Fehlerbehandlung

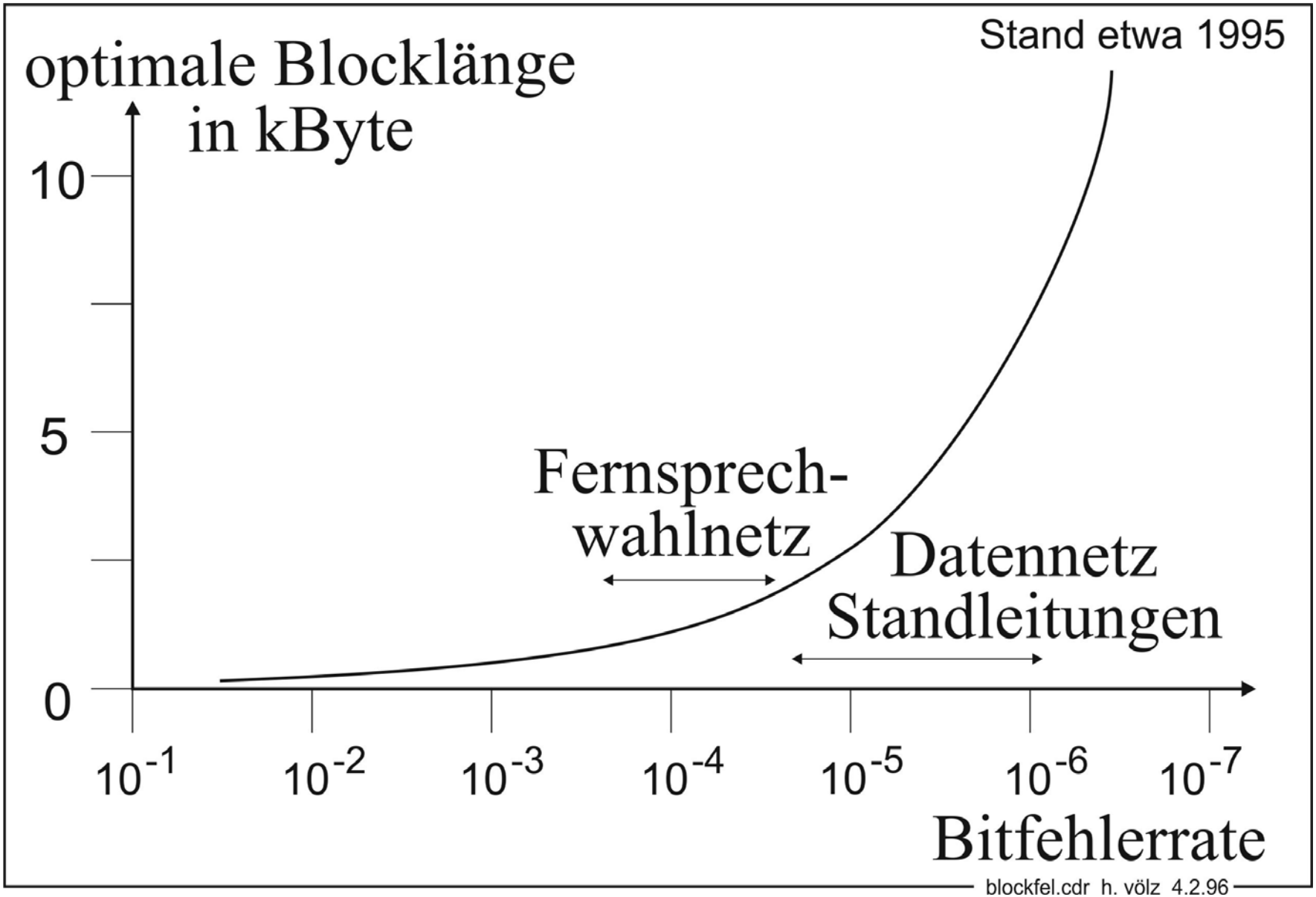
Bei jeder Übertragung (und Speicherung) treten immer mehrere störende Einflüsse auf.
Dennoch sollen empfangene Zeichen, Signale, Dateien usw. exakt mit den gesendeten übereinstimmen.

