

# Adaptive lineare Transformationen AS1-3

---

---

---

---

---

---

---

---

## Lineare Schichten

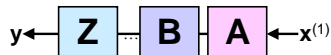
### Sequenz linearer Schichten

$$y^{(1)} = A x^{(1)}$$

$$y^{(2)} = B x^{(2)}$$

...

$$y^{(n)} = Z x^{(n)}$$



$$y^{(n)} = \underline{Z \dots B} A x^{(1)}$$

$$y^{(n)} = M x^{(1)}$$

*Sequenz linearer Schichten = wie nur 1 Schicht !*

---

---

---

---

---

---

---

---

## Hebb'sches Lernen

$$\Delta w = w_i(t) - w_i(t-1) = \gamma_i(t) y_i x \quad \text{Iterative Hebb'sche Lernregel}$$

$$\Delta W = W(t) - W(t-1) = \gamma(t) y x^T$$

$$W = W(1) + W(2) + W(3) + \dots$$

**Problem:** nur anwachsend, ex. kein „Vergessen“,  $w \rightarrow \infty$

**Unendliches Wachstum ??**

---

---

---

---

---

---

---

---

Lösung : Normierung der Gewichte

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \gamma(t) \mathbf{y}\mathbf{x} \quad \text{mit} \quad |\mathbf{w}(t)| = 1$$

Wie?

$$\hat{\mathbf{w}}(t) = \mathbf{w}(t-1) + \gamma(t) \mathbf{y}\mathbf{x}$$
$$\mathbf{w}(t) = \frac{\hat{\mathbf{w}}}{|\hat{\mathbf{w}}|} = \frac{\mathbf{w}(t-1) + \gamma(t) \mathbf{y}\mathbf{x}}{|\hat{\mathbf{w}}|}$$

Wohin konvergiert  $\mathbf{w}(t)$  ?

Rechnung

⇒ Eigenvektoren der Autokorrelationsmatrix  $\mathbf{C}_{xx} := \langle \mathbf{x}\mathbf{x}^T \rangle$  !

---

---

---

---

---

---

---

---

---

---

---

---

Autokorrelation

Autokorrelation zwischen Komponenten von  $\mathbf{x}$

$$\mathbf{A} = \langle \mathbf{x}\mathbf{x}^T \rangle = (a_{ij}), \quad a_{ij} = \langle x_i x_j \rangle_{\mathbf{x}} = \frac{1}{M} \sum_{k=1}^M x_i(k) x_j(k)$$

Beispiel  $\mathbf{x}_1 = (1,2)$   $\mathbf{x}_2 = (3,4)$

$$\mathbf{A} = \frac{1}{2} \begin{bmatrix} 1 \cdot 1 & 2 \cdot 1 \\ 1 \cdot 2 & 2 \cdot 2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 3 \cdot 3 & 3 \cdot 4 \\ 4 \cdot 3 & 4 \cdot 4 \end{bmatrix} = \begin{bmatrix} 5 & 7 \\ 7 & 10 \end{bmatrix}$$

Eigenvektor  $\mathbf{w}$  von  $\mathbf{A}$ :

$$\lambda \mathbf{w} = \mathbf{A}\mathbf{w}$$

Bestimme  $\lambda, \mathbf{w}$  mit

$$\lambda \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{bmatrix} 5 & 7 \\ 7 & 10 \end{bmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

---

---

---

---

---

---

---

---

---

---

---

---

Kovarianz

• Kovarianz zwischen zwei Variablen  $\mathbf{x}, \mathbf{y}$

$$\mathbf{C}_{xy} = \begin{bmatrix} \langle x_1 - \langle x_1 \rangle \rangle \langle y_1 - \langle y_1 \rangle \rangle & \dots & \langle x_1 - \langle x_1 \rangle \rangle \langle y_n - \langle y_n \rangle \rangle \\ \vdots & \ddots & \vdots \\ \langle x_n - \langle x_n \rangle \rangle \langle y_1 - \langle y_1 \rangle \rangle & \dots & \langle x_n - \langle x_n \rangle \rangle \langle y_n - \langle y_n \rangle \rangle \end{bmatrix}$$
$$= \langle \mathbf{x} - \langle \mathbf{x} \rangle \rangle \langle \mathbf{y}^T - \langle \mathbf{y}^T \rangle \rangle$$

---

---

---

---

---

---

---

---

---

---

---

---

## FRAGE

- Was ist der Unterschied zwischen Korrelation  $(x,y)$  und Kovarianz  $(x,y)$  ? Und wann sind sie gleich?

