

# Datenkompression: Überblick und Grundlagen

H. Fernau

email: [fernau@uni-trier.de](mailto:fernau@uni-trier.de)

WiSe 2008/09  
Universität Trier

# Datenkompression



1.  $|\mathcal{X}_c| < |\mathcal{X}|$ .
2.  $\mathcal{X} = \mathcal{Y}$  (verlustfrei)     $\mathcal{X} \approx \mathcal{Y}$  (verlustbehaftet).

## Datenkompression

Drei einführende Motivationen / Beispiele:

- Speicherersparnis
- Übertragungsgeschwindigkeit
- Netzbelastung

**Beispiel 1 Kapazität:** Was passt wo drauf ?

Auflösung:	720×480 Pixel
Farbtiefe:	2 Bytes
Bildwiederholfrequenz:	30 fps (frames per second)
Länge:	2 Stunden

	720	
×	480	
	345600	Pixels
×	2	Bytes
	691200	Bytes je Bild
×	30	fps
	20736000	Bps (Bytes per second)
×	7200	s
	142383 MB	≈ 139 GB

Es werden **219 CDs** benötigt, um das Video auf CD-ROMs mit einer Kapazität von **650 MB** zu speichern (eine CD kann also knapp **33 s** Video speichern).

## **Beispiel 2** Zeit und [Geschwindigkeit bei der Datenübertragung](#).

### **Faxgerät**

Seitengröße:  $8,5 \times 11 \text{ inch} = 93,5 \text{ inch}^2$

Abtasten mit 200 dpi:  $93,5 \times 200^2 = 3740000 \text{ b (bits)}$

Ein Modemgerät mit einer Übertragungsrates von 14,4 kbps benötigt damit

$$3740 \text{ kb} / 14,4 \text{ kbps} = 4 \text{ min } 20 \text{ s}$$

um eine solche Seite zu senden. Eine komprimierte Seite ( $\approx 250 \text{ kb}$ ):

$$250 \text{ kb} / 14,4 \text{ kb/s} = 17 \text{ s.}$$

**Video**  $20736000 \text{ B} = 19,775 \text{ MB}$  pro Sekunde.

Das heißt, um ein Video online zu senden, braucht man eine Verbindung mit einer Bandbreite von mehr als **165 Mbps!** Ein Video im MPEG2 Format lässt sich bereits mit **3 Mbps** online übertragen.

**Beispiel 3** Drucken übers Firmennetz.

Ein wesentlicher Teil der Bandbreite von Intranetzen wird (immer noch) durch Druckjobs ver(sch)wendet.

Quelle: ComputerZeitung Nr 26 vom 26.Juni 2006

**Grund:** "Klassischerweise" wird die häufig komprimierte Datei noch vom Rechner von einem sog. *Druckerspooles* in ein dem Drucker verständliches Format umgesetzt.

Ein typischer Wert: PDF-Datei von 235 KB ergibt 2,18 MB Last fürs Netz.  
Idee (bislang nicht umgesetzt): Dekomprimierung erfolgt erst im Drucker, womöglich sogar Extra-Komprimierung für die Datenübertragung.

Ausgangs- dokumentart	Ausgangsgröße	Druckjob (Rohformat)	Druckjob (extra komprimiert)
.doc	1,17 MB	11,2 MB	0,38 MB
.pdf	1,06 MB	3,23 MB	0,41 MB
.ppt	1,67 MB	218 MB	1,03 MB

## Konkrete Anwendungsübersicht

Anwendung	Unkomprimiert	Komprimiert
Sprache <i>8 ksamples/s, 8 bits/sample</i>	64 kbps	2-4 kbps
Video (langsame Bewegung) <i>Bildauflösung 176×120, 24 bits/pixel</i>	5.07 Mbps	8-16 kbps
Audio-Konferenz <i>8 ksamples/s, 16 bits/sample</i>	128 kbps	6-64 kbps
Video-Konferenz <i>Bildauflösung 352×240, 24 bits/pixel</i>	30.41 Mbps	64-768 kbps
Digitale Stereo-Audio <i>44.1 ksamples/s, 16 bits/sample</i>	1.5 Mbps	128-768 kbps
Video Dateiübertragung (15 fps) <i>Bildauflösung 352×240, 24 bits/pixel</i>	30.41 Mbps	384 kbps
Digitales Video auf CD-ROM (30 fps) <i>Bildauflösung 352×240, 24 bits/pixel</i>	60.83 Mbps	1.5-4 Mbps
Broadcast Video (30 fps) <i>Bildauflösung 720×480, 24 bits/pixel</i>	248.83 Mbps	3-8 Mbps
HDTV (59.94 fps) <i>Bildauflösung 1280×720, 24 bits/pixel</i>	1.33 Gbps	20 Mbps

Quelle: V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards; Algorithms and Architectures*, 2. Auflage, Kluwer, 1998

## Begriffe

*Der Kompressionsquotient* =  $\frac{\# \text{ Bits von } \mathcal{X}_c}{\# \text{ Bits von } \mathcal{X}}$ .