Beim ***kontinuierlichen*** Signal werden hauptsächlich Spektrum und Kurvenform verändert.
Ursachen: Kanal-Linearität, Rauschen, Störimpulse usw. sowie Störabstand und Dynamik geändert.
Deshalb kann hierbei das Empfangene nicht mit dem Gesendetem übereinstimmen.

Bei ***diskreten, digitalen*** Signalen ist es durch die zugelassenen Toleranzbereiche deutlich besser.
Dennoch gibt es andere Störwirkungen: Werte werden in andere umgewandelt, vertauscht.
Sind von *n* übertragenen Werten beim Empfang *m* falsch, so beträgt die ***Fehlerrate*** $f_R = m/n$.

Digitale Fehler sind oft recht einfach zu korrigieren. Sie müssen zunächst erkannt werden.
Das ist Aufgabe der Fehlererkennung (EDC error detection code).
Es genügt die Orte in der Datei zu bestimmen und Dort ist nur noch $0 \rightarrow 1$ bzw. $1 \rightarrow 0$ zu tauschen.
Das realisiert die sehr umfangreiche Fehlerkorrektur (ECC error correction code).
Dabei wird vorausgesetzt, dass keine ***Taktfehler*** auftreten.
Er muss daher zunächst auch aus einem fehlerbehafteten Signal wieder hergestellt werden (VCO).
Für alle Fehlerbehandlungen ist vorteilhaft, wenn in der Datei nur wenige Fehler auftreten.
Deshalb werden sie in passende Abschnitte (***Blöcke)*** zerlegt.
Die optimale Größe ist von der **Fehlerrate** abhängig, optimal sind 10^{-4} bis 10^{-5}. Gilt auch bei der Genetik.
Ein etwas älteres Beispiel zeigt das Bild: die Blockgröße ist annähernd exponentiell zur Fehlerrate
Eine weitere, vorteilhafte Maßnahme bei der Fehlerbehandlung ist das Spreizen (s. u).