- **Antwort**

$$\text{Kovarianz } (x,y) = \langle \mathbf{x} - \langle \mathbf{x} \rangle \rangle \langle \mathbf{y}^T - \langle \mathbf{y}^T \rangle \rangle$$

$$\text{Korrelation } (x,y) = \langle \mathbf{x} \rangle \langle \mathbf{y}^T \rangle$$

---

---

---

---

---

---

---

---

---

---

## PCA-Transformation

Transform Coding

PCA-Netze

Weissen

ICA-Transformation

---

---

---

---

---

---

---

---

---

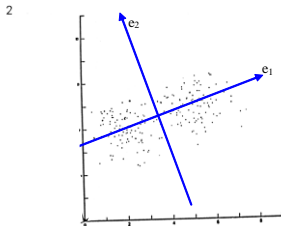
---

## Principal Component Analysis PCA

Zerlegung in orthogonale Eigenvektoren = Basisvektoren

„Hauptkomponentenanalyse“, „principal component analysis“, „Karhunen-Loève-Entwicklung“, „Hotelling-Transformation“, ...

Eigenvektoren – Wozu?



Merkmals-  
transformation  
auf  
Hauptrichtungen

---

---

---

---

---

---

---

---

---

---

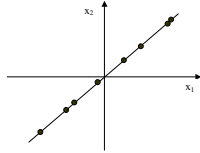
## Principal Component Analysis PCA

### Transformation auf Unkorreliertheit

$$\langle (x_1 - \langle x_1 \rangle)(x_2 - \langle x_2 \rangle) \rangle = 0$$

Unkorreliertheit von  $x_1, x_2$

#### Beispiel



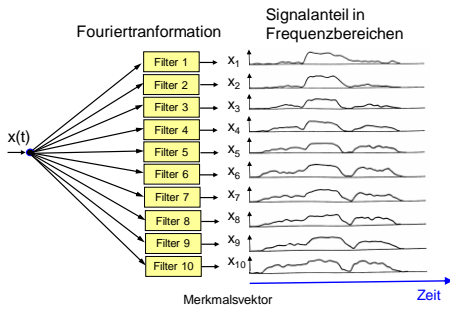
Rauschfrei korrelierte Daten

$$\mathbf{x} = (x_1, x_2) \text{ mit } x_2 = ax_1$$

Rechnung: EV, EW = ?

## Transformation mit minimalem MSE

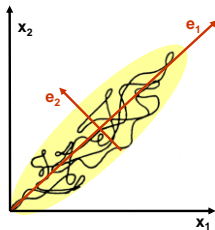
### Beispiel: Sprachkodierung



## Transformation mit minimalem MSE

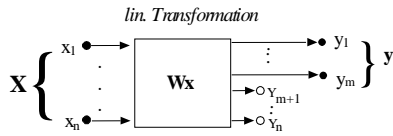
### Beispiel: Sprachkodierung

- Transformation (Rotation) des Koordinatensystems auf Hauptachsen
- Vernachlässigung der Anteile des 2. Kanals: Ersatz des zweiten Kanals durch ersten



## Transformation mit minimalem MSE

### Allgemeine Situation



$$\min_{\mathbf{w}} R(\mathbf{W}) = \min \langle (\mathbf{x} - \hat{\mathbf{x}})^2 \rangle \quad \text{least mean squared error (LMSE)}$$

### Wann minimal ?

## Transformation mit minimalem MSE

### Minimaler Rekonstruktionsfehler

$$\min_{\mathbf{w}} R(\mathbf{W}) = \min \langle (\mathbf{x} - \hat{\mathbf{x}})^2 \rangle \quad \text{least mean squared error (LMSE)}$$

$$\mathbf{x} = \sum_{i=1}^m y_i \mathbf{w}_i + \sum_{i=m+1}^n y_i \mathbf{w}_i \quad \hat{\mathbf{x}} = \sum_{i=1}^m y_i \mathbf{w}_i + \sum_{i=m+1}^n c_i \mathbf{w}_i \quad y_i = \mathbf{x}^T \mathbf{w}_i$$

- Was ist die beste Schätzung für die Konstanten  $c_i$ ?  
 $\min R(c_i) = ?$       **Rechnung!**
- Bei welchen Basisvektoren  $\mathbf{w}_i$  ist der Fehler minimal?  
 $\min R(\mathbf{w}_i) = ?$       **Rechnung!**

## FRAGE

- Angenommen,  $\{\mathbf{x}\}$  bildet einen **Hyperelipsoid**.
  - In welche Richtung gehen die beiden Eigenvektoren?
  - Wie groß sind die Eigenwerte dazu?
- Angenommen,  $\{\mathbf{x}\}$  bildet eine **Hyperkugel** mit  $\sigma^2 = 1$ .
  - In welche Richtung gehen die Eigenvektoren?
  - Wie groß sind die Eigenwerte dazu?

