
0.4
0.2
0
10^-4
10^-3
10^-2
10^-1
10^0

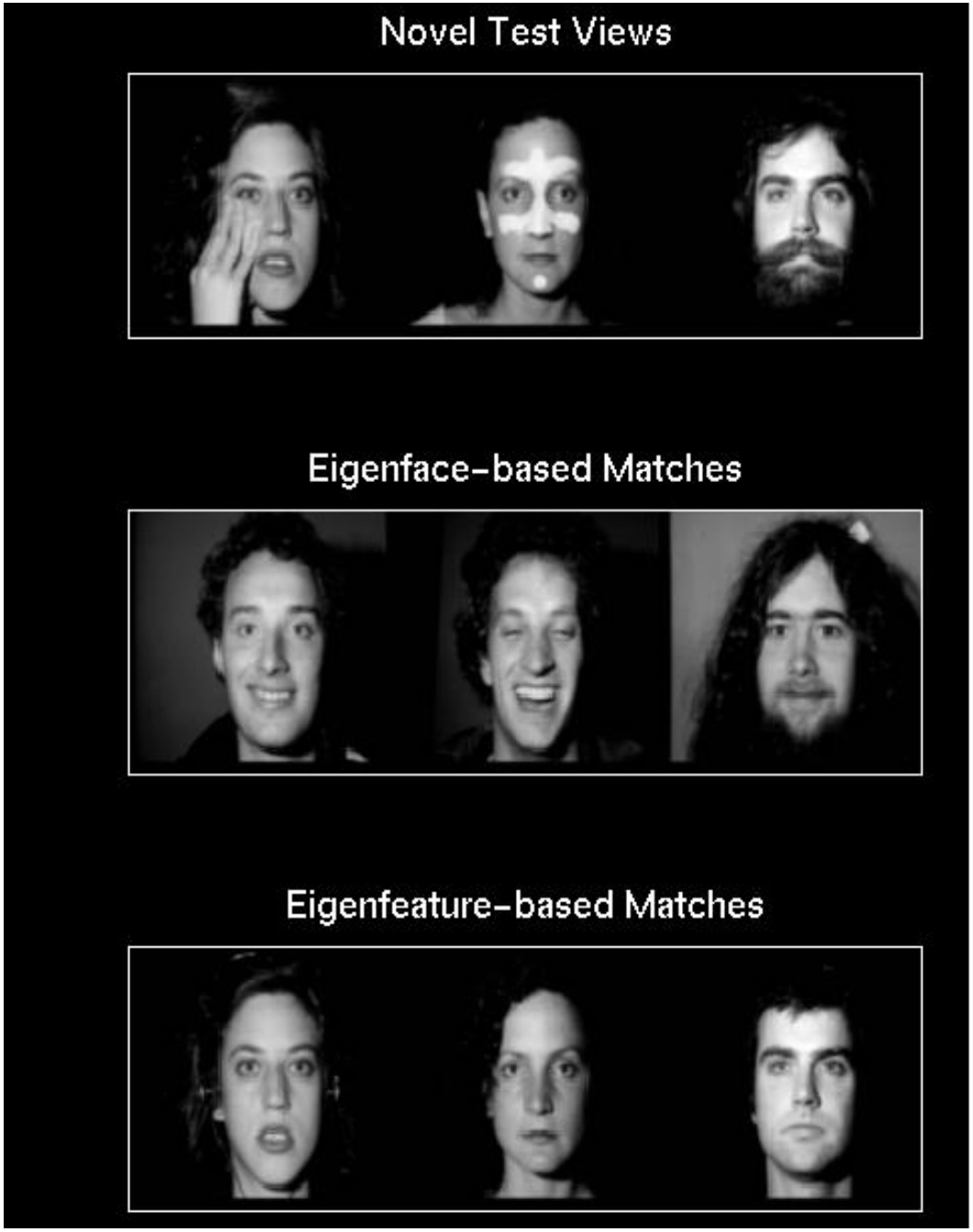
Modulare Erkennung

- 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 Trainingssdatensatz
- Die Zuordnung erfolgte entweder nur auf Basis der Repräsentation des ganzen Gesichtes, oder nur der Modularen Merkmale (Augen, Nase) oder beiden kombiniert
- Gezeigt ist die Erkennungsrate als Funktion von r


