

---

## Binärlogarithmus

$$\log_2 x = \frac{\log x}{\log 2}$$

---

## Entscheidungsgehalt

$H_0 = \log_2 K$  mit  $K$  = Anzahl Symbole

---

## Informationsgehalt

$$I(a_k) = -\log_2 P(a_k) \text{ [Bit]}$$

- Je kleiner  $P(a_k)$ , desto größer  $I$ .
- Wenn  $P(a_k) = 1$ , dann  $I(a_k) = 0$ .

---

## Entropie – mittlerer Info.gehalt

$$H = -\sum_{k=1}^K \left[ P(a_k) \cdot \log_2 P(a_k) \right] \text{ [Bit]}$$

- Wenn alle Sym. gleich wahrscheinlich  
 $I(a_k) = H_0 = H$
- Max. bei  $P(a_k) = \frac{1}{K}$
- Einfachere Berechnung bei  $P(a_k) = \frac{i_k}{c}$ :

$$H = \frac{c \cdot \log(c) - \sum_{k=1}^K \left[ i_k \cdot \log(i_k) \right]}{c \cdot \log(2)} \text{ [Bit]}$$

---

## Redundanz

$$R = H_0 - H \text{ [Bit]}$$

- relative Red.  $R = \frac{H_0 - H}{H}$

---

## Ideale Codewortlänge

$$n = -\log_2 P(a_k) \text{ [Bit]}$$

---

## Mittlere Codewortlänge

$$\bar{m} = \sum_{k=1}^K \left[ P(a_k) \cdot m_k \right] \text{ [Bit]}$$

## Verbundentropie

$$H(a_i, a_j) = - \sum_{i=1}^I \sum_{j=1}^J \left[ P(a_i, a_j) \cdot \log_2(P(a_i, a_j)) \right] \text{ [Bit]}$$

- Einfachere Berechnung bei  $P(a_i, a_j) = \frac{m_{ij}}{c}$ :

$$H(a_i, a_j) = \frac{c \cdot \log(c) - \sum_{i=1}^I \sum_{j=1}^J \left[ m_{ij} \cdot \log(m_{ij}) \right]}{c \cdot \log(2)} \text{ [Bit]}$$

---

## Bedingte Entropie

$$H(a_i|a_j) = - \sum_{i=1}^I \sum_{j=1}^J \left[ P(a_i, a_j) \log_2(P(a_i|a_j)) \right] \text{ [Bit]}$$

---

## Beziehungen zw. den Entropien

$$H(a_i, a_j) = H(a_i|a_j) + H(a_j)$$

---

## Datenrate eines Kanals

$$C = 2 \cdot B \cdot \log_2(L) \text{ [Bit/s]}$$

- mit Anzahl der unterschiedlichen Amplituden  $L$

$$C = B \cdot \log_2 \left( 1 + \frac{S}{N} \right)$$

für  $\frac{S}{N} \ll 1$ :

$$C \approx 1.44 \cdot \frac{B}{\text{Hz}} \cdot \frac{S}{N} \text{ [Bit/s]}$$

für  $\frac{S}{N} \gg 1$ :

$$C \approx 0.332 \cdot \frac{B}{\text{Hz}} \cdot \frac{SNR}{\text{dB}}$$

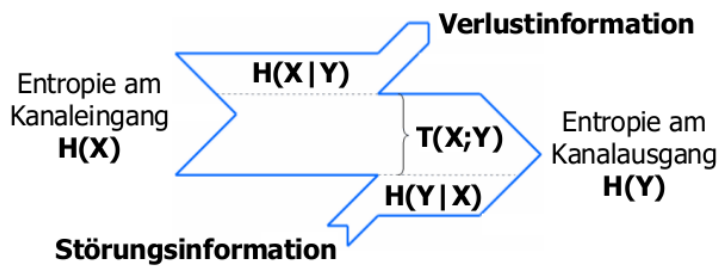
---

## Kanalkapazität

$$C = \max_{P(a_k)} \left[ T(X; Y) \right]$$

---

## Transinformation



$$T(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = T(Y; X)$$

Kanal ist

- *verlustfrei*, wenn Verlustinformation  $H(X|Y) = 0$ .  
→  $C$  ist maximal, wenn alle Eingangssym. gleich wahrscheinlich
- *deterministisch*, wenn Störungsinformation  $H(Y|X) = 0$ .  
→  $C$  ist maximal, wenn alle Ausgangssym. gleich wahrscheinlich
- *ungestört*, wenn er sowohl verlustfrei als auch deterministisch ist  
→  $C$  ist maximal, wenn alle Ausgangssym. gleich wahrscheinlich oder alle Eingangssym. gleich wahrscheinlich

---

## Theorem der Kanalcodierung

Wenn  $H' \leq C'$  gilt, dann existiert immer eine Kanalcodierung, welche eine Übertragung der Quellsymbole mit beliebig kleiner Fehlerwahrscheinlichkeit ermöglicht (u. U. nur mit großem Aufwand).