Kompressionsquotient:

**bpb** bit per bit;

**bpc/bps** bit per character/symbol;

**bpp** bit per pixel.

610856	book		1.0
368423	book.z	<- pack (Huffman)	0.60
250759	book.Z	<- compress (LZW)	0.41
206686	book.gz	<- gzip (LZ77)	0.34
3405108	book.ps		1.0
2076437	book.ps.z	<- pack (Huffman)	0.61
1195513	book.ps.Z	<- compress (LZW)	0.35
879613	book.ps.gz	<- gzip (LZ77)	0.26

**Frage:** *Ist Datenkompression überhaupt möglich?*

Kompression → Informationsverlust.

**Frage:** *Warum funktioniert Datenkompression in der Praxis hervorragend? [DVD, MPEG, MP3, MPEG/audio, ...]*

- Die einzelnen Zeichen tauchen *nicht* mit derselben Wahrscheinlichkeit auf.
- Die Abhängigkeiten der aufeinander folgenden Ereignisse.
- Information ist nicht *zufällig* sondern enthält vielmehr zahllose Regelmäßigkeiten.

Kompression → Reduktion der Redundanz

- Oft kann auf eine exakte Rekonstruierbarkeit des Originals verzichtet werden (menschliche Wahrnehmung).

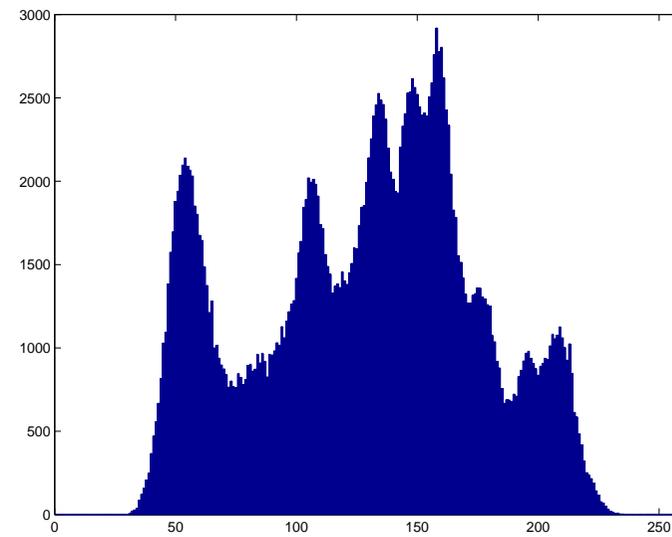
## Charakteristische Häufigkeiten (in %) und von Buchstaben in Texten englischer Sprache

(nach Randy Nichols, *Classical Cryptography Course*):

12	10	8	8	7	7	7	6	5	4-3	2	1	< 1
E	T	A	O	N	I	S	R	H	LDCU	PFMW	YBGV	KQXJZ

In Gruppen zusammengefasst:

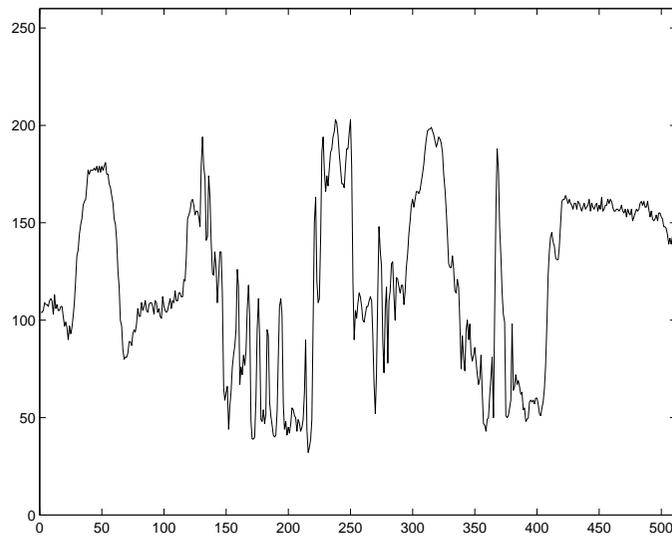
A E I O U	38.58%
L N R S T	33.43%
J K Q X Z	1.11%
E T A O N	45.08%
E T A O N I S R H	70.02%