Recognition Rates of Multi-layered Representation
Recognition Rate
# of Eigenvectors
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10
o – wholeface
x – features
+ – combined

Robustheit

- 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

Ähnlichkeiten zwischen Dokumenten

Merkmalsvektoren für Dokumente

- Gegeben eine Sammlung von N Dokumenten und M möglichen Wörtern (eigentlich Wortstämme) im Dokument
- X: Die *Term-Frequency* (tf) Matrix; $x_{i,j}$ zeigt an, wie häufig das j-te Wort im Wörterbuch in Dokument i vorkommt
- Wenn in zwei Dokumenten nut verschiedene Wörter vorkommen, haben diese einen großen Abstand, auch wenn es sich inhaltlich betrachtet (semantisch) um ähnliche Texte handeln könnte
- Wie zuvor wird daher eine Hauptkomponentenanalyse durchgeführt und man berechnet Abstände in Bezug auf die Hauptkomponenten eines Dokumentes
- Die Analyse ist bekannt als *Latent Semantic Analysis* (LSA) oder auch *Latent Semantic Indexing* (LSA)

Einfaches Beispiel

- Insgesamt 9 Texte (Dokumente):
  - 5 Texte über Mensch-Maschine Interaktion (c1 - c5)
  - 4 Texte über mathematische Graph Theorie (m1 - m4)
- Die 12 Schlüsselwörter sind kursiv markiert

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*

tf-Matrix

- Die tf-Matrix X (leider wird hier mit X^T gearbeitet)
- Basierend auf X ist die Ähnlichkeit zwischen den Termen *human* und *user* geringer als zwischen den Termen *human* und *minors*

$$\{X\} =$$

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

r (human.user) = -.38

r (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

Singulärwertzerlegung

- Zerlegung $X = WSP^T$
- In unserer Notation:

$$X \to X^T$$

$$U \to W$$

(gezeigt sind nur die ersten 9 Spalten, die nicht mit dem Singulärwert Null multipliziert werden)

$$V \to P$$

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

$\{W\} =$

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

Approximation mit $r = 2$

- Rekonstruktion $\hat{X}$ mit $r = 2$
- Gezeigt ist wieder $\hat{X}^T$
- Basierend auf X ist die Ähnlichkeit zwischen den Termen *human* und *user* nun erheblich größer als zwischen den Termen *human* und *minors*
- Im Dokument $m4$: *Graph minors: a survey* wird das vorhandene Wort *survey* als geringer wahrscheinlich angesehen als das nicht vorhandene Wort *trees*

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

	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

r (human.user) = .94

r (human.minors) = -.83

Korrelation im Originalraum und in der Rekonstruktion

- Oben: Korrelation zwischen Dokumenten in X: Oft ist die Korrelation Null oder negativ zwischen Dokumenten in der c-Klasse
- Unten: Korrelation zwischen Dokumenten in $\widehat{X}$: Die korrelationen innerhalb der Klassen ist stark angewachsen und die Korrelationen zwischen beiden Klassen ist nun stark negativ

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

Anwendungen von LSA

- LSA-Ähnlichkeit entspricht oft der menschlichen semantischen Ähnlichkeitsempfindung
- Wird eingesetzt in einem kommerziellen Werkzeug zur Bewertung von Term-Papieren
- Es gibt Indikatoren, dass Suchmaschinenhersteller wie Google und Yahoo, PLA zum Ranken von Seiten einsetzen und um Spam zu filtern (Spam ist ungewöhnlich, anomal)

