

# 6. Karhunen-Loève-Transformation (KLT)

6.1 Definition und Eigenschaften

6.2 Anwendungsbeispiele

6.3 Approximation durch Cosinus-Transformation

## 6.1 Definition und Eigenschaften

- **Idee:** Approximation diskreter Zufallsprozesse  $\mathbf{x} = [x_1, \dots, x_N]^T$  basierend auf den Eigenvektoren ihrer Kovarianzmatrix
- **Anwendungen:** Signalanalyse (Reduktion auf wesentliche Merkmale), effektive Signalcodierung bei der Datenübertragung/-speicherung etc.
- **Gegeben:** diskreter, **mittelwertfreier** Zufallsprozess  $\mathbf{x} - \boldsymbol{\mu}$  mit  $\boldsymbol{\mu} = E\{\mathbf{x}\}$
- **Gesucht:** Approximation von  $\mathbf{x} - \boldsymbol{\mu}$  durch eine zunächst unbekannte **orthonormale** Basis  $\Phi = [\varphi_1, \dots, \varphi_N]$ ,  $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$ ,  $\Phi^T \Phi^* = \mathbf{I}$ :

$$\mathbf{x} - \boldsymbol{\mu} = \Phi \cdot \mathbf{b}$$

- Aufgrund der **Orthonormalität** der Basis werden die Koeffizienten durch Projektion berechnet:

$$b_i = \langle \mathbf{x} - \boldsymbol{\mu}, \varphi_i \rangle = (\mathbf{x} - \boldsymbol{\mu})^T \varphi_i^* = \varphi_i^H (\mathbf{x} - \boldsymbol{\mu})$$

$$\Rightarrow \mathbf{b} = \Phi^H \cdot (\mathbf{x} - \boldsymbol{\mu}) \quad \text{mit} \quad \Phi^H = \Phi^{T*}$$

KLT

### Forderung

- Koeffizienten der KLT sollen paarweise **unkorreliert** sein (d. h. Kovarianzmatrix des Koeffizientenvektors ist eine Diagonalmatrix)

$$E\{b_i b_j^*\} = E\{\varphi_i^H (\mathbf{x} - \boldsymbol{\mu}) (\mathbf{x} - \boldsymbol{\mu})^H \varphi_j\} = \varphi_i^H \mathbf{C}_{xx} \varphi_j \stackrel{!}{=} \lambda_j \delta_{ij}$$

$$\Leftrightarrow \mathbf{C}_{xx} \varphi_j = \lambda_j \varphi_j$$

- **Eigenvektoren der Kovarianzmatrix** entsprechen den gesuchten Basisfunktionen; die  $\lambda_j$  sind die zugehörigen Eigenwerte

### Eigenschaften

- Erwartungswert der Signalenergie ist gleich der Summe aller Eigenwerte:

$$E\{\|\mathbf{x} - \boldsymbol{\mu}\|^2\} = E\{\|\Phi \mathbf{b}\|^2\} = E\left\{\left(\sum_i b_i \varphi_i\right)^T \left(\sum_j b_j \varphi_j\right)\right\}$$

$$= \sum_i \sum_j E\{b_i b_j\} \underbrace{\varphi_i^T \varphi_j}_{\delta_{ij}} = \sum_i E\{b_i^2\} = \sum_i \lambda_i$$

- Gute Signalkompression durch Berücksichtigung der größten Eigenwerte

- $\mathbf{C}_{xx}$  symmetrisch  $\Rightarrow$   $N$  reelle, nicht-negative Eigenwerte  $\lambda_j \geq 0$

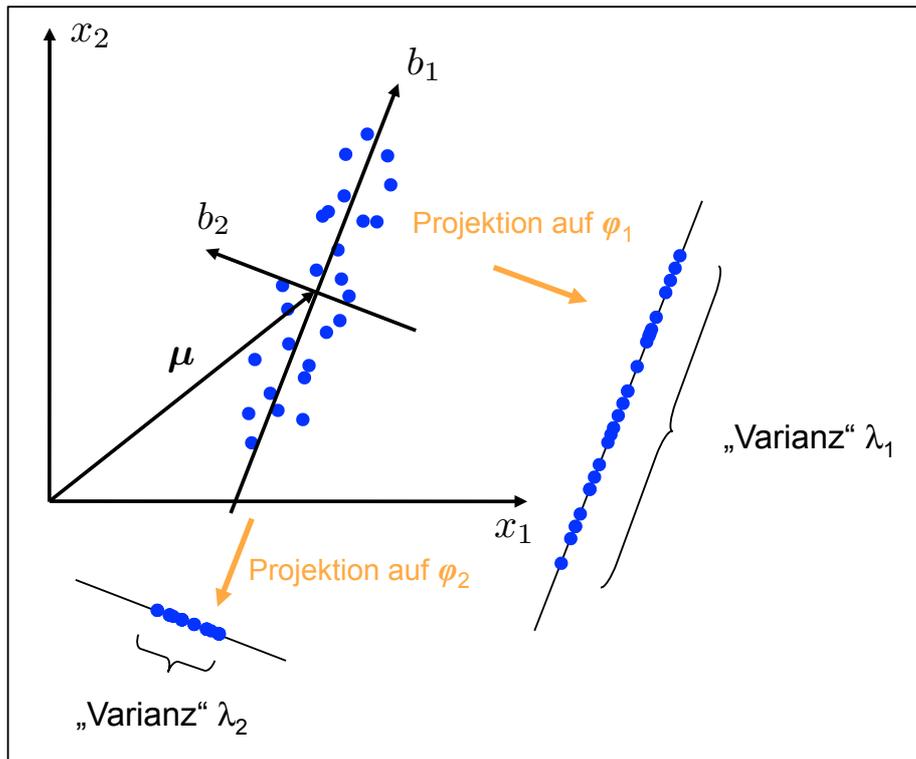
- Approximation von  $\mathbf{x}$  durch  $m < N$  Basisvektoren mit größtem Eigenwert  $\rightarrow$  Signalenergie des Fehlers dadurch minimal:

