



Grundlagen der Multimediatechnik - Tutorium

3. Besprechung Übungsblatt02

Stephan Amann

Universität Tübingen

18.11.2021

was wir heute vorhaben

- ▶ Quiz
- ▶ Besprechung Übungsblatt 2
- ▶ Tipps Übungsblatt 3
- ▶ Fragen
- ▶ Kodierungen

Vorlesung Recap

► Was ist ein Ton?

Vorlesung Recap

- ▶ **Was ist ein Ton?**
- ▶ Schallwelle
 - Amplitude = Lautstärke
 - Frequenz = Tonhöhe

Vorlesung Recap

- ▶ **Erkläre Maskierungseffekte.**

Vorlesung Recap

- ▶ **Erkläre Maskierungseffekte.**
- ▶ Überlagerung eines Tons durch "stärkeren" Ton.
 - **zeitliche Maskierung:** starker Wechsel der Lautstärke verschluckt aufeinanderfolgende Töne (Böller)
 - **Frequenzmaskierung:** dominante Töne (höhere Intensität) verdecken schwache Töne (Bass Überlagerung)

Vorlesung Recap

- ▶ **Was ist das Psychoakustische Modell?**

Vorlesung Recap

- ▶ **Was ist das Psychoakustische Modell?**
 - definiert den Frequenzbereich, den der Mensch hören kann
 - teilt diesen in Frequenzbänder auf und definiert Hörschwelle
- ▶ Verhältnis von Schallereignissen und den jeweiligen Höreindrücken

Vorlesung Recap

- ▶ **Nenne die Formel der Entropie, nenne auch deren Bestandteile.**

Vorlesung Recap

- ▶ **Nenne die Formel der Entropie einer diskreten, gedächtnislosen Quelle X . Nenne auch deren Bestandteile.**
- ▶ Bestandteile
 - Quelle $X = \{x_1, x_2, \dots, x_n\}, n \in \mathbb{N}_0$
 - Wahrscheinlichkeit eines Zeichens: $P(x_i) = -\log_2(x_i) = p_i$
 - Informationsgehalt eines Zeichens: $I(P(x_i)) = -\log_2(p_i) \text{ bit}$
- ▶ Entropie

$$H(X) = \sum_{i=0}^n P(x_i) \cdot I(P(x_i)) = - \sum_{i=0}^n p_i \cdot \log_2(p_i) \text{ bit}$$

Vorlesung Recap

- ▶ Wieso kann die Einheit *bit* durch eine Kommazahl dargestellt werden?

Vorlesung Recap

- ▶ Wieso kann die Einheit *bit* durch eine Kommazahl dargestellt werden?
- ▶ !!! *Bit* \neq *bit* !!!
 - Bits werden durch 0 und 1 repräsentiert (Speicher)
 - *bit* = *shannon* \leftarrow Einheit für Informationsgehalt

▶ Übungsblatt 02: Besprechung

▶ Aufgabe 1

- Zahlen dürfen auch sinnvoll gerundet werden (keine ewig langen Dezimalzahlen).
- ABER: $I(P(B)) = -\log_2(0.2) \approx 2.322$
- Zu jeder Rechnung gehört eine Formel!
- in 1.2 a) und b): p_i definieren!

▶ Aufgabe 2

- Bei Schwierigkeiten mit Coding oder Verständnis, mich oder andere fragen!
- WhatsApp oder Discord Gruppe für dieses Tutorium?

Tipps zu Übungsblatt 03

► Aufgabe 1: LZW-Kodierung

- *bekommt ihr hin!*

► Aufgabe 2: RLE

- Foliensatz *03 Textkomprimierung* ab Folie 32
- Theoriefragen dürfen auch im Notebook beantwortet werden.
- Achtung! RLE Format: Zeichen Zähler, Bsp.: A5
- **KEINE Bibliotheken!**

► Aufgabe 3: Burrows-Wheeler-Transformation

- Foliensatz *03 Textkomprimierung* ab Folie 59
- **KEINE Bibliotheken!**
- Letzte Frage wieder Theorie!

Fragen?