Weitere Anwendungen der Hauptkomponentenanalyse

- Beachte: LSA verwandte 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
- Spektrales Clustern und verwandte Algorithmen beruhen auf der Dimensionsreduktion mit der SVD und anschließendem Clustern in reduzierten Raum

Ein Wort der Warnung

- Die SVD wird manchmal leicht unterschiedlich definiert; z.B werden in einigen Definitionen Glieder Spalten in U mit Indices größer M sind gleich Null gesetzt. Dadurch sind einige Gleichungen manchmal leicht unterschiedlich. Z.B. sind die Zeilen von U nicht mehr orthonormal. Unsere Konvention entspricht z.B. der von Matlab.

Appendix

Interpretationen

I: Reduzierter Rang

- Beste Approximation durch Rang-Reduzierung der Design-Matrix

$$\hat{X} = U_r D_r V_r^T = U_r U_r^T X = X V_r V_r^T$$

- Die Approximation minimiert die Frobenius Norm

$$\sum_{i=1}^{N} \sum_{j=1}^{M} (x_{j,i} - \hat{x}_{j,i})^2 = \sum_{j=r+1}^{\tilde{r}} d_j^2$$

II. Datenpunktapproximation

- Wir wiederholen noch einmal

$$\hat{X} = XV_rV_r^T = Z_r^TV_r^T \quad \hat{X}^T = V_rV_r^TX^T = V_rZ_r$$

mit $Z_r = V_r^TX^T = D_rU_r^T$. Die i-te Spalte von Z_r enthält die r Hauptkomponenten von $\mathbf{x}_i$

- In dieser Interpretation sind die **Spalten** von V_r die Basisfunktionen zur Approximation der **Zeilen** von X also der Spalten von X^T (Datenpunktapproximation)

- Wir erhalten als Gewichte

$$W = (V_r^TV_r)^{-1}V_r^TX^T = Z_r$$

Wie erwartet: die i-te **Spalte** von Z_r enthält de Gewichte zur Approximation von $\mathbf{x}_i$, also der $i$ten Zeile von X.

Zur Approximation Diskreter Funktionen

- In einigen Fällen kann man einen Datenpunkt $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,M})^T$ als diskrete Funktion des Index j betrachten. Z.B. kann dann $\mathbf{x}_i$ eine Zeitreihe darstellen oder ein Bild (siehe Beispiel Gesichtserkennnung) und der Index j ist die diskrete Zeit oder ein Pixel-Index

- Hier kann man dann die Spalten von V_r als diskrete Eigenfunktionen des Index j betrachten; das heißt man lernt orthonormale Basisfunktionen!

- Die Dimensionsreduktion in der Transformation eines Signals auf die Hauptkomponenten wird zur Kompression eines Signals verwandt (siehe Karhunen-Loeve Transformation)

Spezialfall: Diskrete Fourier Transformation

- Wenn die Kovarianzmatrix gewisse Strukturen aufweist (invariant gegen Translation, genauer eine gewisse Bandstruktur in Form einer zyklischen Toeplitz Matrix) und entspricht die Transformation einer diskreten Fourier Transformation (DFT) mit Sinus- und Kosinusfunktionen als Eigenfunktionen
- Das heißt, $\mathbf{z}_i = V\mathbf{x}_i$ berechnet die DFT von $\mathbf{x}_i$ und $\mathbf{x}_i = V^T\mathbf{z}_i$ berechnet die inverse DFT.
- Wenn ich daher aufgrund meines Vorwissens annehmen kann, dass das Signal eine entsprechende Invarianz besitzt, kann ich mir die aufwändige Singulärwertzerlegung ersparen, da die Eigenfunktionen a priori bekannt sind. Die Berechnung der Hauptkomponenten kann über die sehr effiziente FFT (*Fast Fourier Transformation*) geschehen

Hauptkomponentenanalyse mit zentrierten Daten schätzt Zentrum und Hauptachsen der Kovarianz-Matrix einer Gauß-Verteilung