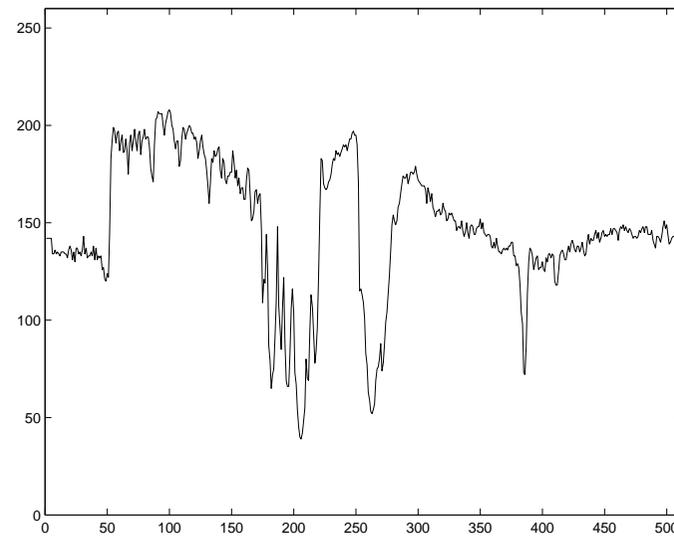
Histogramm des  
Lena-Bildes

**Trigramme** (die häufigsten aus einem deutschen Text mit 60046 Zeichen)

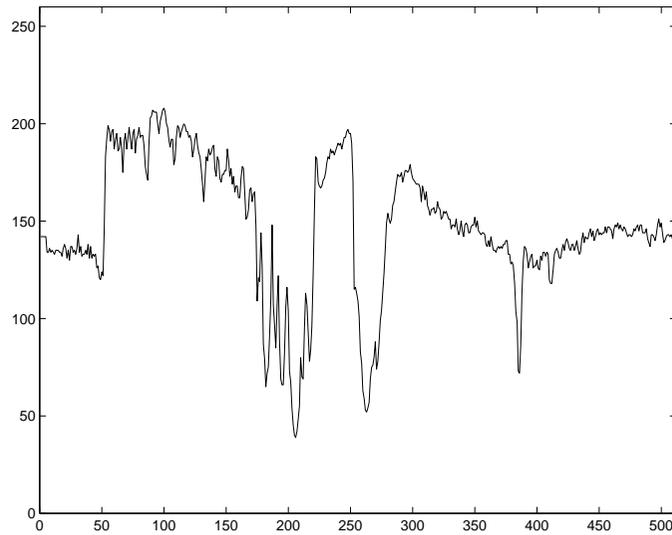
SCH-	666	ERE-	313	NEN-	198	AUS-	162	IST-	142	HRE-	124
DER-	602	ENS-	270	SSE-	191	TIS-	159	STA-	141	HER-	122
CHE-	599	CHT-	264	REI-	190	BER-	157	DES-	140	ACH-	119
DIE-	564	NGE-	263	TER-	188	ENI-	157	FUE-	139	GES-	118
NDE-	541	NDI-	259	REN-	185	ENG-	155	NTE-	139	ABE-	117
EIN-	519	IND-	254	EIT-	184	ION-	154	UER-	138	ERA-	117
END-	481	ERD-	248	EBE-	178	SEN-	152	ERU-	137	BEN-	116
DEN-	457	INE-	247	ENE-	175	ITI-	151	TUN-	136	MEN-	115
ICH-	453	AND-	246	LIC-	175	AUF-	149	SEI-	133	RIE-	112
TEN-	425	RDE-	239	EGE-	173	IES-	149	ESE-	132	VER-	110
UNG-	377	ENA-	214	DAS-	172	ASS-	148	ERT-	128	LAN-	109
HEN-	332	ERS-	212	ENU-	171	ENW-	148	NDA-	127	ENB-	108
UND-	331	EDE-	209	NUN-	169	ENT-	146	IED-	126	ESS-	108
GEN-	321	STE-	205	NER-	166	ERI-	143	ERN-	125	LLE-	108
ISC-	317	VER-	204	RUN-	163	EST-	142	NAU-	108	TSC-	107
ENN-	106	ERG-	106	RIT-	106	EHR-	105	CHA-	104	VON-	104
SIC-	103	IGE-	102	ITE-	101	ENZ-	100	ERB-	100	EUT-	100