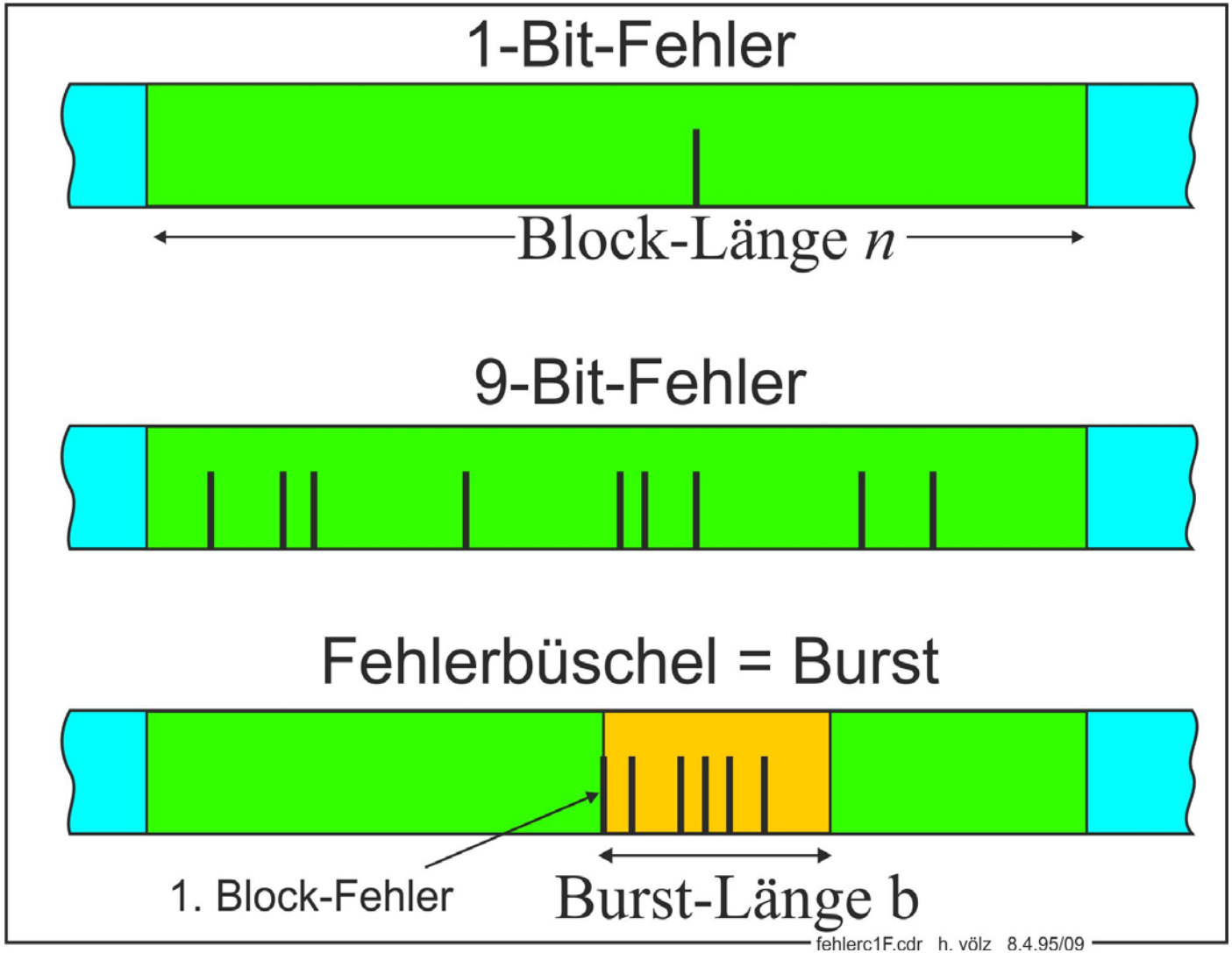
Blockgröße und Fehlerrate



Fehlertypen

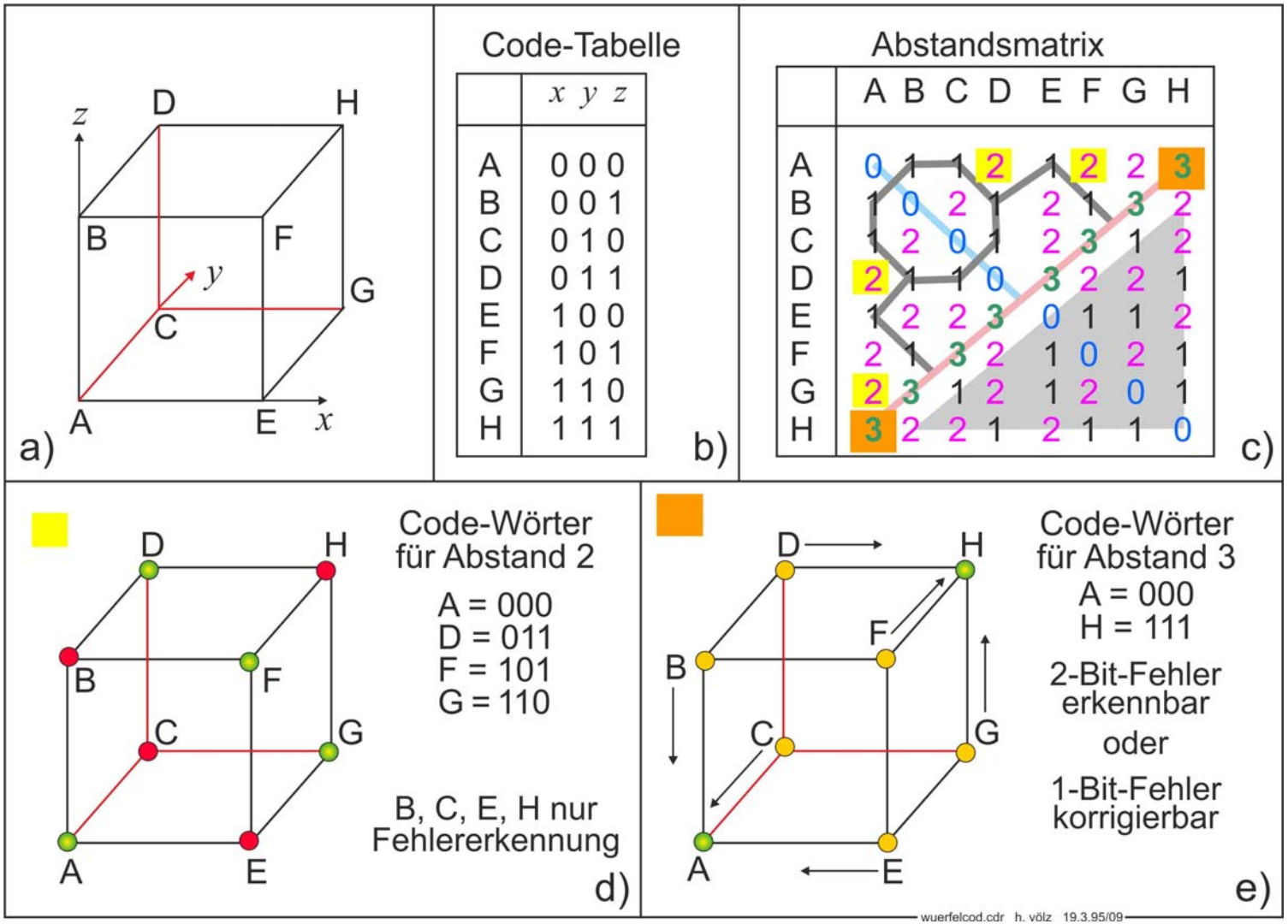
Die meisten *Fehler treten zufällig* auf. Daher besteht ein statistischer Zusammenhang.
Vergleich: Eine 6 tritt dabei mit der Wahrscheinlichkeit $^1/_6$ auf.
Die Wahrscheinlichkeit dafür, dass beim nächsten Wurf wieder eine 6 auftritt ist ebenfalls $^1/_6$.
Wahrscheinlichkeit dafür, dass bei zwei nacheinander erfolgenden Würfen zwei 6 auftreten $^1/_6 \cdot ^1/_6 = ^1/_{36}$.
Beträgt bei einer Übertragung die Fehlerrate für 1-Bit-Fehler 10^{-3},
so treten auch immer 10^{-6} 2-Bit-Fehler und 10^{-9} 3-Bit-Fehler usw. auf.
Werden mit Fehlerkorrektur alle 1-Bit-Fehler korrigiert, so bleiben die 2-Bit-Fehler, 3-Bit-Fehler usw.
Die Gesamtfehlerrate wird also nie Null, sie sinkt nur von z. B. 10^{-3} auf 10^{-6} usw.
Es kann also ***keine absolut fehlerfreie*** *Übertragung* geben. Dennoch sind sehr kleine Fehler erreichbar.
Jedoch ihre Messung ist sehr aufwändig bis unmöglich.
Wird z. B. bei der recht großen Datenrate von 10^9 Bit/s eine Fehlerrate von 10^{-14} gefordert,
Im statistischen Mittel würde der erste Fehler nach 10^5 Sekunden, also rund 80 Stunden zu erwarten sein.
Für eine leidliche Sicherheit des Messwertes dauerte die Messung mindestens eine Woche.
Neben streng zufälligen Fehlern treten zuweilen auch Bündel- bzw. Büschelfehler = ***Burst*** auf.
Bei ihnen liegen mehrere Fehler-Bit dicht gehäuft beieinander. (magnetischen Speicherung!)
auch bei Funken, Blitze und Ein- und Ausschalten von Geräten usw.
Im folgenden Bild ist die **Burstlänge** *b* gelb hervorgehoben.
Anschaulich scheinen sie für eine Korrektur nachteilig zu sein. Sie jedoch oft sogar vorteilhaft.
Ab dem ersten fehlerhaften Bit ist nur die relativ kleine Anzahl *b* der folgenden Bit zu beachten.
Das wirkt sich ähnlich wie eine Verkleinerung der Blockgröße aus.

