
Information and Entropy

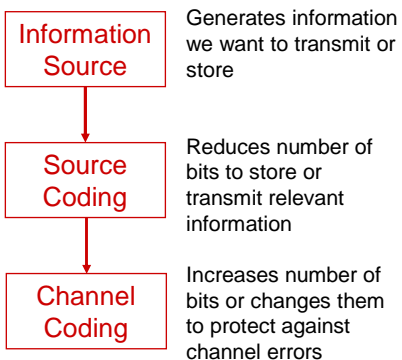
- Shannon's Separation Principle
- Source Coding Principles
- Entropy
- Variable Length Codes
- Huffman Codes
- Joint Sources
- Arithmetic Codes
- Adaptive Codes



Shannon's Separation Principle

Assumptions:

- Single source and user
- Unlimited complexity and delay



Practical Systems

- Many applications are not uni-directional point-to-point transmissions:
 - Feedback
 - Networks
- In any practical system, we cannot effort unlimited complexity *neither unlimited delay*:
 - There will always be a small error rate unless we tolerate sub-optimality
 - It might work better to consider source and channel coding jointly
 - Consider effect of transmission errors on source decoding result



Source Coding Principles

- The source coder shall represent the video signal by the minimum number of (binary) symbols without exceeding an acceptable level of distortion.
- Two principles are utilized:
 1. Properties of the information source that are known a priori result in redundant information that need not be transmitted ("redundancy reduction").
 2. The human observer does not perceive certain deviations of the received signal from the original ("irrelevancy reduction").
- **Lossless** coding: completely reversible, exploit 1. principle only
- **Lossy** coding: not reversible, exploit 1. and 2. principle

from: Girod



Entropy of a Memoryless Source

- Let a memoryless source be characterized by an ensemble U_0 with:

Alphabet $\{ a_0, a_1, a_2, \dots, a_{K-1} \}$
Probabilities $\{ P(a_0), P(a_1), P(a_2), \dots, P(a_{K-1}) \}$

- Shannon: information conveyed by message " a_k ":

$$I(a_k) = -\log(P(a_k))$$

- "Entropy of the source" is the average information contents:

$$H(U_0) = E\{I(a_k)\} = -\sum_{k=0}^{K-1} P(a_k) * \log(P(a_k))$$

- For „log“ = „log₂“ the unit is bits/symbol

from: Girod



Entropy and Bit-Rate

- Properties of entropy:

$$H(U_0) \geq 0$$

$$\max \{ H(U_0) \} = \log K \text{ with } P(a_j) = P(a_k) \text{ for all } j, k$$

- The entropy $H(U_0)$ is a lower bound for the average word length I_{av} of a decodable variable length code
- Conversely, the average word length I_{av} can approach $H(U_0)$, if sufficiently large blocks of symbols are encoded jointly.
- Redundancy of a code:

$$\rho = I_{av} - H(U_0) \geq 0$$

from: Girod



Encoding with Variable Word Lengths

- A code without redundancy, i.e.

$$\lambda_{av} = H(U_0)$$

is achieved, if all individual code word lengths

$$\lambda_{cw}(a_k) = -\log(P(a_k))$$

- For binary code words, all probabilities would have to be binary fractions:

$$P(a_k) = 2^{-\lambda_{cw}(a_k)}$$

from: Girod



Redundant Codes: Example

a_i	$P(a_i)$	redundant code	optimum code
a_1	0.500	00	0
a_2	0.250	01	10
a_3	0.125	10	110
a_4	0.125	11	111
$H(U_0) = 1.75 \text{ bits}$		$\lambda_{av} = 2 \text{ bits}$ $\rho = 0.25 \text{ bits}$	$\lambda_{av} = 1.75 \text{ bits}$ $\rho = 0 \text{ bits}$



Variable Length Codes

- Unique decodability: Where does each code word start or end
- Insert start symbol: 01.0.010.1. wasteful
- Construct prefix-free code
- Kraft Inequality: test for uniquely decodable codes

Uniquely decodable code exists if
$$\zeta = \sum_{k=0}^{K-1} 2^{-\lambda_{cw}(a_k)} \leq 1$$