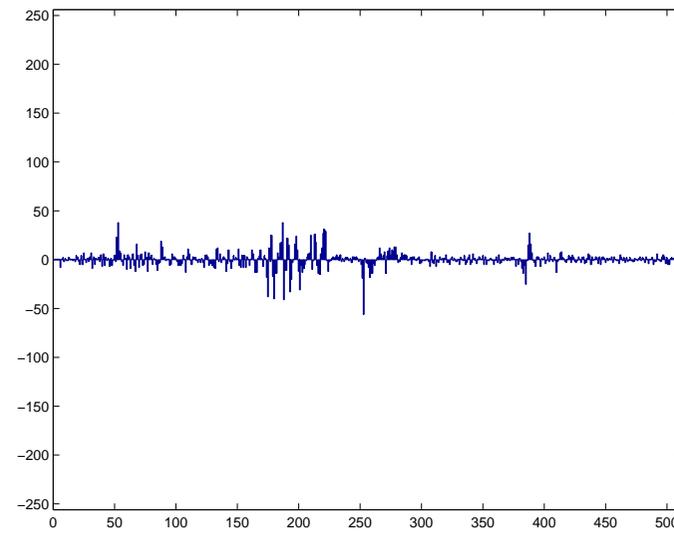
Zeilenprofil (row=256)



Spaltenprofil (col=256)

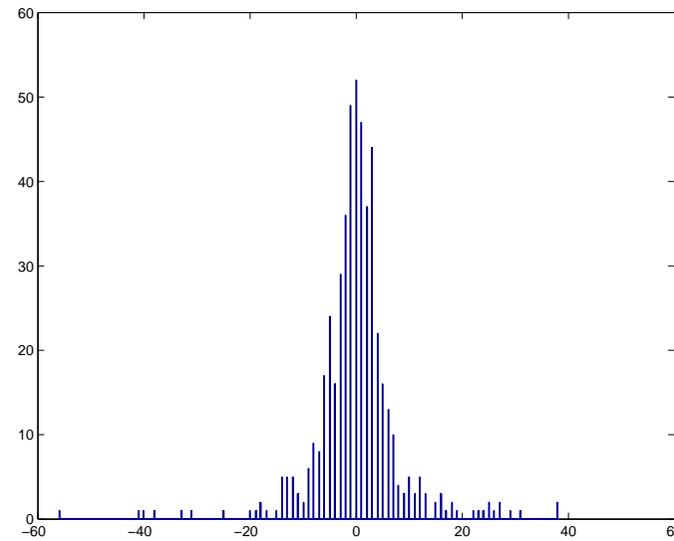


Profil der 256. Spalte



und entspr. Differenzen

Histogramm der  
Differenzen



## Redundanz der natürlichen Sprachen

Wird die Redundanz reduziert ist das Lesen viel mühsamer  
WIRD DIE REDUNDANZ REDUZIERT IST DAS LESEN VIEL MÜHSAMER  
WIRDDIEREDUNDANZREDUZIERTISTDASLESENVIELMÜHSAMER  
WI DD ER DU DA ZR DU IE TI TD SL SE VI LM HS ME

$H$  (inkl. Zwischenraum) der deutschen Sprache  $\approx 4.9$  bit.  
Berücksichtigt man jedoch die Häufigkeiten der Trigramme, so erhält man

$$H_3 \approx 1.6 \text{ bit.}$$

Die Redundanz der deutschen Schriftsprache ist somit

$$\geq 4.9 \text{ bit} - 1.6 \text{ bit} = 3.3 \text{ bit.}$$

$\leadsto$  Ein Text wäre auch dann noch lesbar, wenn jedes zweite Zeichen fehlte.

**Beispiel** Reduktion der Redundanz.

Betrachten wir eine Folge :  $x_1, x_2, x_3, x_4, \dots$

9 11 11 11 14 13 15 17 16 17 20 21 ...

Das Modell:  $\hat{x}_n = n + 8 \quad n = 1, 2, \dots$

9 10 11 12 13 14 15 16 17 18 19 20 ...

plus Korrektur (Differenz zwischen dem Modell und  $x_1, x_2, x_3, x_4, \dots$ )

$$e_n = x_n - \hat{x}_n \quad 0 \ 1 \ 0 \ -1 \ 1 \ -1 \ 0 \ 1 \ -1 \ -1 \ 1 \ 1 \ \dots$$

Kompressionsquotient: 2 bps.