$$E\left\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right\} = E\left\{\left\|\sum_{i=m+1}^N b_i \varphi_i\right\|^2\right\} = \sum_{i=m+1}^N \lambda_i$$

### Diskussion

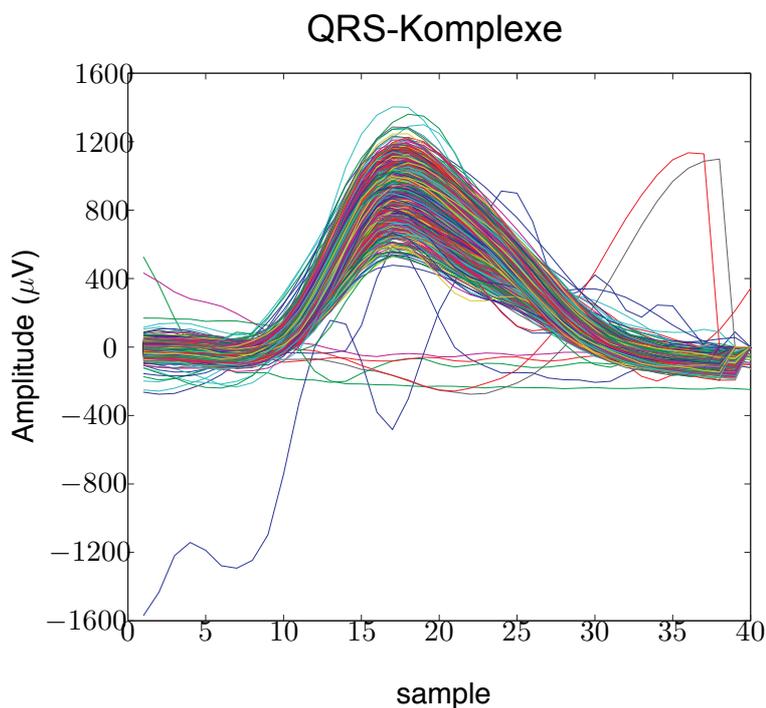
- Approximation entspricht einer linearen Projektion auf einen Unterraum geringerer Dimension
- Dabei liefert die KLT die Projektion, die einen gegebenen Zufallsvektor  $\mathbf{x}$  im Sinne der kleinsten Fehlerquadrate **optimal** repräsentiert
- Koeffizienten sind paarweise **unkorreliert**
- Varianzen  $\lambda_i$  der Koeffizienten  $b_i$  sind **maximal ungleichverteilt**: minimale Entropie der Varianzen