- Application:

a_i	$P(a_i)$	$-\log_2(P(a_i))$	Code A	Code B
a_1	0.5	1	0	0
a_2	0.2	2.32	01	10
a_3	0.2	2.32	10	110
a_4	0.1	3.32	111	111

Not uniquely decodable

$$\zeta = 1.125$$

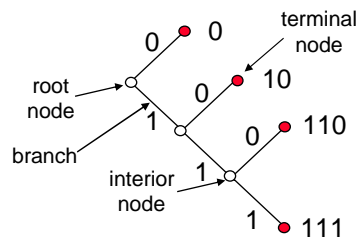
$$\zeta = 1$$

Uniquely decodable



Prefix-Free Codes

- Prefix-free codes are instantaneously and uniquely decodable
- Prefix-free codes can be represented by trees



- Terminal nodes may be assigned code words
 - Interior nodes cannot be assigned code words
 - For binary trees: N terminal nodes: $N-1$ interior nodes
- Code 0, 01, 11 is not a prefix-free code and uniquely decodable but: non-instantaneous



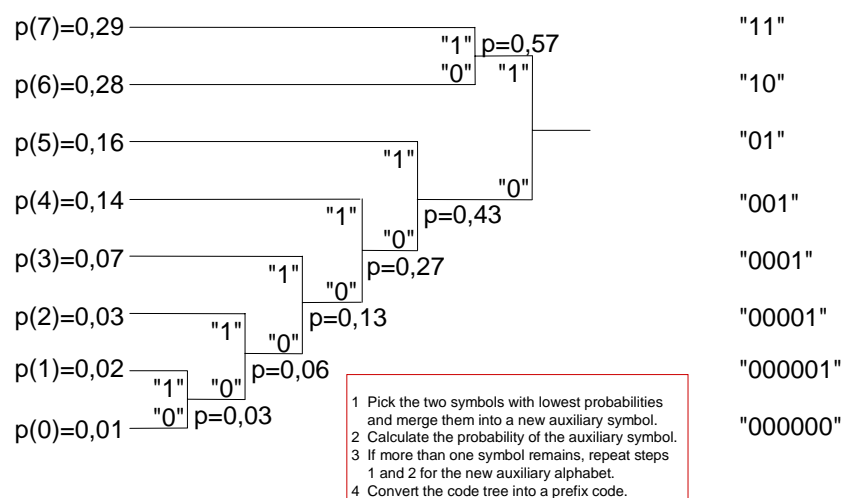
Huffman Code

- Design algorithm for variable length codes proposed by D. A. Huffman (1952) always finds a code with minimum redundancy.
- Obtain code tree as follows:

- Pick the two symbols with lowest probabilities and merge them into a new auxiliary symbol.
- Calculate the probability of the auxiliary symbol.
- If more than one symbol remains, repeat steps 1 and 2 for the new auxiliary alphabet.
- Convert the code tree into a prefix code.



Huffman Code: Example



Joint Sources

- Joint sources generate N symbols simultaneously.
- A coding gain can be achieved by encoding those symbols jointly.
- The lower bound for the average code word length is the joint entropy:

$$H(U_1, U_2, \dots, U_N) = -\sum_{u_1} \sum_{u_2} \dots \sum_{u_N} P(u_1, u_2, \dots, u_N) \cdot \log(P(u_1, u_2, \dots, u_N))$$

- It generally holds that

$$H(U_1, U_2, \dots, U_N) \leq H(U_1) + H(U_2) + \dots + H(U_N)$$

with equality, if U_1, U_2, \dots, U_N are statistically independent.

from: Girod



Markov Process

- Neighboring samples of the video signal are **not** statistically independent:

Source with memory

$$P(u_T) \neq P(u_T | u_{T-1}, u_{T-2}, \dots, u_{T-N})$$

- A source with memory can be modeled by a Markov random process.
- Conditional probabilities of the source symbols u_T of a Markov source of order N :

$$P(u_T | Z_T) = P(u_T | u_{T-1}, u_{T-2}, \dots, u_{T-N})$$

state of the Markov source at time T

from: Girod



Entropy of Source with Memory

- Markov source of order N : conditional entropy

$$\begin{aligned} H(U_T | Z_T) &= H(U_T | U_{T-1}, U_{T-2}, \dots, U_{T-N}) \\ &= -E \{ \log(p(u_T | u_{T-1}, u_{T-2}, \dots, u_{T-N})) \} \\ &= -\sum_{u_T} \dots \sum_{u_{T-N}} p(u_T, u_{T-1}, u_{T-2}, \dots, u_{T-N}) \log(p(u_T | u_{T-1}, u_{T-2}, \dots, u_{T-N})) \end{aligned}$$

$$H(U_T) \geq H(U_T | Z_T)$$

(equality for memoryless source)

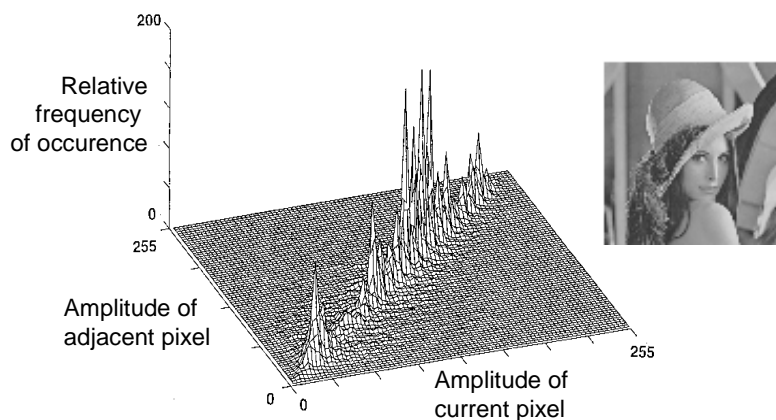
- Average code word length can approach $H(U_T | Z_T)$ e.g. with a switched Huffman code
- Number of states for an 8-bit video signal:

$$\begin{aligned} N=1 &\implies 256 \text{ states} \\ N=2 &\implies 65536 \text{ states} \\ N=3 &\implies 16777216 \text{ states} \end{aligned}$$

from: Girod



Second Order Statistics for Luminance Signal

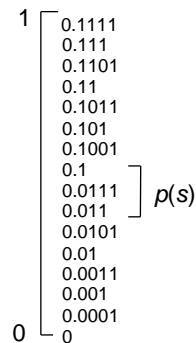


Histogram of two horizontally adjacent pels
(picture: female head-and-shoulder view)



Arithmetic Coding

- Universal entropy coding algorithm for strings
- Representation of a string by a subinterval of the unit interval $[0,1)$
- Width of the subinterval is approximately equal to the probability of the string $p(s)$



- Interval of width $p(s)$ is guaranteed to contain one number that can be represented by b binary digits, with

$$-\log(p(s)) + 1 \leq b \leq -\log(p(s)) + 2$$

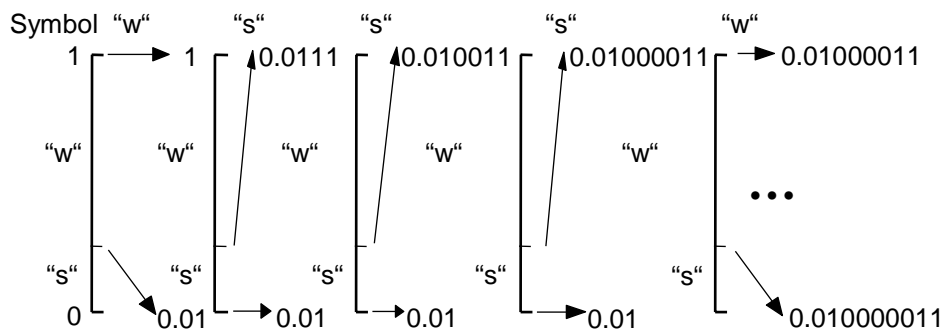
- Each interval can be represented by a number which needs 1 to 2 bits more than the ideal code word length