- Betrachten wir wieder die zentrierte SVD:

$$\tilde{X} = \tilde{U}\tilde{D}\tilde{V}^T$$

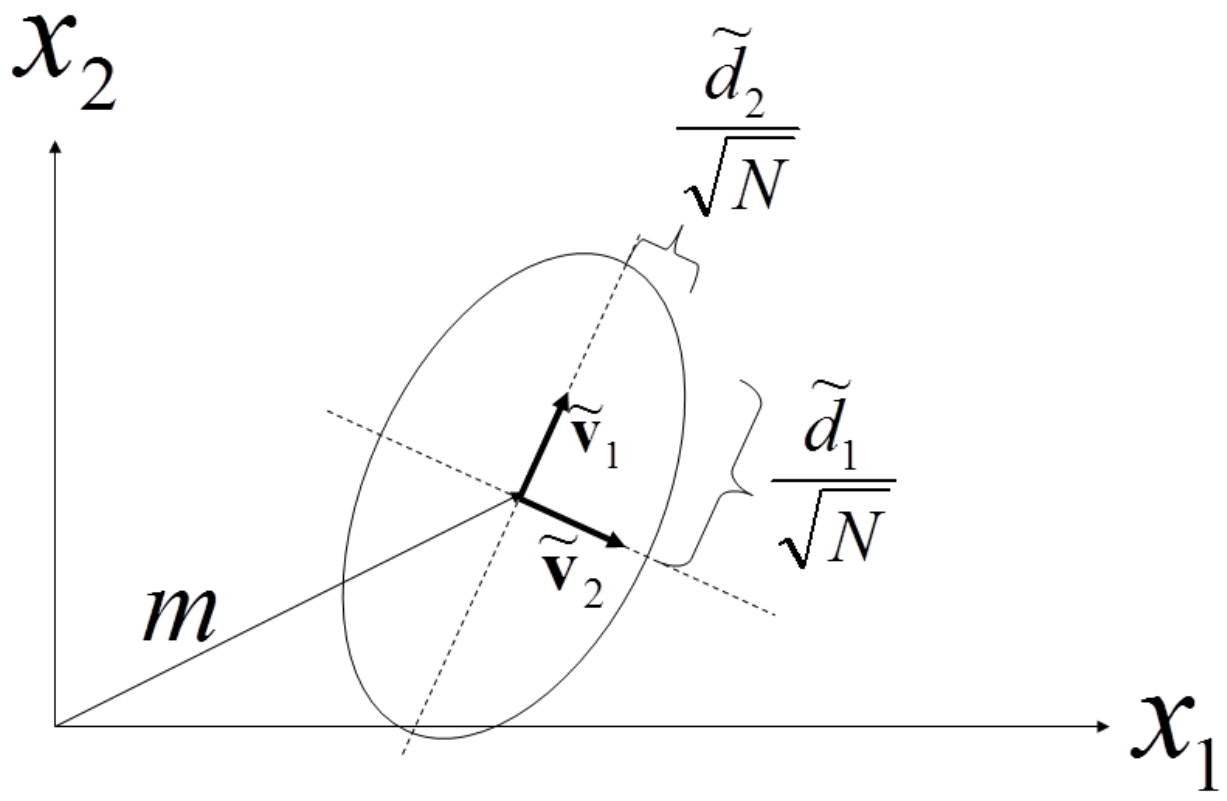
dann ist

$$\hat{\mathbf{x}}_i = \mathbf{m} + \sum_{l=1}^{r} \tilde{\mathbf{v}}_l \tilde{z}_{i,l}$$

mit $m = (m_1, \ldots, m_M)^T$

$$\tilde{z}_{i,l} = \tilde{\mathbf{v}}_l^T \tilde{\mathbf{x}}_i$$

- Sind die Daten Gauß-verteilt, entsprechen die Spalten von $\tilde{V}$ Schätzungen der Hauptachsen der Gauß-Verteilung


x_2
$\frac{\tilde{d}_2}{\sqrt{N}}$
$\tilde{\mathbf{v}}_1$
$\frac{\tilde{d}_1}{\sqrt{N}}$
$\tilde{\mathbf{v}}_2$
m
x_1