## PCA-Transformation

# Transform Coding

## PCA-Netze

## Weissen

## ICA-Transformation

## Transform coding – Wozu?

Verlustfreie Kodierung:  $\text{Max} \approx 1:2$  (Zip).

Unnötig bei

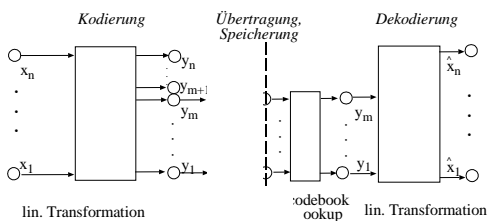
- Telekommunikation (z.B. Bildtelefon)
- Satellitenübertragung (z.B. Wettersatelliten etc.)
- Bilddatenbanken (z.B. Umweltdaten, Medizindaten, Industrieteile, Teleshopping etc.)
- Digitale Musik (MP3)
- Hochauflösendes Fernsehen (HDTV)

Allgemein: Multi-Media Daten (MPEG 7)

## Konzept Transform Coding

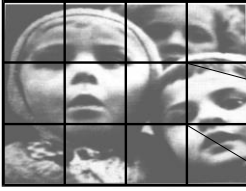
### Kodierung und Dekodierung

Anhang C



## Aufteilung in Unterblöcke

### Kodierung und Dekodierung



JPEG, MPEG Blockgröße:  
8x8 Pixel SW  
16x16 Pixel Farbe

---

---

---

---

---

---

---

---

---

---

## Transform Coding

**Kodierung:** 256x256 Pixel, 8 Bit Grauwert, 32x32 Unterbilder,  
je 8x8=64 Pixel, 8 Neuronen. *Kompression = ?*



und Dekodierung:



Kodierung in 0,36 Bits/Pixel statt 8  
= Kompression mit Faktor 22

---

---

---

---

---

---

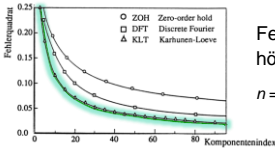
---

---

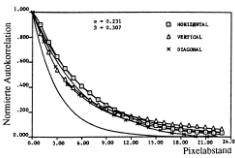
---

---

## Eigenschaften natürlicher Bilder



Fehler beim Vernachlässigen  
höherer Komponenten  
 $n = 256 \times 256 = 65536$  Komponenten



Bildmodellierung durch  
Pixelkorrelationen

$$C_{xx}(x-x') = e^{-\alpha |x_1 - x'_1| - \beta |x_2 - x'_2|}$$

---

---

---

---

---

---

---

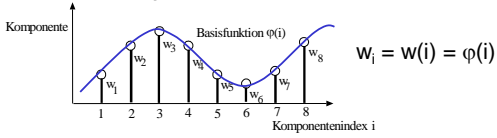
---

---

---

## Eigenvektoren und Eigenfunktionen

Min. Fehler  $\Leftrightarrow$  Eigenvektoren. Und im kontin. Fall?



Eigenvektor  $\mathbf{w}$  = diskretisierte Eigenfunktion  $\phi(\mathbf{w})$

## kont. Eigenwertgleichung

$\mathbf{A}\mathbf{w} = \lambda_{12}\mathbf{w}$   $\mathbf{A} = \langle (x_1, x_2) \rangle$  der Bilder  $w(x_1, x_2)$

$$\int_0^N \int_0^N C(x_1, x_2, x_1', x_2') \phi(x_1', x_2') dx_1' dx_2' = \lambda_{12} \phi(x_1, x_2)$$

Separierbare Korrelationen

$$C(x_1, x_2, x_1', x_2') = C(x_1, x_1') C(x_2, x_2')$$

→ Separierbare Basisfunktionen

$$\phi(x_1, x_2) = \phi(x_1)\phi(x_2), \quad \lambda_{12} = \lambda_1 \lambda_2$$