Fehler-Arten



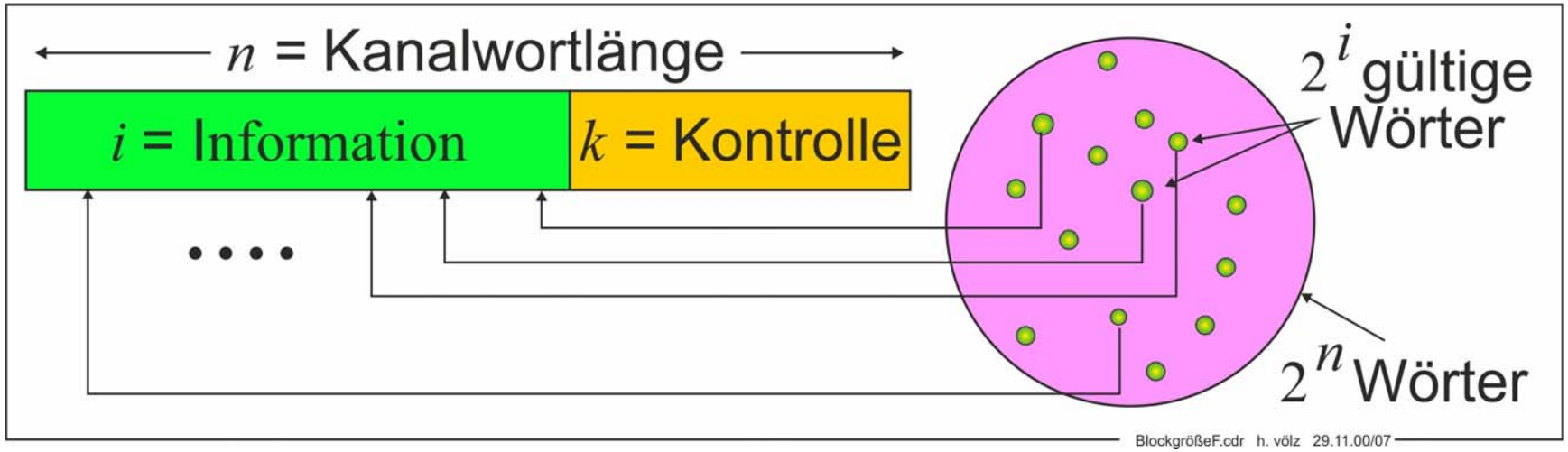
Anschauliche Einführung Korrektur

3 mögliche Bit in den drei *xyz*-Koorinaten lassen sich als **Würfel** darstellen (a).
Ermöglichen die 3-Bit-Wörter von A bis H der Codetabelle (b).
Bei Fehlern dürfen nicht alle verwendet werden.
Der Koordinatenursprung ist mit A = 000 gegeben.
Jeder Schritt davon weg entspricht einer 1.
Die Abstandsmatrix (c) zeigt, wie viele 0 ↔ 1 vertauscht wurden
Wegen der Symmetrie kann das graue Dreieck entfallen.
(d) zeigt Wörter mit 2-Bit-Abstand rot bzw. grün.
Werden z. B. die (grünen) Zeichen **A**, **D**, **F** und **G** als gültige, für die Übertragung genutzt, so führen 1-Bit Fehler zu den ungültigen Wörtern *B, C, E* oder *H*.



Drei Wort-Arten

Es sind 3 Wortarten zu unterscheiden.
Die *m* zu ***übertragenden*** Wörter mit der maximalen Bit-Länge ***i*** (von Information)
die beiden, aus *n* > *i* Bit (Kanalwortlänge *n*) bestehenden ***gültigen*** bzw. ***ungültigen*** Wörter.



Die **Zuordnung** zwischen den zu **übertragenden** und **gültigen** Wörtern kann recht unterschiedlich sein
Aus den 2^n möglichen langen Wörtern müssen lediglich *m* gültige Wörter so festgelegt werden,
und zwar so. dass der Bit-Abstand zwischen ihnen allen möglichst groß ist.
Bei der vorigen 3-Bit-Anordnung mit Würfel war das noch recht übersichtlich.
Beim Versuch mit 4 oder gar mehr Wörtern ist ähnliches Probieren recht kompliziert.
Daher ist die Auswahl der gültigen Wörter die erste schwierige Aufgabe für die Fehlerbehandlung.