Probabilistic PCA (pPCA)

- Ein r-dimensionaler Zufallsvektor $\mathbf{e}_i$ wird generiert nach $\mathbf{e}_i \sim N(0, I)$
- Ein Datenpunkt wird generiert nach

$$\mathbf{x}_i = W\mathbf{e}_i + \mathbf{m} + \sigma^2 I$$

  wobei $\mathbf{m}$ ein Mittelwert ist und σ^2 eine Rauschvarianz.

- Eine Maximum-Likelihood Lösung ergibt sich zu

$$\hat{W} = \tilde{V}_r \text{diag}_r \left(\sqrt{\frac{1}{N}\tilde{d}_i^2 - \sigma^2} \right)$$

  (Beachte: die Lösung ist nicht eindeutig; z.B. $\hat{W}R$ mit Rotation R ist auch eine Lösung)

- Die Rekonstruktion wird

$$\hat{\mathbf{x}}_i = \tilde{V}_r \text{diag}_r \left(1 - \frac{\sigma^2}{\frac{1}{N}\tilde{d}_i^2} \right) \tilde{V}_r^T (\mathbf{x}_i - \mathbf{m}) + \mathbf{m}$$

- Beachte: mit $\sigma^2 \to 0$ ergibt sich die SVD Rekonstruktion

- Wir wiederholen noch einmal

$$\hat{X} = XV_rV_r^T = Z_r^TV_r^T \quad \hat{X}^T = V_rV_r^TX^T = V_rZ_r$$

mit $Z_r = V_r^TX^T = D_rU_r^T$. Die i-te Spalte von Z_r enthält die r Hauptkomponenten von $\mathbf{x}_i$

- Wir interpretieren wir die **Spalten** von Z_r^T als Basisfunktion zur Approximation der **Spalten** von X (Funktionsapproximation)

- Als LS-Gewichte ergibt sich

$$W = (Z_rZ_r^T)^{-1}Z_rX = (D_rU_rU_r^TD_r^T)^{-1}D_rU_r^TU_rD_rV_R^T = V_r^T$$

Das heißt, wie erwartet, die $j-$te **Zeile** von V_r enthält die Gewichte zur Approximation der j-ten Spalte von X

- Beachte, dass die Rolle von Z und V in dieser und der letzten Interpretationen vertauscht ist, was leicht zur Verwirrung führen kann!

Kern-Version

- Betrachten wir die Hauptkomponenten als Basisfunktionen ergibt sich der Kern

$$K_r = U D_r D_r^T U^T = X^T V_r V_r^T X$$

- Eine regularisierte Regression mit Regularisierungsparameter λ ist

$$\hat{X} = K_r(K_r + \lambda I)^{-1} X = U \text{ diag}(d_r^2/(d_r^2 + \lambda I)) \, U^T X$$

- Mit $\lambda \to 0$, erhält man $\hat{X} = U_r U_r^T X$

Flaschenhals

- $\widehat{\mathbf{x}}_i$ ist der Eingangsvektor. r lineare Systeme mit Gewichtsvektoren $\mathbf{v}_1, \ldots, \mathbf{v}_r$ transformieren den Eingangsvektor in eine reduzierte Repräsentation aus r Hauptkomponenten, nach

$$\mathbf{z}_i = V_r^T \mathbf{x}_i$$

  (Flaschenhals)

- Wir rekonstruieren

$$\widehat{\mathbf{x}}_i = V_r \mathbf{z}_i$$

- Interpretation als Funktionsapproximation: Die Knoten im Bottleneck repräsentieren die Basisfunktionen, die dann gewichtet summiert werden