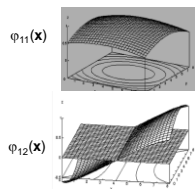
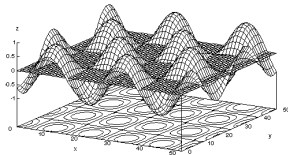
→ Zwei 1-dimensionale Differentialgleichungen

$$\lambda \phi''(x) + (2\gamma - \lambda\gamma^2)\phi(x) = 0$$

## Transform Coding

Lösungen: Eigenfunktionen mit Eigenfrequenzen  $b_1\gamma N/2\pi$

$$\phi_i(x) = a \cos(b_1\gamma(x-N/2)) \text{ mit } \lambda_i = \frac{2}{\gamma(b_1^2 + 1)} \text{ wobei } b_1 \tan(b_1\gamma N/2) = 1$$



$$N=8, \alpha=0, 125, \beta=0, 249$$



## Beispiel: Bildkodierung

### Eigenfunktion coding vs. Kosinustransformation (JPEG)

Eigenfunktion 1-dim

$$x_i = \sum_{j=1}^m y_j \phi_{ij}$$

Eigenfunktionen bei abfall.  
Korrelation

$$\phi_i(x) = a \cos(b_i \gamma (x-N/2))$$



Kosinus

$$x_i = \sum_{j=1}^m y_j \cos(\pi i j / m)$$

EF für Parameter  $\alpha, \beta \sim$  mittl. Bild  
identisch mit **cos**

→ bildunabh. Kodierung

## PCA-Transformation

## Transform Coding

## PCA-Netze

## Weissen

## ICA-Transformation

## Oja-Lernregel

### EINE Lernregel für Hebb-Lernen und Gewichtsnormierung

1. Hebb-Regel  $\mathbf{W}(t) = \mathbf{W}(t-1) + \gamma(t) \mathbf{y}\mathbf{x}$

2. Normierung  $w_i(t) \rightarrow \frac{w_i(t)}{|\mathbf{w}(t)|}$

Einsetzen 1. in 2.  $\mathbf{W}(t-1) + \gamma(t) \mathbf{y}\mathbf{x}$

$$\left[ \sum_i (w_i(t-1) + \gamma \mathbf{x}_i \mathbf{y})^2 \right]^{1/2}$$

und  $f(\gamma)$  in einer Taylorreihe nach  $\gamma$  entwickeln.

Terme mit  $\gamma^2$  vernachlässigen ergibt

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \gamma(t) \mathbf{y} \left[ \mathbf{x}(t) - \mathbf{W}(t-1)\mathbf{y} \right] \quad \text{Oja Lernregel}$$

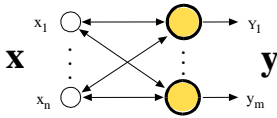
$\mathbf{x}_{\text{neu}}$

PCA Netze für den Unterraum



Oja-Netz

$w(t) = w(t-1) + \gamma(t) y [x(t) - w(t-1)y]$  *Oja Lernregel*  
 $w_i(t) = w_i(t-1) + \gamma(t) y_i [x(t) - x^-]$



$x^- = w_i(t-1)y_i$   
 $x^- = \sum_{i=1}^m w_i y_i$

Ansatz: Zielfunktion  $R(w) = \langle (x-x^-)^2 \rangle$  minimieren

Konvergenzziel: Unterraum der EV mit größtem EW

---

---

---

---

---

---

---

---

---

---

PCA Netze für geordnete Zerlegung



Sanger-Methode

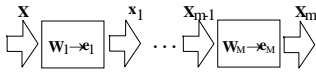
Sanger 1988

Vollständige Zerlegung von  $\{x\}$  in  $n$  Eigenvektoren (Gram-Schmidt)

$\{x\}_0 = \{x\}, i=1$

1. Suche die Richtung größter Varianz in  $\{x\}_{i-1}$ . Dies ist  $e_i$ .
2. Ziehe diese Richtung (Dimension) von  $\{x\}_{i-1}$  ab. Wir erhalten  $\{x\}_i$ .
3. Wenn  $i < n$ , setze  $i := i+1$ , gehe zu 1.

Diskret für stochastisches  $x, w$ : Sanger-Netz




---

---

---

---

---

---

---

---

---

---

PCA Netze für geordnete Zerlegung



Sanger-Methode: stochastischer Algorithmus

Lernen:  $w_i(t) = w_i(t-1) + \gamma(t) y_i x_i$  "Generalisierte" Hebb-Regel

Musterreduktion Stufe 1

$x_2 := x_1 - w_1(t-1)y_1$       $x_2 \perp w_1$  !!

...