Weitere Versuche

Die Betrachtungen des Würfels lassen sich leider nicht auf 4- und mehr Dimensionen fortsetzen.
mögliche Versuche zur Gewinnung von nützlichen Daten zeigt das nächste Bild.
Links ist die Abstandsmatrix für alle 4-Bit Wörter gezeigt.
Sie demonstriert wie kompliziert die Zusammenhänge sind.
Es gibt aber nur 2 gültige Wörter 0000 und 1111 oder andere gleichwertige Kombinationen.
Wird zu 3-Bit-Wörtern übergegangen, so existieren 4 Wörter mit 3*1.
Sie besitzen aber gegeneinander nur den Abstand 2. für Abstand 3 kann von ihnen ein Wort zu den obigen 2 ergänzt werden.
Wird auf den Abstand 2 übergegangen, so gilt die rechte Matrix.
Zu den obigen 2 Wörtern können dann also noch 6 hinzugenommen werden.

4-Bit-Wörter
Abstandsmatrix

	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000		1	1	2	1	2	2	3	1	2	2	2	2	3	3	4
1 0001	1		2	1	2	1	3	2	2	1	3	2	3	2	4	2
2 0010	1	2		1	2	3	2	2	2	3	1	2	1	4	2	3
3 0011	2	1	1		3	2	2	1	3	2	2	1	4	3	3	2
4 0100	1	2	2	3		1	1	2	2	2	3	4	1	2	2	3
5 0101	2	1	3	2	1		2	1	3	2	4	3	2	1	2	2
6 0110	2	3	2	2	1	2		1	3	4	2	2	3	2	1	2
7 0111	3	2	2	1	2	1	1		4	3	3	2	3	2	2	1
8 1000	1	2	2	3	2	3	3	4		1	1	2	1	2	2	3
9 1001	2	1	3	2	2	2	4	3	1		2	1	1	1	3	2
A 1010	2	3	1	2	3	4	2	3	1	2		1	2	3	1	2
B 1011	2	2	2	1	4	3	2	2	2	1	1		3	2	2	1
C 1100	2	3	1	4	1	2	3	3	1	1	2	3		1	1	2
D 1101	3	2	4	3	2	1	2	2	2	1	3	2	1		2	1
E 1110	3	4	2	3	2	2	1	2	2	3	1	2	1	2		1
F 1111	4	2	3	2	3	2	2	1	3	2	2	1	2	1	1	

Wörter mit Abstand 4
0000 und 1111

Bei Abstand 3 nur möglich
1 zusätzliches von
0111, 1011, 1101, 1110

Bei Abstand 2 gilt die Matrix
Es kommen also 6 hinzu

	0011	0110	0101	1100	1010	1001
0011		2	2	4	2	2
0110			2	2	2	4
0101				2	4	2
1100					2	2
1010						2
1001						

Fehlerfeld1.cdr h. völz 1.3.1

5-Bit Wörter usw.

Der Versuch kann für 5-Bit-Wörter ähnlich fortgesetzt werden, bringt aber nichts prinzipielles Neues.
Das Bild zeigt dazu nur Einiges.

5-Bit-Wörter mit 3*1 Bit

	00111	01110	11100	10010	10011	11001
00111		2	4	3	2	4
01110			2	3	4	4
11100				3	4	2
10010					2	3
10011						2
11001						

Abstände 6*2, 4*3, 5*4
Gegenüber 00000 alles 3-Bit-Abstand

Bei 4*1 nur Abstände 2

Fehlerfeld.cdr H. Völz 1.3.18

Damit entsteht die Frage nach einen effektiven Verfahren für ein bestmögliches gewinnen von gültigen Wörtern mit größtmöglichen Abstand.
Das ist eines der Hauptprobleme der Fehlerkorrektur und wird im folgenden versucht möglichst einfach darzustellen.