## Maße für die Qualität der DK-Verfahren

- Kompressionsquotient,
- Geschwindigkeit der Komprimierung,
- Geschwindigkeit der Dekomprimierung,
- *Entstellung* oder *Verzerrung* (engl. distortion):  $\mathcal{Y} - \mathcal{X}$ .



Bild	Bit Map	JPEG
Brücke	65 536	17 272
Kamera	65 536	10 889

## Typische Verzerrungsmaße

quadratische Fehlermaß

$$d(x_n, y_n) = (x_n - y_n)^2$$

Betragsfehlermaß

$$d(x_n, y_n) = |x_n - y_n|$$

mittlerer quadratischer Fehler

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - y_n)^2$$

mittlerer Betragsfehler

$$d_1 = \frac{1}{N} \sum_{n=1}^N |x_n - y_n|$$

Verhältnis von Signal zu Verzerrung

$$SNR = \frac{\sigma_X^2}{\sigma^2}$$

SNR: signal-to-noise,  $\sigma_X^2$ : der MSE des Signals

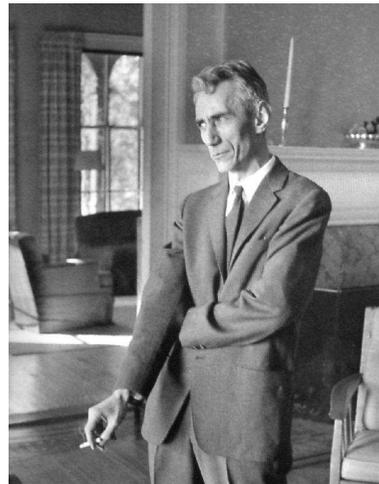
SNR logarithmisch skaliert [Dezibel]

$$SNR(dB) = 10 \log_{10} \frac{\sigma_X^2}{\sigma^2}$$

## Ein bisschen Informationstheorie

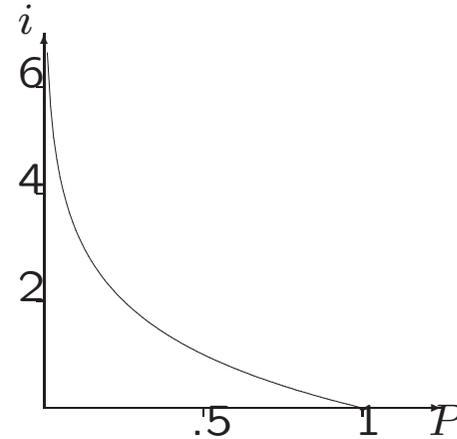
Hinweis: Unregelmäßige Spezialveranstaltung zu diesem Thema

**Claude Elwood Shannon (1916–2001)**



Sei  $A$  ein Ereignis und  $P(A)$  die Wahrscheinlichkeit, dass  $A$  eintritt. *Informationsgehalt*

$$i(A) = \log_2 \frac{1}{P(A)}$$



Sei  $\mathcal{S}$  eine bestimmte Informationsquelle. Die *Entropie* ist der *mittlere Informationsgehalt pro Nachricht aus  $\mathcal{S}$* :

$$H(\mathcal{S}) = \mathbb{E}[i(X)] = \sum_{A \in \mathcal{S}} P(A) \cdot i(A)$$

Für  $P(A) = 0$  betrachten wir  $P(A) \cdot i(A)$  als 0 ( $\lim_{x \rightarrow 0} x \cdot \log \frac{1}{x} = 0$ ).

Wir werden meistens  $S$  als endliches Quellenalphabet

$$\Sigma = \{a_1, \dots, a_n\}$$

betrachten. Denn ist  $P(a_i)$  die Wahrscheinlichkeit, dass  $a_i$  eintritt und  $i(a_i) = \log_2 1/P(a_i)$  beschreibt die Anzahl von Bits für die Codierung von  $a_i$ . Die Entropie:

$$H(\Sigma) = \sum_{i=1}^n P(a_i) \cdot \log \frac{1}{P(a_i)}$$

ist die mittlere Anzahl von Bits, um die Nachricht aus  $\Sigma$  zu codieren.