Stufe k

$x_{k+1} := x_k + a w_k(t-1)y_k$      und  $x_1 := x, a := -1, k = 1..i-1$

Lernen Stufe k

$\rightarrow w_{k+1}(t) = w_k(t-1) + \gamma(t) y_k (x_k + a \sum_{j=1}^{k-1} w_j y_j)$   
 $a = -1 \text{ max EW, } a = +1 \text{ min EW}$

---

---

---

---

---

---

---

---

---

---

## PCA-Transformation

## Transform Coding

## PCA-Netze

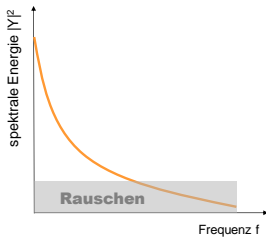
## Weissen

## ICA-Transformation

### Whitening Filter

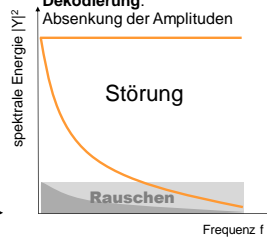
#### Problem

Störung von Signalen durch Rauschen



#### Lösung

**Kodierung:**  
Verstärkung zu geringer Amplituden  
**Dekodierung:**  
Absenkung der Amplituden



### Whitening Filter

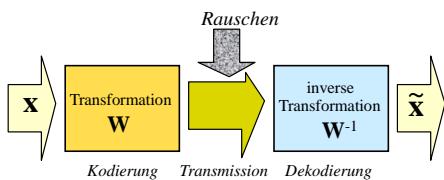
**Shannon:** Whitening für alle Frequenzen, d.h. alle diskreten Signalbänder

**Übertragung** auf parallele Signale  $y_i$ : gleiche Varianz aller durch Transformation  $\mathbf{W}$ .

**Anhebung** zu geringer Amplituden: Wähle  $\mathbf{W}$  so,

$$\text{daß } \langle y_i, y_j \rangle = 1 \text{ bei } i = j, \text{ und } = 0 \text{ sonst; also } \langle y y^T \rangle = \mathbf{I}$$

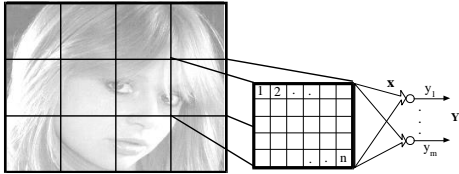
**Absenkung** der Amplituden: durch inverse Matrix  $\mathbf{W}^{-1}$



## Beispiel: Bildentstörung

### Bildkodierung

Zerteilen in Blöcke, jeder Block = Mustervektor




---

---

---

---

---

---

---

---

---

---

## Beispiel: Bildentstörung




---

---

---

---

---

---

---

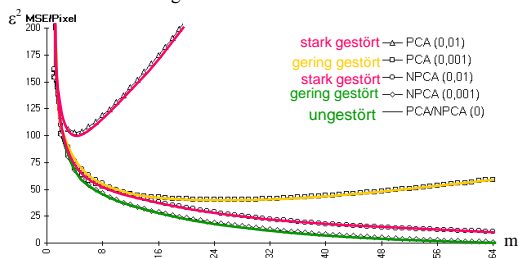
---

---

---

## Rauschunterdrückung

Problem: Vollständige Rekonstruktion




---

---

---

---

---

---

---

---

---

---

## PCA-Transformation

## Transform Coding

## PCA-Netze

## Weissen

## ICA-Transformation

---

---

---

---

---

---

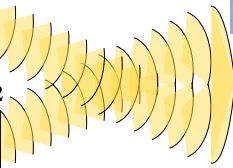
---

---

### Einleitung

#### Lineare Mischung unabhängiger Quellen

Sprecher 1



Mikro 1



Sprecher 2



Mikro 2



---

---

---

---

---

---

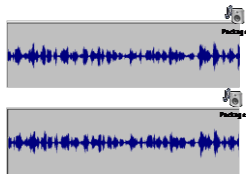
---

---

### ICA Anwendung: Audioanalyse

Coctail-Party-Effekt

mixed sources



2 Sprecher

demixed sources



---

---

---

---

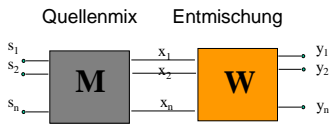
---

---

---

---

## Lineares ICA-Modell



Ziel:  $W \rightarrow M^{-1}$   
 $y \rightarrow s$

mit  $p(y) = p(y_1, \dots, y_n) = p(y_1) \dots p(y_n)$  unabhängige Kanäle

Unabhängigkeit **notwendig** zur Quellentrennung.

Yelling, Weinstein (1994): auch **hinreichend**!