RLE

- ▶ Führt Run-Length-Encoding aus:
- ▶ **GGGTTTSSSSTTTTTTOLLLL**
- ▶ **UUNNIITTUUEEBBIINNGGEENN**
 - Waren die Kompressionen sinnvoll?

RLE

- ▶ Führt Run-Length-Encoding aus:
- ▶ **GGGMTTIISSSSTTTTTTOLLLL**
 - G3M1T2I1S4T5O1L4
 - sinnvolle KOverpression
- ▶ **UUNNIITTUUEEBBIINNGGEENN**
 - U2N2I2T2U2E2B2I2N2G2E2N2
 - nicht sinnvolle Kompression

LZW-Kodierung

- ▶ Führt die LZW-Kodierung aus:
- ▶ **ROKOKO**
 - Tabellenform einhalten!

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- ▶ *mit 26 weiter machen.*

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			"R"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 ("R")	" O"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 (" R")	" O"
K (10)	("OK", 27)	14 (" O")	" K"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 (" R")	" O"
K (10)	("OK", 27)	14 (" O")	" K"
O (14)	("KO", 28)	10 (" K")	" O"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 (" R")	" O"
K (10)	("OK", 27)	14 (" O")	" K"
O (14)	("KO", 28)	10 (" K")	" O"
K (10)			" OK"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 (" R")	" O"
K (10)	("OK", 27)	14 (" O")	" K"
O (14)	("KO", 28)	10 (" K")	" O"
K (10)			" OK"
O (14)	("OKO", 29)	27 (" OK")	" O"

LZW-Kodierung

LZW-Kodierung: ROKOKO

Lesezeichen	Neuer Wörterbucheintrag	Ausgabe	akt. Puffer
R (17)			" R"
O (14)	("RO", 26)	17 (" R")	" O"
K (10)	("OK", 27)	14 (" O")	" K"
O (14)	("KO", 28)	10 (" K")	" O"
K (10)			" OK"
O (14)	("OKO", 29)	27 (" OK")	" O"
EOF		14 (" O")	

Burrows-Wheeler-Transformation

- ▶ Führt die Burrows-Wheeler-Transformation aus:
- ▶ **ROKOKO**
 - Inklusive Rücktransformation!

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O
O	R	O	K	O	K

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O
O	R	O	K	O	K
K	O	R	O	K	O

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O
O	R	O	K	O	K
K	O	R	O	K	O
O	K	O	R	O	K

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O
O	R	O	K	O	K
K	O	R	O	K	O
O	K	O	R	O	K
K	O	K	O	R	O

Burrows-Wheeler-Transformation

1. Zyklische Permutation des Eingabestrings

String: ROKOKO

R	O	K	O	K	O
O	R	O	K	O	K
K	O	R	O	K	O
O	K	O	R	O	K
K	O	K	O	R	O
O	K	O	K	O	R

Burrows-Wheeler-Transformation

2. Matrix sortieren

↓

K	O	K	O	R	O
K	O	R	O	K	O
O	K	O	K	O	R
O	K	O	R	O	K
O	R	O	K	O	K
R	O	K	O	K	O

Output:

Burrows-Wheeler Transformation

2. Matrix sortieren

↓	K	O	K	O	R	O
	K	O	R	O	K	O
	O	K	O	K	O	R
	O	K	O	R	O	K
	O	R	O	K	O	K
	R	O	K	O	K	O

↓ Output: OORKKO

Burrows-Wheeler-Transformation

2. Matrix sortieren

↓	K	O	K	O	R	O
	K	O	R	O	K	O
	O	K	O	K	O	R
	O	K	O	R	O	K
	O	R	O	K	O	K
	R	O	K	O	K	O

← Index = 5

↓ Output: OORKKO mit Index 5

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: R

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: R**O**

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: ROK

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: ROKO

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: ROKOK

Burrows-Wheeler-Transformation

Rücktransformation

- ▶ Start bei Index 5 in sortierte Codierung:

Codierung	O	O	R	K	K	O
Position	0	1	2	3	4	5
sortierte Codierung	K	K	O	O	O	R
sortierte Position	3	4	0	1	5	2

Output: ROKOKO