**Beispiel** Sei  $\Sigma = \{a, b, c, d, e\}$  mit

$$P(a) = P(b) = P(c) = 0.25 \quad \text{und} \quad P(d) = P(e) = 0.125.$$

Dann ist

$$H(\Sigma) = 3 \cdot 0.25 \cdot \log 4 + 2 \cdot 0.125 \cdot \log 8 = 1.5 + 0.75 = 2.25.$$

## Weitere Beispiele

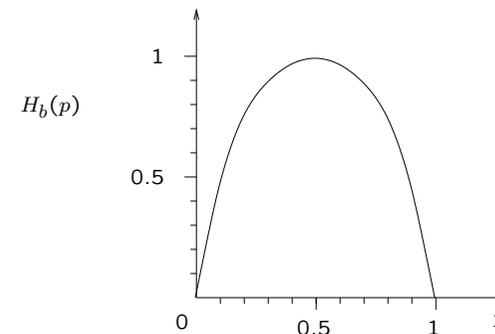
	$P(a)$	$P(b)$	$P(c)$	$P(d)$	$P(e)$	$H$
1.	0.2	0.2	0.2	0.2	0.2	2.322
2.	0.5	0.25	0.125	0.0625	0.0625	1.875
3.	0.75	0.0625	0.0625	0.0625	0.0625	1.3
4.	0.94	0.01	0.01	0.01	0.01	0.322

Die Entropie wird für ungleichmäßigere Verteilungen kleiner.

Die binäre Entropie-Funktion

$$H_b(p) = p \log \frac{1}{p} + (1 - p) \log \frac{1}{1-p}$$

bezeichnet die Entropie-Funktion für zwei Quellensymbole mit Wahrscheinlichkeiten  $p$  und  $1 - p$ .



## Weitere Eigenschaften der Entropie:

- $H$  ist symmetrisch, das heißt für jede Permutation

$$\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\} \quad \text{gilt}$$

$$H(p_1, p_2, \dots, p_n) = H(p_{\pi(1)}, p_{\pi(2)}, \dots, p_{\pi(n)})$$

wobei  $H(p_1, p_2, \dots, p_n)$  die Entropie für das Quellenalphabet  $\Sigma = a_1, a_2, \dots, a_n$  mit  $P(a_1) = p_1, \dots, P(a_n) = p_n$  bezeichnet.

- Untere und obere Grenze für  $H$ :

$$0 = H(1, 0, \dots, 0) \leq H(p_1, \dots, p_n) \leq H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = \log n$$

Unser Modell:  $\mathcal{S} = (\Sigma, \{P(a_1), \dots, P(a_n)\})$

→ in  $x_1, x_2, x_3, x_4, \dots$   $x_i, x_{i+1}$  unabhängig  $\forall i \geq 1$ .

Verallgemeinerung:  $(\Sigma = \{a_1, \dots, a_n\})$

$$G_2(\mathcal{S}) = - \sum_{i=1}^n \sum_{j=1}^n P(X_1 = a_i, X_2 = a_j) \log P(X_1 = a_i, X_2 = a_j),$$

und entsprechend die Entropie  $G_m$ , für  $m > 2$ .

↷ Entropie der Quelle  $\mathcal{S}$ :

$$H(\mathcal{S}) = \lim_{m \rightarrow \infty} \frac{1}{m} G_m.$$

**Beispiel** Komprimierungsschemen für Dateien mit Texten in Englisch.

Wir benutzen 96 druckfähige ASCII Zeichen.

Bei Annahme einer **Gleichverteilung**:

$$H(\mathcal{E}) = \log 96 = \mathbf{6,6 \text{ bps.}}$$

Verteilung: **Häufigkeiten** der Vorkommen von Zeichen in einer großen Sammlung englischer Texte.

$$H(\mathcal{E}) = \mathbf{4,5 \text{ bps.}}$$

Für die Codierung: *jedes Zeichen separat codiert wird*, ist die Huffman-Methode optimal: **4,7 bps.**

Die Codierung: Blöcke von Zeichen der Länge 8. Uniforme Verteilung:

$$G_8(\mathcal{E}) = \sum_B P(B) \log 96^8 = \log 96^8 = 8 \times 6,6 \text{ Bits je Block} = \mathbf{6,6 \text{ bps.}}$$

Die Verteilung, die der englischen Sprache entspricht:

$$G_8(\mathcal{E}) = 19 \text{ Bits je Block und das heißt } \mathbf{2,4 \text{ bps.}}$$