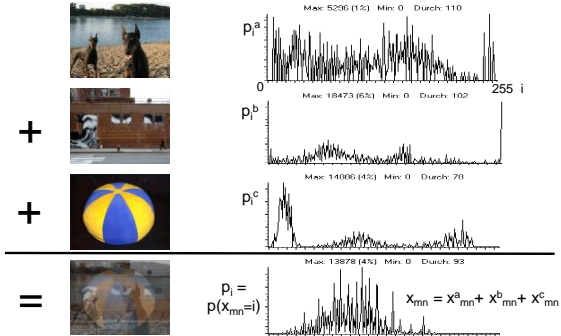
## ICA-Einschränkungen

- **Quellenzahl = Mischzahl**  
M muß **regulär** sein  $\Leftrightarrow$  nur dann **ex.  $M^{-1}$**
- **Ausgabereihenfolge unbestimmt**  
**Reihenfolge** in Produkt  $p(y_1) \dots p(y_n)$  ist unwichtig  
 $\Rightarrow M^{-1}$  bis auf Permutation P bestimbar:  $M^{-1} \rightarrow M^{-1} P$
- **Unbekannte Skalierung**  
 $\Rightarrow \sigma_i := 1$
- **2 Gaußsche Quellen lassen sich nicht trennen**  
 $\Rightarrow$  **max 1** Gaußsche Quelle

## Erzeugung einer Verteilung

- **Erzeugen der Normalverteilung**  
**Problem:**  $\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt$   
Verteilungsfkt.  $\Phi$  hat keine analyt. Lösung, nur tabelliert  
 $\Rightarrow$  **Inverse Funktion ex. nicht**  
**Lösung:** Überlagern von Verteilungen  
Nutzung des Zentralen Grenzwertsatzes:  
 $\lim_{n \rightarrow \infty} x(n) = N(0,1)$  mit  $x(n) = \sum_{i=1}^n \frac{(X_i - \mu)}{\sigma}$   
mit beliebig verteilten  $X_i$  und gemeinsamen  $\mu = \langle X_i \rangle$  und  $\sigma$   
 $X_i$  uniform aus  $U(-1/2, 1/2) \Rightarrow N(0,1)$  durch  $n = 12$

## Erzeugung einer Normalverteilung




---

---

---

---

---

---

---

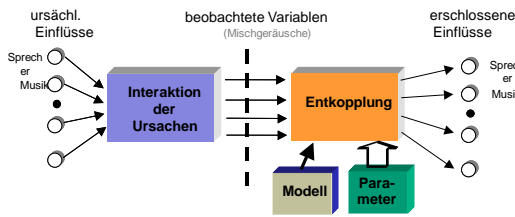
---

---

---

## ICA Systemüberblick

### Trennung unbekannter Quellen



### Independent Component Analysis (ICA)

---

---

---

---

---

---

---

---

---

---

## DEF Information

$$I \sim n = \text{ld}(2^n) = \text{ld} \text{ (Zahl der möglichen Daten)}$$

$$I \sim \text{ld}(1/P) \quad [\text{Bit}]$$

DEF  $I(X) := \ln(1/P(x^k)) = -\ln(P(x^k))$  *Information*

DEF  $H(X) := \sum_k P(x^k) I(x^k) = \langle I(x^k) \rangle_k$  *Entropie*

$$H(X) := \int_{-\infty}^{+\infty} p(x) \ln p(x)^{-1} dx \quad \textit{differenzielle Entropie}$$

Frage: Wieviel Information hat eine 32-bit floating-point Zahl?

---

---

---

---

---

---

---

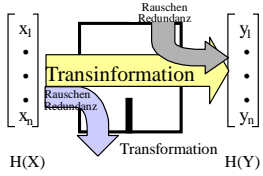
---

---

---

## DEF Transinformation

DEF  $H(X,Y) = H(X) + H(Y) - I(X;Y)$  *Verbundentropie*



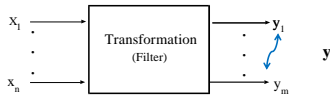
DEF  $I(X;Y) = H(X) + H(Y) - H(X,Y)$  *Transinformation*

## ICA - Algorithmen 1

Ziel: *minimale Transinformation zwischen den Ausgaben  $y_i$*

$x$  = Kanäle,

stoch. Variablen



*Transinformation*  $I(x_1;x_2) = H(x_1) + H(x_2) - H(x_1,x_2)$

*minimal* bei  $I(x_1;x_2) = 0$  bzw. *maximaler* Entropie  $H(x_1,x_2) = H(x_1) + H(x_2)$   
bzw.  $p(x_1,x_2) = p(x_1) \cdot p(x_2)$  *stochastische Unabhängigkeit* der Variablen

$$W(t+1) = W(t) - \frac{\partial}{\partial W} \gamma I(y_1; y_2; \dots; y_n) \quad \text{Gradientenabstieg}$$