---

## Kraft'sche Ungleichung

Für einen eindeutigen, binären Code mit  $K$  Codewörtern der Länge  $m_k$  gilt:

$$\sum_{k=1}^K 2^{-m_k} \leq 1$$

---

## Theorem der Quellencodierung

Die mittlere Länge  $\bar{m}$  eines Präfixcodes kann stets so gewählt werden, dass gilt:

$$H \leq \bar{m} < H + 1$$

Beim Huffman-Code:

$$H \leq \bar{m} < H + p_{\max} + 0.086$$

bzw. wenn  $p_{\max} > 0.5$ :

$$H \leq \bar{m} < H + p_{\max}$$

ID03

---

## Transformationen

**WHT – Walsh-Hadamard**

Einfach; Rechteck-Funktion mit -1 und 1.

**DFT – Diskrete-Fourier**

Sinus- und Cosinus-Funktionen als Basis

**DCT – Diskrete-Cosinus**

Cosinus-Funktionen als Basis

Guter Kompromiss aus Aufwand und Nutzen

**KLT – Karhunen-Loeve**

individuelle Basisfunktionen, hoher Rechenaufwand

---

## Präcodierungen

- Lauflängencodierung
- Burrows-Wheeler-Transformation
- Lempel-Ziv-Verfahren
- Move-to-Front-Codierungen
- Transformationen (z. B. DCT)
- Teilband-Zerlegungen

---

## $2 \times 2$ -WHT-Hintransformation

$$F(u, v) = \frac{1}{2} \sum_{x=0}^1 \sum_{y=0}^1 f(x, y) \cdot (-1)^{x \cdot u + y \cdot v} \quad \text{für } u = 0, 1 \text{ und } v = 0, 1$$

---

## $2 \times 2$ -WHT-Rücktransformation

$$f(x, y) = \frac{1}{2} \sum_{u=0}^1 \sum_{v=0}^1 F(u, v) \cdot (-1)^{x \cdot u + y \cdot v} \quad \text{für } x = 0, 1 \text{ und } y = 0, 1$$

---

## 1D-DCT-Hintransformation

$$F(u) = \frac{C(u)}{\sqrt{N}} \cdot \sum_{x=0}^{N-1} \left[ f(x) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{2N} \right] \quad \text{für } u = 0, 1, \dots, N-1$$

$$C(u) = \begin{cases} 1 & \text{für } u = 0 \\ \sqrt{2} & \text{sonst} \end{cases}$$

---

## 1D-DCT-Rücktransformation

$$f(x) = \sum_{u=0}^{N-1} \left[ \frac{C(u)}{\sqrt{N}} \cdot F(u) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{2N} \right] \quad \text{für } x = 0, 1, \dots, N-1$$

$$C(u) = \begin{cases} 1 & \text{für } u = 0 \\ \sqrt{2} & \text{sonst} \end{cases}$$

---

## 2D-DCT-Hintransformation

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left[ f(x, y) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{2N} \cdot \cos \frac{(2y+1) \cdot v \cdot \pi}{2N} \right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{sonst} \end{cases}$$

---

## 2D-DCT-Rücktransformation

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \left[ C(u) C(v) F(u, v) \cdot \cos \frac{(2x+1) \cdot u \cdot \pi}{2N} \cdot \cos \frac{(2y+1) \cdot v \cdot \pi}{2N} \right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{sonst} \end{cases}$$

# Allgemeines

---

## Datenkompression Optimierung

- Bestimmte Mindestkompression (Kanälen mit begrenzter Datenrate)
  - Echtzeit (z.B. bei Videokonferenz-Systemen oder digitalen Videorecordern)
  - Bei verlustbehafteter DK bestimmte Mindestqualität nicht unterschreiten
  - Maximale Decodierungszeit darf nicht überschritten werden (Videokonferenzsystemen, Bildtelefonen)
- 

## Quantisierung Optimierung

- Minimierung der Quantisiererfehlerleistung (objektives Maß)
  - Minimierung der subjektiven Wahrnehmbarkeit von Fehlern Bildtelefonen)
- 

## wahrnehmungsbasierte Codierung

Perception-based coding

---

## Vektor-Quantisierung

Aufwand beim Encoder ist deutlich größer als beim Decoder. Gut für Broadcast.

---

## Huffman

Dekrementieren und mit Einsen auffüllen  
11 101 100 011 010 0011 0010 0001 0000

## Kommazahl zu Binär

In TR mit 2 multiplizieren  
wenn vor Komma ungerade: 0  
wenn gerade: 1

---

## LZ Algorithmen

### LZ78

- Wörterbuch zu Beginn leer

### LZW

- Wörterbuch zu Beginn gefüllt