Eine Vermutung: die untere Schranke für die Entropie liegt bei **1,3 bps.**

## Benchmarks für die englische Sprache (Calgary-Corpus)

	Bytes	Beschreibung
bib	111261	Bibliographic files (refer format)
book1	768771	Hardy: Far from the madding crowd
book2	610856	Witten: Principles of computer speech
geo	102400	Geophysical data
news	377109	News batch file
obj1	21504	Compiled code for Vax: compilation of progp
obj2	246814	Compiled code for Apple Macintosh: Knowledge supp. system
paper1	53161	Witten, Neal and Cleary: Arithmetic coding for data compr.
paper2	82199	Witten: Computer (in)security
paper3	46526	Witten: In search of "autonomy"
paper4	13286	Cleary: Programming by example revisited
paper5	11954	Cleary: A logical implementation of arithmetic
paper6	38105	Cleary: Compact hash tables using bidirectional linear probing
pic	513216	Picture number 5 from the CCITT Facsimile test files (text and drawings)
progc	39611	C source code: compress version 4.0
progl	71646	Lisp source code: system software
progp	49379	Pascal source code: prediction by partial matching evaluation program
trans	93695	Transcript of a session on a terminal

## Was geht beim Calgary-Corpus ? (Daten in bps)

Method	bib	b1	b2	geo	news	ob1	ob2	p1	p2	pic	progc	progl	progp	trans	W.Av.
szip-b	1.96	2.35	2.02	4.28	2.48	3.77	2.47	2.49	2.43	0.80	2.50	1.72	1.77	1.54	2.07
ppmD5	1.89	2.34	1.98	4.96	2.42	3.70	2.35	2.36	2.34	0.95	2.40	1.69	1.72	1.50	2.08
bzip-9	1.95	2.40	2.04	4.48	2.51	3.87	2.46	2.46	2.42	0.77	2.50	1.72	1.71	1.50	2.09
szip	1.98	2.36	2.03	4.29	2.50	3.78	2.48	2.50	2.44	0.82	2.52	1.75	1.82	1.59	2.09
bzip-6	1.95	2.49	2.06	4.48	2.51	3.87	2.46	2.46	2.42	0.77	2.50	1.72	1.71	1.50	2.12
ppmD7	1.93	2.54	2.04	4.96	2.44	3.71	2.35	2.40	2.40	0.94	2.42	1.67	1.72	1.47	2.15
bzip2-9	1.97	2.42	2.06	4.45	2.52	4.01	2.48	2.49	2.44	0.78	2.53	1.74	1.74	1.53	2.11
bzip2-6	1.97	2.51	2.08	4.45	2.52	4.01	2.48	2.49	2.44	0.78	2.53	1.74	1.74	1.53	2.13
bzip-1	2.07	2.79	2.38	4.49	2.84	3.87	2.56	2.46	2.42	0.77	2.50	1.72	1.71	1.50	2.31
ppmD3	2.12	2.54	2.25	4.91	2.68	3.72	2.52	2.48	2.45	0.99	2.48	1.84	1.80	1.72	2.26
bzip2-1	2.10	2.81	2.40	4.47	2.85	4.01	2.59	2.49	2.44	0.78	2.53	1.74	1.74	1.53	2.33
ppmC-896	2.12	2.52	2.28	5.01	2.77	3.68	2.59	2.48	2.46	0.98	2.49	1.87	1.82	1.75	2.28
bred-r3	2.19	2.98	2.51	4.89	2.94	3.91	2.67	2.58	2.58	0.82	2.58	1.79	1.78	1.56	2.44
dmc-50M	2.20	2.48	2.19	4.80	2.77	4.12	2.76	2.73	2.59	0.82	2.75	1.99	2.00	1.92	2.26
dmc-16M	2.20	2.51	2.19	4.80	2.77	4.12	2.76	2.73	2.59	0.82	2.75	1.99	2.00	1.92	2.27
ppmCnx	2.22	2.58	2.35	5.44	2.91	3.87	2.72	2.62	2.57	1.02	2.64	1.96	1.92	1.82	2.37
dmc-5M	2.20	2.74	2.49	4.80	3.01	4.12	2.82	2.73	2.59	0.82	2.75	1.99	2.00	1.92	2.42
gzip-b	2.51	3.25	2.70	5.34	3.06	3.84	2.63	2.79	2.89	0.82	2.68	1.80	1.81	1.61	2.59
gzip-d	2.52	3.26	2.71	5.35	3.07	3.84	2.65	2.79	2.90	0.88	2.68	1.82	1.82	1.62	2.61
gzip-f	3.15	3.80	3.26	5.45	3.48	3.98	3.04	3.25	3.41	1.02	3.12	2.24	2.17	2.05	3.03
ppmC-56	3.51	3.78	3.34	5.42	3.93	3.87	3.09	3.37	3.35	0.91	3.21	2.11	2.13	2.68	3.11
ppmCnx-56	3.76	4.12	3.61	5.83	4.19	4.04	3.27	3.62	3.63	0.96	3.43	2.25	2.27	2.84	3.35
yabba-d	2.94	3.34	3.36	5.96	3.93	5.16	4.08	3.38	3.28	1.03	3.45	2.64	2.69	2.82	3.14
srank-d	3.82	4.06	3.53	6.25	3.97	4.76	3.71	3.69	3.69	1.14	3.60	2.68	2.82	2.86	3.40
compress	3.35	3.46	3.28	6.08	3.86	5.23	4.17	3.77	3.52	0.97	3.87	3.03	3.11	3.27	3.21
huffword2	4.00	3.10	2.97	9.06	4.11	7.99	5.72	4.24	3.67	2.03	4.55	3.02	3.08	3.69	3.55
lzw1	4.77	5.47	4.75	6.77	4.94	4.93	4.13	4.63	4.90	2.05	4.37	3.53	3.43	3.71	4.44
char	5.23	4.54	4.80	5.66	5.20	6.01	6.28	5.02	4.63	1.22	5.25	4.80	4.91	5.56	4.40
pack	5.24	4.56	4.83	5.69	5.23	6.08	6.30	5.03	4.65	1.66	5.26	4.81	4.91	5.58	4.49
yabba512	5.59	5.74	5.58	6.71	6.46	7.24	7.05	7.00	7.22	1.36	6.24	6.03	6.22	7.41	5.35
cat	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00