(Amari, Young, Cichocki 1996)

Entwicklung von  $p(y_1, y_2, \dots, y_n)$  in  $I(y_1; y_2; \dots; y_n)$  nach höheren Momenten

$$W(t+1) = W(t) - (1 - f(y)) y^T W(t) \quad \text{mit } f(y) = \frac{3}{4} y^{11} + \frac{25}{4} y^9 - \frac{14}{3} y^7 - \frac{47}{4} y^5 + \frac{29}{4} y^3$$

## ICA Anwendung: Audioanalyse

Coctail-Party-Effekt

2 Sprecher

mixed sources

demixed sources





**FRAGE**

- Wodurch werden die Quellen bei der Informationsmethode getrennt?
- a) Durch maximale Transinformation zwischen den Ausgabevariablen
  - b) Durch maximale Entropie der Ausgaben
  - c) Durch minimale Transinformation zwischen den Ausgabevariablen
  - d) Durch minimale Verbundentropie der Ausgabekanäle

---

---

---

---

---

---

---

---

---

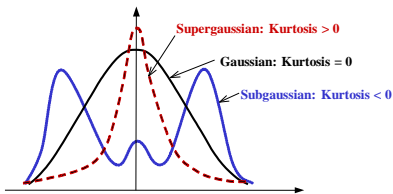
---

**Statist. Momente und Kurtosis**

**Momente** einer Zufallsvariablen  $x$  :  
 $\alpha_k = \langle x^k \rangle$  z.B.  $\alpha_1 = \langle x \rangle$  **Mittelwert**

**Zentrale Momente** einer Zufallsvariablen  $x$ :  
 $m_k = \langle (x - \alpha_1)^k \rangle$ , z.B.  $m_2 = \langle (x - \alpha_1)^2 \rangle$  **Varianz**

**Wölbungsmaß Kurtosis**:  $kurt(x) = [\langle (x - \alpha_1)^4 \rangle - 3m_2^2] / m_2^2$




---

---

---

---

---

---

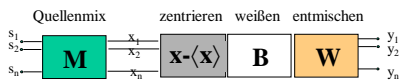
---

---

---

---

**ICA-Algorithmen: Vorverarbeitungsfolge**



**Zentrieren**  $\langle x \rangle = 0$   $\langle (x - \langle x \rangle)^2 \rangle = 1$   
 • Mittelwertbildung, z.B. iterativ durch  $w_0(t+1) = w_0(t) - \gamma (w_0 \cdot x)$ ,  $\gamma = 1/t$

**Weißen**  
 • PCA durchführen:  $w_i$  Eigenvektoren von  $C = \langle xx^T \rangle$  mit  $|w_{ij}| = 1$  und Eigenwerten  $\lambda_i$   
 • Gewichtsvektoren  $w_i$  normieren zu  $w_i / \lambda_i^{1/2}$ . Dies führt zu  $\langle y^2 \rangle = w_i^T \langle xx^T \rangle w_i = w_i^T \lambda_i w_i = 1$

**Entmischen**  
 • ICA Algorithmen, z.B. minimale Transinformation, maximale Kurtosis etc.  
 • *Speziell:* dekorrelierte  $x$  benötigen nur eine orthogonale Matrix  $W$  (Vereinfachung)

---

---

---

---

---

---

---

---

---

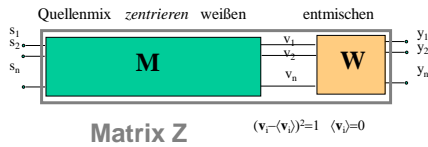
---

## ICA – Algorithmen 2

**Ziel: extreme Kurtosis** (Delfosse, Loubaton 1995)

Extrema bei  $s_j = \text{unabh. Komp.}$  und  $z_j = +/-1$

$$\text{kurt}(y) = \text{kurt}(\mathbf{w}^T \mathbf{v}) = \text{kurt}(\mathbf{w}^T \mathbf{M} \mathbf{s}) = \text{kurt}(\mathbf{z}^T \mathbf{s}) = \sum_{j=1}^n z_j^4 \text{kurt}(s_j)$$