## Veranschaulichung der Karhunen-Loève-Transformation



# 6.2 Anwendungsbeispiele

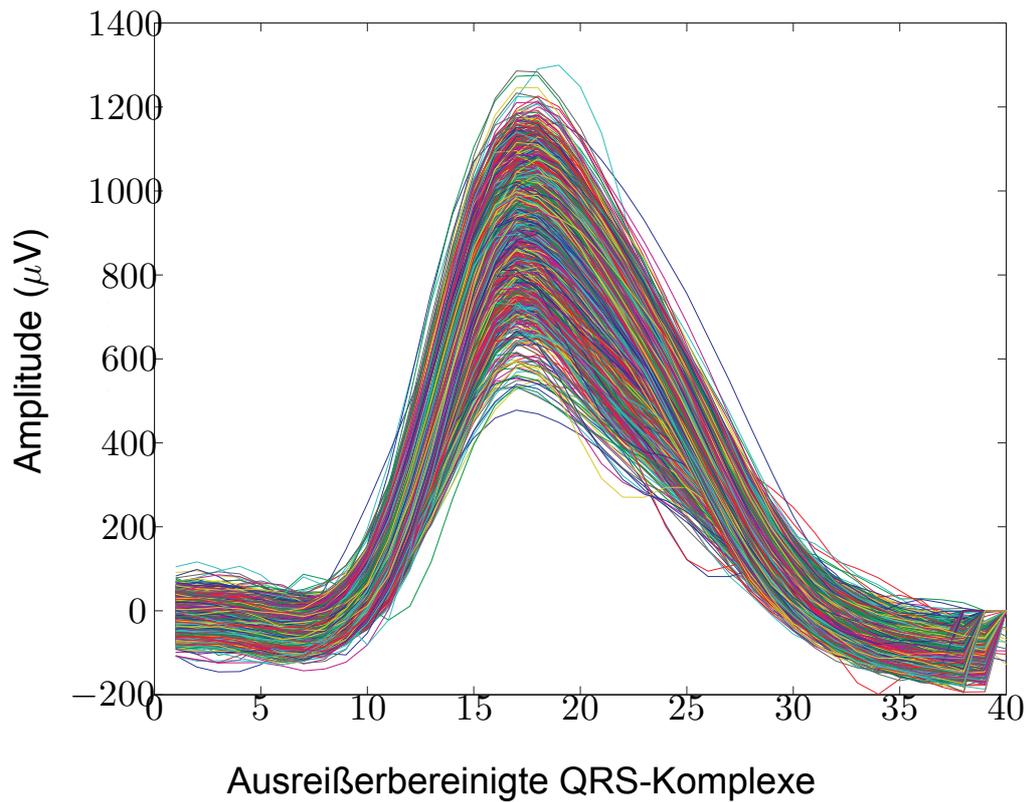
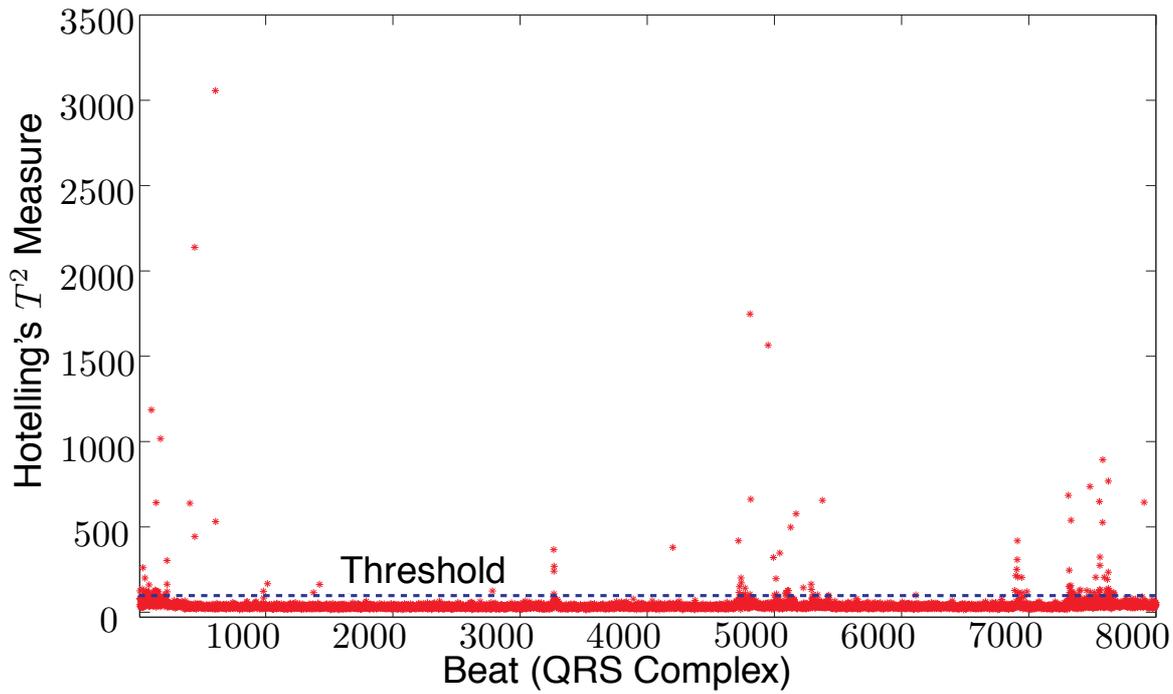
## Beispiel: Erkennung von Ausreißern in der automat. EKG-Auswertung