from: Girod



Arithmetic Coding: Probability Intervals

- Random experiment: pmf $p("s") = (0.01)_b$ and $p("w") = (0.11)_b$

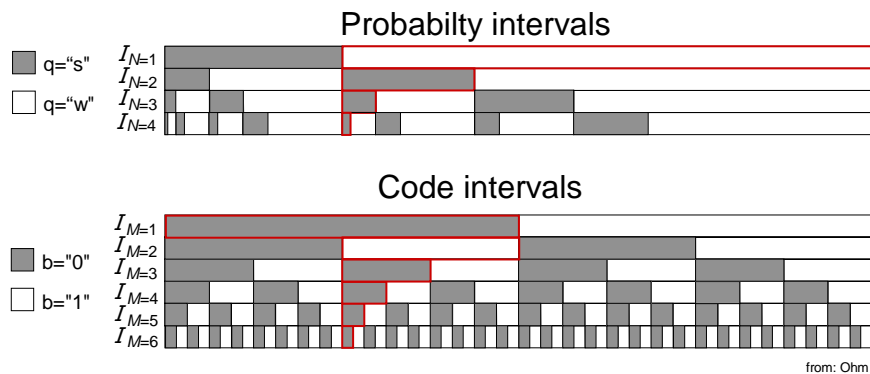


- Multiplications and additions with (potentially) very long word length
- Universal coding: probabilities can be changed on the fly:
e.g., use $p("s" | "s")$, $p("s" | "w")$, $p("w" | "s")$, $p("w" | "w")$



Arithmetic Encoding and Decoding

- Encoding: "w", "s", "s", "s" → 010000
- Decoding: 010 → "w", "s"

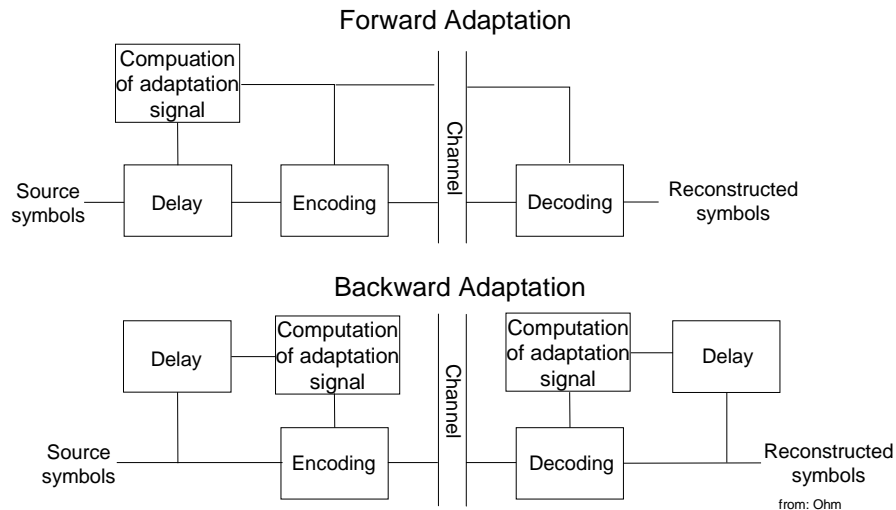


Adaptive Entropy Coding

- For non-adaptive coding methods: pdf of source must be known a priori (inherent assumption: stationary source)
- Image and video signals are not stationary: sub-optimal performance
- Solution: adaptive entropy coding
- Two basic approaches to adaptation:
 1. Forward Adaptation
 - Gather statistics for a large enough block of source symbols
 - Transmit adaptation signal to decoder as side information
 - Drawback: increased bit-rate
 2. Backward Adaptation
 - Gather statistics simultaneously at coder and decoder
 - Drawback: error resilience
- Combine the two approaches and circumvent drawbacks (Packet based transmission systems)



Forward vs. Backward Adaptive Systems



Summary

- Shannon's information theory vs. practical systems
- Source coding principles: redundancy & irrelevancy reduction
- Lossless vs. lossy coding
- Redundancy reduction exploits the properties of the signal source.
- Entropy is the lower bound for the average code word length.
- Huffman code is optimum entropy code.
- Huffman coding: needs code table.
- Arithmetic coding is a universal method for encoding strings of symbols.
- Arithmetic coding does not need a code table.
- Adaptive entropy coding: gains for sources that are not stationary