## ICA – Algorithmen 2

### Sequentielle Extraktion aller Komponenten

**Gegeben:** Trainingsmenge  $\{v(0)\}$     **Fixpunktalgorithmus**

$$\mathbf{w}_1(t+1) = \langle (\mathbf{w}_1^T \mathbf{v})^3 \mathbf{v} \rangle - 3 \mathbf{w}_1 \quad \text{mit } |\mathbf{w}_1| = 1$$

Konvergenz zum **1. ICA-Vektor**. (Hyvarinen, Oja 1996)

Dann **neue Trainingsmenge** durch  $\mathbf{v}(1) = \mathbf{v}(0) - \mathbf{w}_1 y_1$

$$\mathbf{w}_2(t+1) = \langle (\mathbf{w}_2^T \mathbf{v})^3 \mathbf{v} \rangle - 3 \mathbf{w}_2 \quad \text{mit } |\mathbf{w}_2| = 1$$

Konvergenz zum **2. ICA-Vektor**, usw.

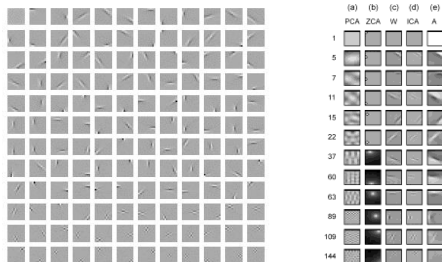
### Schnellere Konvergenz: Orthogonalisierung

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) - \sum_{j=1}^{i-1} (\mathbf{w}_i^T \mathbf{w}_j) \mathbf{w}_j \quad j < i$$

## ICA-Anwendungen: Bildprimitive

Zerteilung von **Naturbildern** in 12x12 Unterbilder = 144 Kanäle = 1 Sample

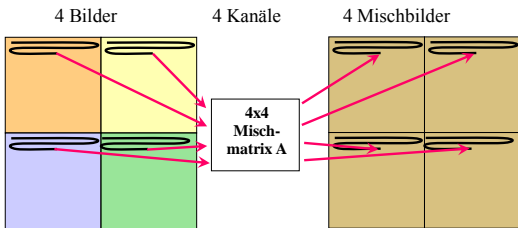
**Unabhängige Komponenten** (Filter-Koeff.)    **Alternative Analyseverfahren**



Bell, Sejnowski (Vision Research 1996)

**ICA Anwendung: Bildentmischung**

4 Bilder, sequentiell gerastert = 4 Quellen (Hyvärinen, Oja 1996)



Automatische Entmischung ?

---

---

---

---

---

---

---

---

---

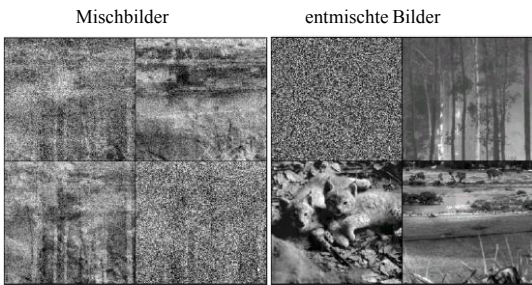
---

---

---

**ICA Anwendung: Bildentmischung**

4 Bilder, sequentiell gerastert = 4 Quellen (Hyvärinen, Oja 1996)




---

---

---

---

---

---

---

---

---

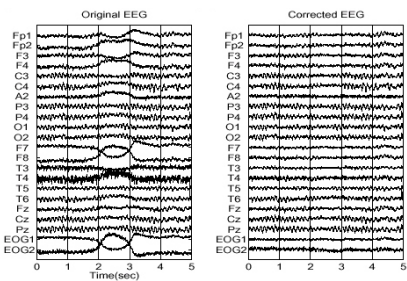
---

---

---

**ICA-EEG-Filterung**

Korrigierte EEG-Aufnahmen ohne 5 ICA-Muskelaktivitäten, Mischung




---

---

---

---

---

---

---

---

---

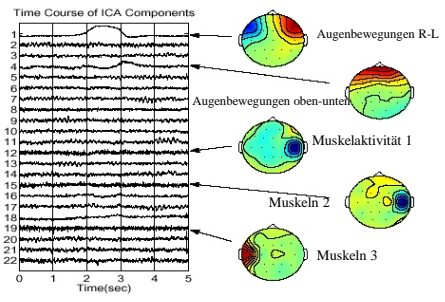
---

---

---

# ICA-EEG-Analyse

Entmischte EEG-Aufnahmen: unabhängige Zentren, z.B. Muskelaktivität



---

---

---

---

---

---

---

---

---

---