### Vorgehensweise bei der Hauptkomponentenanalyse (PCA)

- PCA (*Principal Component Analysis*): empirische Variante der KLT
- Einzelnes Signal:  $\mathbf{x}_m = [x_{1m}, \dots, x_{Nm}]^T, \quad m = 1, \dots, M$
- Mittelwertvektor:  $\bar{\mathbf{x}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m$
- Spaltenvektoren als Matrix:  $\mathbf{Z} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_M - \bar{\mathbf{x}}]$
- Kovarianz- bzw. Streumatrix:  $\mathbf{C}_{xx} = \text{E}\{\mathbf{Z}\mathbf{Z}^H\} \approx \frac{1}{M} \mathbf{Z}\mathbf{Z}^H = \frac{1}{M} \sum_m \mathbf{z}_m \mathbf{z}_m^H$
- Lösung des Eigenwertproblems: Berechnung der Eigenwerte  $\lambda_1, \dots, \lambda_N$  und der Eigenvektoren  $\Phi = [\varphi_1, \dots, \varphi_N]$
- Sortierung nach der Größe der Eigenwerte

- KLT:  $\mathbf{B} = \Phi^H \cdot \mathbf{Z} = [\mathbf{b}_1, \dots, \mathbf{b}_M] = \begin{bmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NM} \end{bmatrix}$
- Da die Eigenwerte  $\lambda_n$  der Kovarianzmatrix sortiert wurden, konvergieren sie für große  $n$  gegen null:  $\lambda_1 \gg \lambda_N$
- Für ein Signal  $\mathbf{z}_m$ , das der Statistik der Mehrheit aller Signale folgt, konvergieren die Koeffizienten  $b_{nm}$  für große  $n$  ebenfalls gegen null
- Für Signale, deren Statistik sich von der der Mehrheit aller Signale unterscheidet (Ausreißer), konvergieren die Koeffizienten dagegen nicht
- Hotelling- $T^2$ -Maß:  $T_m^2 = \sum_{n=1}^N \frac{b_{nm}^2}{\text{E}\{b_{nm}^2\}} = \sum_{n=1}^N \frac{b_{nm}^2}{\lambda_n} \quad (\text{Gleichrichtung})$
- $T_m^2$  groß für Ausreißer wegen Gewichtung mit  $1/\lambda_n \rightarrow$  gut für Detektion!



### Beispiel: Gesichtserkennung mit Eigengesichtern (*eigenfaces*)

Grauwertbild eines Gesichtes:  $g_m(x, y), \quad m = 1, \dots, M$

Bild als Spaltenvektor:  $\mathbf{x}_m := \text{Row}\{g(x, y)\} \in \mathbb{R}^{N^2}$

Datenmatrix:  $\mathbf{Z} := [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_M - \bar{\mathbf{x}}], \quad \bar{\mathbf{x}} = \frac{1}{M} \sum_i \mathbf{x}_i$

Kovarianz- bzw. Streumatrix:  $\mathbf{C}_{zz} = E\{\mathbf{Z}\mathbf{Z}^H\} \approx \frac{1}{M} \mathbf{Z}\mathbf{Z}^H = \frac{1}{M} \sum \mathbf{z}_m \mathbf{z}_m^H$   
 $N^2 \times N^2$ -Matrix  $\rightarrow$  Eigenvektoren praktisch nicht berechenbar

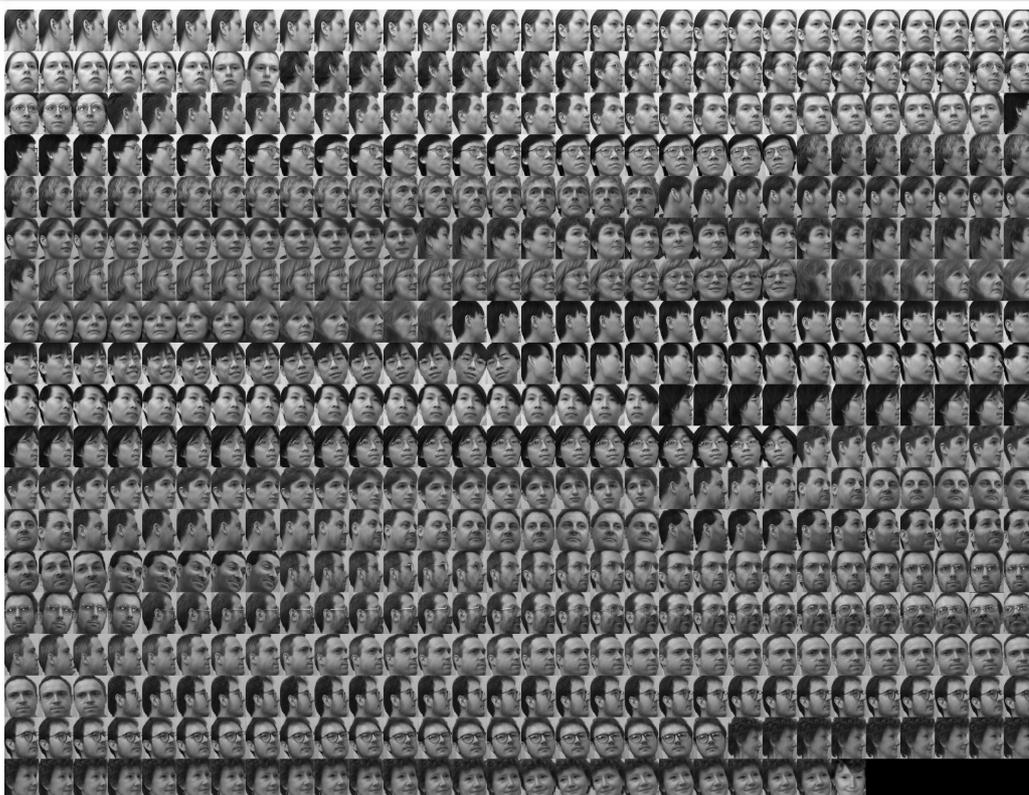
Rechentrick:  $\mathbf{L}_{zz} := \frac{1}{N^2} \mathbf{Z}^H \mathbf{Z}$