- Interpretation als Datenpunktapproximation: Die Knoten im Bottleneck repräsentieren die Gewichtungen, mit denen dann die Basisvektoren gewichtet werden

Multivariate Modellierung

- Betrachten wir ein Problem mit M_x inputs und M Ausgängen; sei X die $N \times M_x$ Design Matrix der Eingangsvektoren und Y die $N \times M$ Matrix der Ausgangsvektoren
- Wenn wir jeden Ausgang getrennt Modellieren, $\widehat{y}_{i,j} = f_j(x_i)$, verlieren wir die Information, die in der Korrelation der Ausgangswerte liegt
- Ansatz 1: Wenn die Zusammenhänge zwischen Eingang und Ausgang linear sind, kann man eine Hauptkomponentenanalyse der Datenmatrix (X, Y) durchführen und alle Dimensionen über die Rekonstruktion bereinigen
- Ansatz 2: Man führt eine Hauptkomponentenanalyse nur von $Y = UDV^T$ durch; Dann kann man M Modelle $f_j(x_i)$ trainieren, bei denen die Zielgrößen die bereinigten Werte $\widehat{Y}$ sind

Multivariate Modellierung (fortgesetzt)

- Ansatz 3: Man führt eine Hauptkomponentenanalyse wieder nur von $Y = UDV^T$ durch; man trainiert r Modelle zur Vorhersage der $f_l^z(\mathbf{x}_i) = \widehat{z}_l(\mathbf{x}_i), l = 1, \ldots, r$ und kombiniert dann

$$\widehat{y}_{i,j} = \sum_{l=1}^{r} v_{i,l} f_l^z(\mathbf{x}_i)$$

  beachte, dass die Basisfunktionen für alle Dimensionen der Zielgröße gleich sind, aber dass sich die gelernten Gewichte unterscheiden für jede Dimension unterscheiden

- Die Vorteile der Ansätze 2 und 3 sind dass man auch nichtlineare Abhängigkeiten von Eingang zu Ausgang modellieren kann und dass man ein prädiktives Modell für neue Eingänge $\mathbf{x}$ erhält, bei denen keinerlei Zielgrößen vorliegen

IV: Produktform

- Die dritte Interpretation (I3) ist symmetrisch. Betrachten wir, dass man schreiben kann

$$\hat{X} = \left(U_r\sqrt{D_r}\right)\left(V_r\sqrt{D_r}\right)^T = \left(XV_r\sqrt{D_r^{-1}}\right)\left(X^TU_r\sqrt{D_r^{-1}}\right)^T$$

  Mit

$$\mathbf{a}_i = \sqrt{D_r^{-1}}V_r^T\left(x_{i,.}\right)^T$$

  und

$$\mathbf{b}_j = \sqrt{D_r^{-1}}U_r^T x_{.,j}$$

  ist

$$\hat{x}_{i,j} = \mathbf{a}_i^T\mathbf{b}_j$$

- $\mathbf{a}_i$ wird interpretiert als abgeleitete Attribute eines Datenpunktes (Zeile) i und $\mathbf{b}_i$ wird interpretiert als abgeleitete Attribute einer Spalte j.

- Dies stellt eine Lösung der folgenden Problemstellung dar:
- Sei A eine $N \times r$ Matrix A und B eine $M \times r$ Matrix. Die $N \times M$ Matrix X wird generiert nach

$$x_{i,j} = \sum_{l=1}^{r} a_{i,l} b_{j,l} + \epsilon_{i,j}$$

- $\epsilon_{i,j}$ ist i.i.d. Gauß-Rauschen
- Die log-Likelihood wird

$$\log L = const - \frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} (x_{i,j} - \widehat{x}_{i,j})^2$$

  mit $\widehat{X} = A^T B$

- Wenn nur X bekannt ist, ist die Zerlegung nicht eindeutig
- Eine Lösung ist, die oben angegebene