---

Datum	bps	Schema	Autoren
May 1977	3,94	LZ77	Ziv, Lempel
1984	3,32	LZMW	Miller and Wegman
1987	3,30	LZH	Brent
1987	3,24	MTF	Moffat
1987	3,18	LZB	Bell
.	2,71	GZIP	.
1988	2,48	PPMC	Moffat
.	2,47	SAKDC	Williams
.	2,47	PPMD	Howard
Nov 1993	2,34	PPMC	Moffat
Oct 1994	2,34	PPM*	Cleary, Teahan, Witten
18 Nov 1994	2,33	PPMD	Moffat
1995	2,29	BW	Burrows, Wheeler
31 Jan 1995	2,27	PPMD	Teahan
1997	1,99	BOA	

Jahr	Schema	Faktor
1980	CCITT Group 3, T.4	7,7
1984	CCITT Group 4, T.6	15,5
1988	IBM's Q-Coder	19,0
1991	2-level coding	21,4
1993	JBIG	19,7
1995	TIC	22,3

CCITT fax test images Benchmark (*CCITT* changed its name to *ITU-T* International Telecommunications Union)

## Die Vorlesung

### **Grundlegende Codes**

Präfixcodes, Shannon- Shannon-Fano- Huffman-Algorithmus;

### **Arithmetische Codes** JBIG, JPEG;

### **Wörterbuch-Techniken**

Statistische und Dynamische Verfahren,  
LZ77 (gzip), LZ78, LZW (compress, GIF);

### **Weitere Verfahren**

Laufängencodierung, ppm (prediction with partial match),  
Burrows-Wheeler-Transformation (BWT);

### **Verlustfreie Komprimierung von Bildern**

Bedingte Entropie, Markov-Modell, Facsimile Codierung (MH, MR),  
Fortschreitende Bildübertragung;

## **Verlustbehaftete Komprimierung – Grundlagen**

### **Skalarquantisierung**

Gleichquantisierer, adaptive Quantisierung (Jayant), Lloyd Verfahren;

### **Vektorquantisierung**

Lindo-Buzo-Gray-Verfahren (LBG)

### **Differentialcodierung**

Delta-Modulierung;

### **Teilbandcodierung**

Frequenzfilter, Filterbänke,

Wavelets, Shapiros EZW-Algorithmus, JPEG 2000;

### **Fraktale Codierung**

### **Transformcodierung und JPEG-Standard**

Spezielle Transformationen für Bildverarbeitung, KLT, DFT, DCT,  
JPEG;

### **Weitere Anwendungen: MPEG.**

## Literatur

1. K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, Inc., 2. Auflage, 2000.  
Mittlerweile gibt es eine dritte Auflage
2. Bovik A. Bovik, Ed., *Handbook of Image and Video Processing*, AP Series in Communication, Networking and Multimedia, AP, 2000.
3. Sal D. Salomon. *A Guide to Data Compression Methods*, Springer, 2002.



Fortschreitende Bildübertragung (8 × 8 Pixel)

## Vektorquantisierung