$M \times M$ -Matrix  $\rightarrow$  Eigenvektoren praktisch berechenbar

**Es gilt:** Die zu den größten  $M$  ( $M < N^2$ ) Eigenwerten gehörenden Eigenvektoren von  $\mathbf{C}_{zz}$  lassen sich aus den fallend sortierten Eigenvektoren  $\boldsymbol{\eta}_i$  von  $\mathbf{L}_{zz}$  berechnen mittels:

$$\boldsymbol{\varphi}_i = \frac{\mathbf{Z} \boldsymbol{\eta}_i}{\|\mathbf{Z} \boldsymbol{\eta}_i\|}, \quad i = 1, \dots, M \quad \text{Eigengesichter}$$

## 6.2 Anwendungsbeispiele



Mittleres Gesicht  $\bar{\mathbf{x}}$



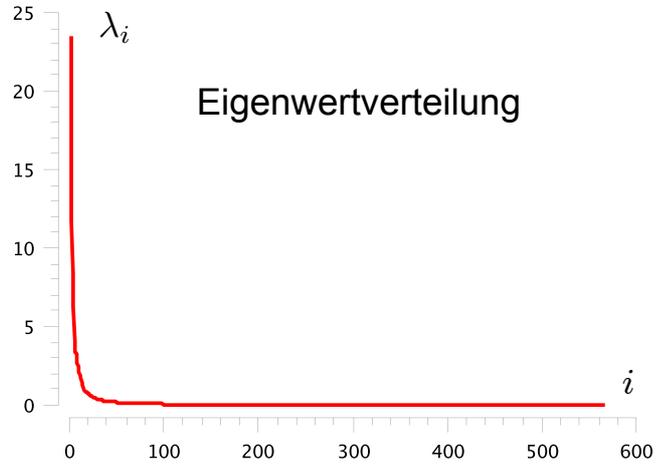
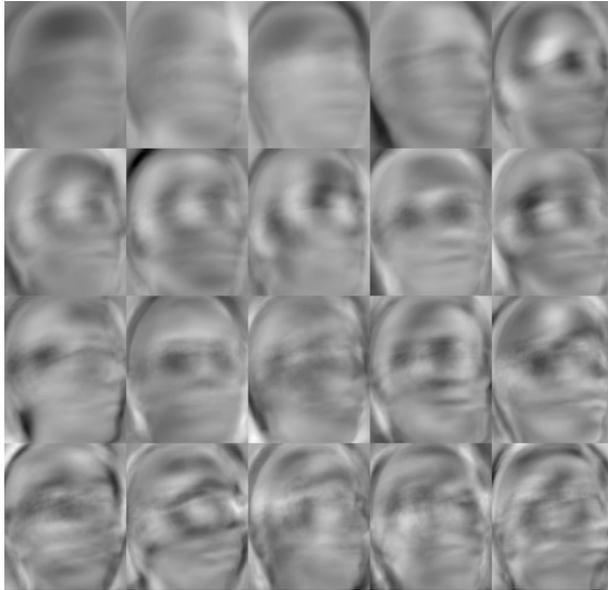


Mittleres Gesicht  $\bar{x}$

Merkmale für die Gesichtserkennung:

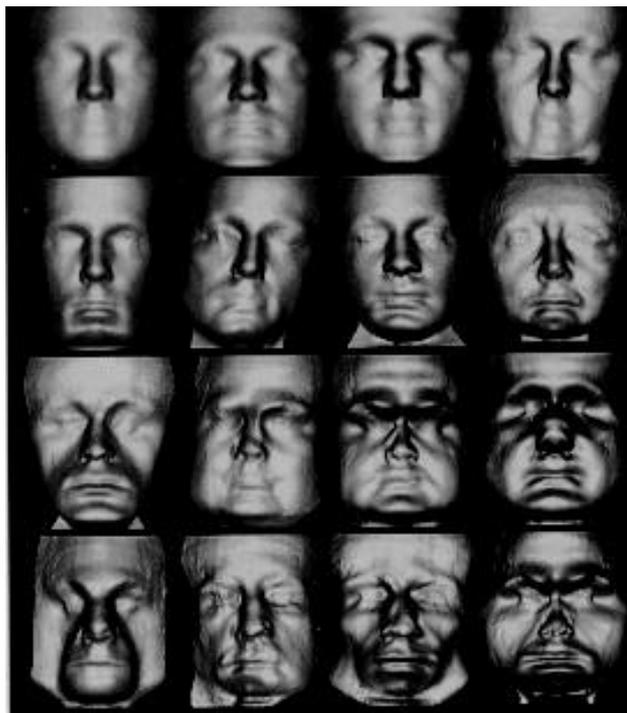
Die ersten 20 Eigengesichter  $\varphi_i$

$$\mathbf{b}_m = [\varphi_1, \dots, \varphi_M]^H (\mathbf{x}_m - \bar{\mathbf{x}})$$



### Beispiel: Gesichtserkennung mit Eigengesichtern (3D-Gesichtsdaten)

Erste 16 Eigengesichter von 3D-Gesichtsdaten



<http://www.ausbcomp.com/~bbott/wik/eigenface.htm>

### Beispiel: Interpolation zwischen Gesichtern

- Grundlage: 3D- und Farbdaten
- Voraussetzung: Gesichter wurden zuvor merkmalsgestützt zur perfekten Deckung gebracht



Anti-John



Durchschnitts-  
mann



50% John



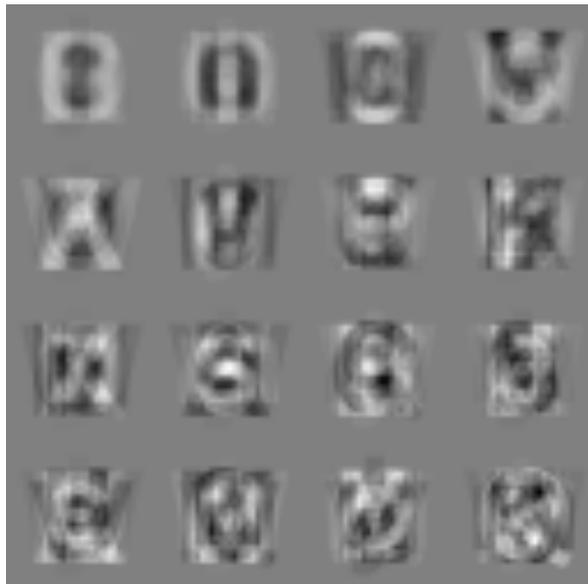
John

Quelle: Leopold, D. A., O'Toole, A. J., Vetter, T. and Blanz, V. (2001)

### Beispiel: Merkmale für die Zeichenerkennung



Zeichenvorrat  
(Bildgröße jeweils  $32 \times 32$ )



Erste 16 von 64 Eigenvektoren  
für die Zeichenerkennung



Zeichenrekonstruktion durch Berücksichtigung von 16 Merkmalen



Zeichenrekonstruktion durch Berücksichtigung von 32 Merkmalen

## 6.3 Approximation durch Cosinus-Transformation

### Markov-Prozess:

- Farbiger Rauschprozess, der als Ausgangssignal eines LTI-Systems 1. Ordnung interpretiert wird, das mit weißem Rauschen  $w_n$  angeregt wird:

$$x_n = a \cdot x_{n-1} + w_n, \quad 0 < a < 1$$

- Rekursives Einsetzen ergibt:  $x_n = \sum_{i=0}^{\infty} a^i w_{n-i}$  (Faltungssumme)
- Durch das Tiefpassverhalten ist der Ausgangsprozess  $x_n$  korreliert
- Nach Abs. 7.2.4 lautet die Autokovarianzmatrix eines Markov-Prozesses:

$$C_{xx} = \sigma_x^2 \begin{bmatrix} 1 & a & a^2 & \dots & a^{N-1} \\ a & 1 & a & \dots & a^{N-2} \\ \vdots & & \ddots & \ddots & \vdots \\ a^{N-1} & a^{N-2} & & & 1 \end{bmatrix}$$

- Die Eigenvektoren der **Autokovarianzmatrix** des Markov-Prozesses sind die Basisvektoren der Karhunen-Loève-Transformation

- **Idee:** Approximation der signalabhängigen KLT durch die signalunabhängige **diskrete Cosinus-Transformation** (DCT), um sich die aufwendige Berechnung der Basisvektoren zu ersparen
- **Voraussetzung:** stark korrelierter Prozess (s. u.)
- Man kann zeigen, dass die Basisvektoren der DCT:

$$\varphi_i(m) = \sqrt{\frac{2}{N}} \gamma_i \cdot \cos((m + 1/2)\pi i/N), \quad \gamma_i = \begin{cases} 1/\sqrt{2} & i = 0, N \\ 1 & \text{sonst} \end{cases}$$

die Eigenvektoren von Matrizen der folgenden Form sind:

$$\mathbf{Q} = \begin{bmatrix} (1-b) & -b & & & \mathbf{0} \\ & -b & 1 & \ddots & \\ & & \ddots & \ddots & -b \\ \mathbf{0} & & & & (1-b) \end{bmatrix}$$