Vergleich zur Probabilistische PCA

- Das betrachtete Modell ist

$$\prod_i \prod_j N(x_{i,j}; (A^T B)_{i,j}, \sigma^2)$$

- pPCA betrachtet

$$\prod_i N(\mathbf{x}_i; W\mathbf{e}_i, \sigma^2 I) N(\mathbf{e}_i; 0, \sigma^2 I)$$

- Gegeben A, B sind im ersten Modell alle $x_{i,j}$ unabhängig. Im Gegensatz sind im zweiten Modell die Komponenten von $\mathbf{x}_i$ korreliert, da $\mathbf{e}_i$ als zufällig angenommen wird. Nur gegeben $\mathbf{e}_i$ sind im zweiten Modell die Komponenten unabhängig
- Faktor Analyse: Wie pPCA, nur dass die Rauschvarianz Komponentenabhängig sein kann

Sonstiges

Eigenwertzerlegung

- Sei X eine quadratische $N \times N$ Matrix mit N linear unabhängigen Eigenvektoren und der Eigenzerlegung

$$QX = Q\Lambda$$

  Hier ist Λ die Diagonalmatrix der Eigenwerte; die Spalten von Q enthalten die (in der Regel nicht orthogonalen) Eigenvektoren

- Dann gilt die Zerlegung

$$X = Q\Lambda Q^{-1}$$

- Wenn zusätzlich X symmetrisch ist, dann sind Hauptkomponentenanalyse und Eigenwertzerlegung identisch

- Google's Page Rank basiert auf einer Eigenwertzerlegung der Verbindungsmatrix, während Kleinbergs Hyperlink-Induced Topic Search (HITS) auf einer SVD der Verbindungsmatrix beruht

Kern PCA

- Wir betrachten den Fall, dass man nicht im Merkmalsraum arbeiten möchte, weil dieser z.B. zu hoch dimensional ist, $M >> 1$
- Bekannt ist jedoch das innere Produkt der Merkmalsvektoren $k(x_i, x_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Wir wollen nun nach $\mathbf{z}_i = V_r^T \mathbf{x}_i$ die Hauptkomponenten nicht einfach berechnen, da $\mathbf{x}_i$ zu hoch dimensional ist und sich V_r sich nur aus der unbekannten Kovarianzmatrix berechnen lässt
- Aus $X = UDV^T$ folgt jedoch, dass (D_r^+ ist die Transponierte von D_r, bei der Singulärwerte ungleich Null durch das Inverse ersetzt wurden)

$$V_r^T = D_r^+ U_r^T X$$

Somit

$$V_r^T \mathbf{x}_i = D_r^+ U_r^T X \mathbf{x}_i = D_r^+ U_r^T k(., \mathbf{x}_i)$$

- Das heißt, wir können die Hauptkomponenten über eine Zerlegung der $N \times N$ Matrix $K = UDD^T U^T$ berechnen

Zentrierte Kern Hauptkomponentenanalyse

- Auch die Hauptkomponenten von zentrierten Daten lässt sich berechnen. Dazu benötigen wir

$$\tilde{K} = (X - \frac{1}{N}\mathbf{1}\mathbf{m}^T)(X - \frac{1}{N}\mathbf{1}\mathbf{m}^T)^T$$

$$= XX^T + \frac{1}{N^2}\mathbf{1}\mathbf{m}^T\mathbf{m}\mathbf{1}^T - \frac{1}{N}X\mathbf{m}\mathbf{1}^T - \frac{1}{N}\mathbf{1}\mathbf{m}^T X^T$$

Hier ist $\mathbf{1}$ ein N-dimensionaler Spaltenvektor aus Einsen. Wir können auch schreiben $\mathbf{m} = \frac{1}{N}X^T\mathbf{1}$ und erhalten

$$\tilde{K} = +\frac{1}{N^4}EKE - \frac{1}{N^2}KE - \frac{1}{N^2}EK$$

wobei E eine $N \times N$ Matrix von Einsen ist

Nyström Approximation

- Nehmen wir an, dass eine Kernfunktion $k(\mathbf{x}_i, \mathbf{x}_j)$ gegeben ist. Die Kern matrix $K = UDD^TU^T$ enthält endlich viele Realisierungen von $k(\mathbf{x}_i, \mathbf{x}_j)$.