- Weiterhin gilt: Eine Matrix und ihre Inverse haben dieselben Eigenvektoren

- Die inverse der Kovarianzmatrix lässt sich geschlossen angeben (Kap. 7):

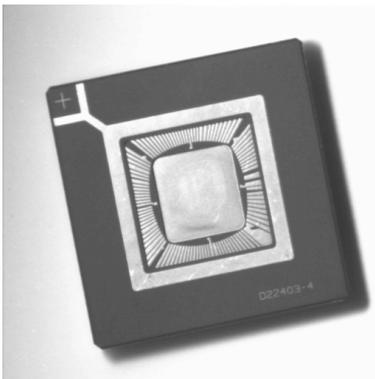
$$\mathbf{C}_{xx}^{-1} = \frac{1+a^2}{\sigma_x^2(1-a^2)} \cdot \mathbf{Q} \cdot \begin{bmatrix} \left(1 - \frac{a^2}{1+a^2}\right) \frac{-a}{1+a^2} & & & \mathbf{0} \\ \frac{-a}{1+a^2} & 1 & \ddots & \\ (1-b) & -b & \ddots & \mathbf{0} \\ -b & \mathbf{0} & \ddots & \frac{-a}{1+a^2} \left(1 - \frac{a^2}{1+a^2}\right) \end{bmatrix}$$

- Für  $0,9 < a < 1$  gilt näherungsweise:  $\frac{-b}{(1-a^2)} \approx \frac{a^2}{1+a^2}$
- Multiplikation einer Matrix mit e. Konstante ändert die Eigenvektoren nicht

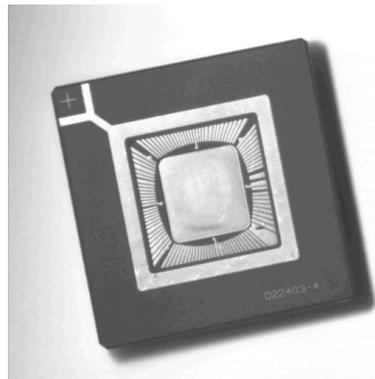
■  $\Rightarrow$  Für  $0,9 < a < 1$  (starke Korrelation) approximiert die DCT die KLT

### Beispiel: Bildcodierung

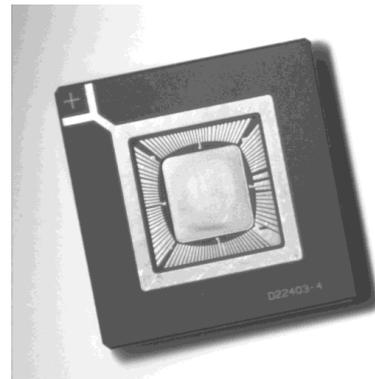
- Grauwertbildzerlegung in quadratische Blöcke:  $x(n,m)$ ,  $n,m = 0 \dots N-1$
- Zweidimensionale DCT: 
$$X(i,j) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) \varphi_i^*(n) \varphi_j^*(m)$$
- Quantisierung des Spektrums zur Speicherreduktion: 
$$X_q(i,j) = \left\lfloor \frac{X(i,j)}{Q(i,j)} \right\rfloor$$
- Die sog. Quantisierungstafeln  $Q(i,j)$  unterdrücken dabei hohe Frequenzanteile selektiv unter Berücksichtigung der menschlichen Wahrnehmung:
  - Intensitäten grober Strukturen (niedrige Ortsfrequenzen) werden empfindlicher wahrgenommen → feiner quantisiert: kleineres  $Q(i,j)$
  - Intensitäten feiner Strukturen (hohe Ortsfrequenzen) werden weniger empfindlich wahrgenommen → gröber quantisiert: größeres  $Q(i,j)$
- Rekonstruktion:
  - Requantisierung: 
$$X_r(i,j) = X_q(i,j) \cdot Q(i,j)$$
  - Inverse DCT



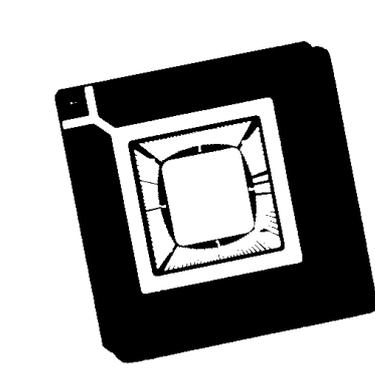
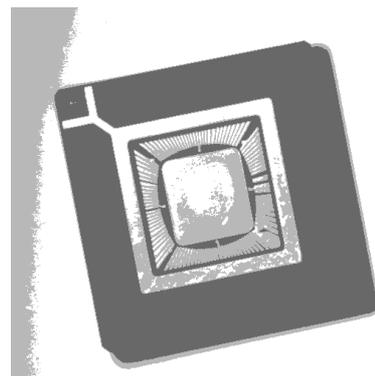
↑ 64 Werte ↓ 8 Werte



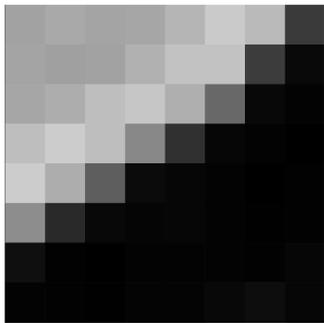
↑ 32 Werte ↓ 4 Werte



↑ 16 Werte ↓ 2 Werte



# 6.3 Approximation durch Cosinus-Transformation



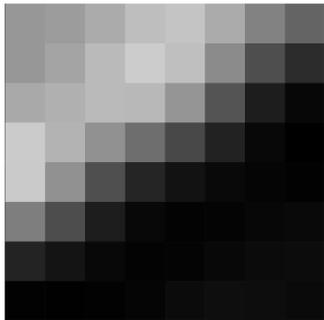
≙

187	191	188	189	197	208	200	115
188	186	187	195	204	204	117	55
189	193	202	206	194	151	53	42
202	209	202	172	106	50	41	35
209	193	144	58	50	41	34	39
175	98	53	47	48	41	38	40
66	40	35	43	42	41	39	50
41	39	37	45	45	53	63	50

DCT →

887	255	-33	51	-6	3	6	-1
411	72	-118	27	-32	-2	1	-4
5	-200	-49	41	-19	20	-10	-4
-16	-21	97	35	-8	12	-7	12
21	18	44	-43	-32	8	-19	12
-1	-14	-15	-35	17	13	-10	0
17	-4	1	14	15	2	-1	0
0	-8	2	0	-12	-20	10	3

$$X_r(i, j) = \left[ \frac{X(i, j)}{Q(i, j)} \right] \cdot Q(i, j) \downarrow$$



≙

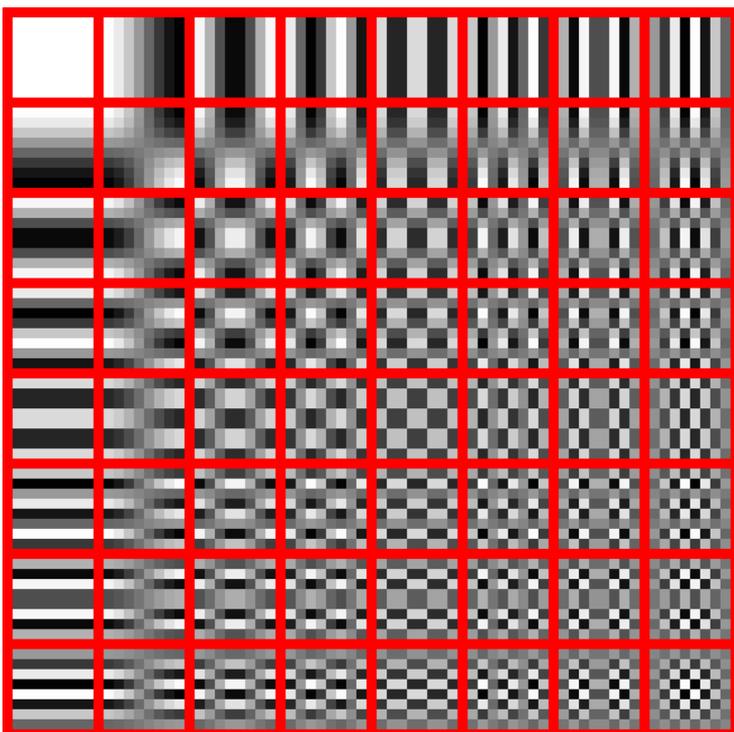
177	180	188	198	201	189	164	144
177	184	196	206	199	170	128	97
187	191	197	195	176	133	79	41
205	192	173	151	123	86	45	17
205	174	129	89	63	47	35	27
162	127	79	43	30	33	41	46
86	67	43	31	33	43	50	52
25	23	25	36	50	57	54	47

← IDCT

880	240	-25	37	0	0	0	0
405	57	-112	0	0	0	0	0
0	-196	-35	0	0	0	0	0
0	0	90	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

# 6.3 Approximation durch Cosinus-Transformation

## Basisfunktionen der 2D-DCT



Quelle: Wikipedia

## Vergrößertes JPEG-Bild



Quelle: Wikipedia