- Wenn $\mathbf{x}_i$ and $\mathbf{x}_j$ Einträge in K darstellen gilt

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{u}_i^T DD^T \mathbf{u}_j = \mathbf{z}_i^T \mathbf{z}_j$$

- Für beliebige $\mathbf{x}_i$ und $\mathbf{x}_j$ ist das eine Approximation und entspricht der Berechnung der Hauptkomponenten für neue Daten. Dann kann man weiter umformen mit den Kern PCA Gleichungen

$$k(\mathbf{x}_i, \mathbf{x}_j) \approx k^T(., \mathbf{x}_i) UD^+ \ (D^+)^T U^T k(., \mathbf{x}_j) = k^T(., \mathbf{x}_i) K^{-1} k(., \mathbf{x}_j)$$

- Betrachten wir eine hoch dimensionale $L \times L$ Kernmatrix K_G, dann ergibt sich die Zerlegung

$$K_G \approx K^{*T} K^{-1} K^*$$

wobei K^* eine $L \times N$ matrix ist mit den Kernfunktionen zwischen den Daten in K_G und K. Diese Zerlegung kann verwandt werden, um approximative Inverse zu K_G zu berechnen

If we we use a rank r approximation we have

$$K_G \approx {K^*}^T U D^+ \ (D^+)^T U^T K^*$$

Hauptkomponentenanalyse, Clustern, Spectral Clustering

- Anstatt im Originalraum mit Datenpunkten $\mathbf{x}_i$ zu clustern macht es Sinn, im Raum der Hauptkomponenten

$$\mathbf{z}_i = V^T \mathbf{x}_i$$

  die Clusterung durchzuführen

- Wenn wieder nicht die Merkmale sondern die Kernmatrix K bekannt sind, berechnet man

$$\mathbf{z}_i = D_r^+ U_r^T k(., \mathbf{x}_i)$$

- Wenn man $k(., \mathbf{x}_i)$ als Ähnlichkeitsfunktion interpretiert, verwendet man den Begriff *Spectral Clustering*

- *Spectral Clustering* (SC) wird häufig verwendet, wenn Datenpunkte durch Knoten im Graphen definiert werden und die Ähnlichkeitsfunktion die Ähnlichkeit von Knoten im Graphen beschreibt. Anstatt die Singulärvektoren von K zu arbeiten, arbeitet man mit $K - I$ oder $K - D^{(K)}$, wobei $D^{(K}$ eine Diagnonalmatrix ist mit $d_{ii}^k = sum_j w_{i,j}$.

Die Eigenvektoren von $K - I$ sind die selben wie die von K, die Eigenvektoren von $K - D^{(K)}$ sind anders

- Genaugenommen arbeitet SC äquivalenterweise mit den Eigenvektoren mit den kleinsten Eigenwerten von $L = I - K$, bzw. von $D^{(K)} - K$. Hier sind Eigenvektoren und Singulärvektoren identisch (eventuell bis auf Vorzeichen).

Eindeutigkeit?

- Wie beschrieben sind SVD und Hauptkomponentenanalyse eindeutig; dies begründet sich aus der Forderung, dass die Rekonstruktion für jedes r optimal sein muss
- Fordert man hingegen nur, dass man die beste Rang-r Approximation findet für ein festes r wird die Lösung uneindeitig, denn mit einer beliebigen Rotationsmatrix R kann man schreiben

$$\widehat{X} = XV_rV_r^T = (XV_rR)(V_rR)^T$$

mit neuer orthogonaler Basis V_rR und neuen Hauptkomponenten XV_rR