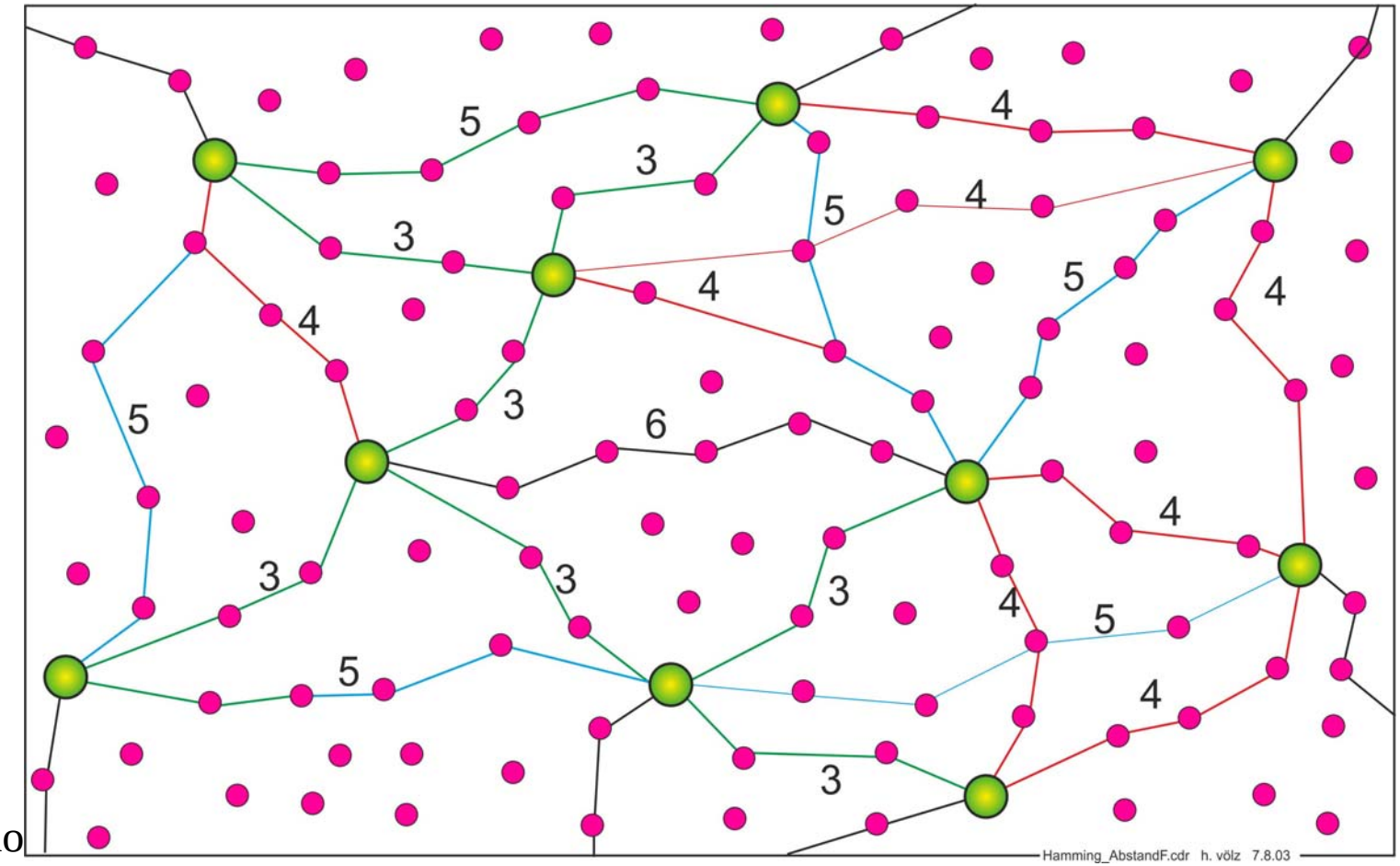
Hamming-Abstand

Für ***n*-Bit-Wörter** ist ein kaum vorstellbarer *n*-dimensioner Raum erforderlich.
Im Bild wird dazu ein eingeebneter Ausschnitt seiner Oberfläche benutzt.
Gültigen Zeichen sind wieder grün hervorgehoben.
0↔1-Vertauschungen führen über mehrere ungültige Zeichen (Abstände 3 bis 6).
Auf der gesamten Oberfläche gibt es einen geringste Abstand *h* (im Beispiel $h = 3$)

Er heißt ***Hamming-Abstand*** und bestimmt die Grenzen der Fehlererkennung und Fehlerkorrektur.

Das schwierige Problem der Fehlerkorrektur besteht in der optimalen Auswahl der 2^i gültigen Wörter aus den 2^n möglichen Wörtern

So, dass der **größtmögliche Hamming-Abstand** auftritt

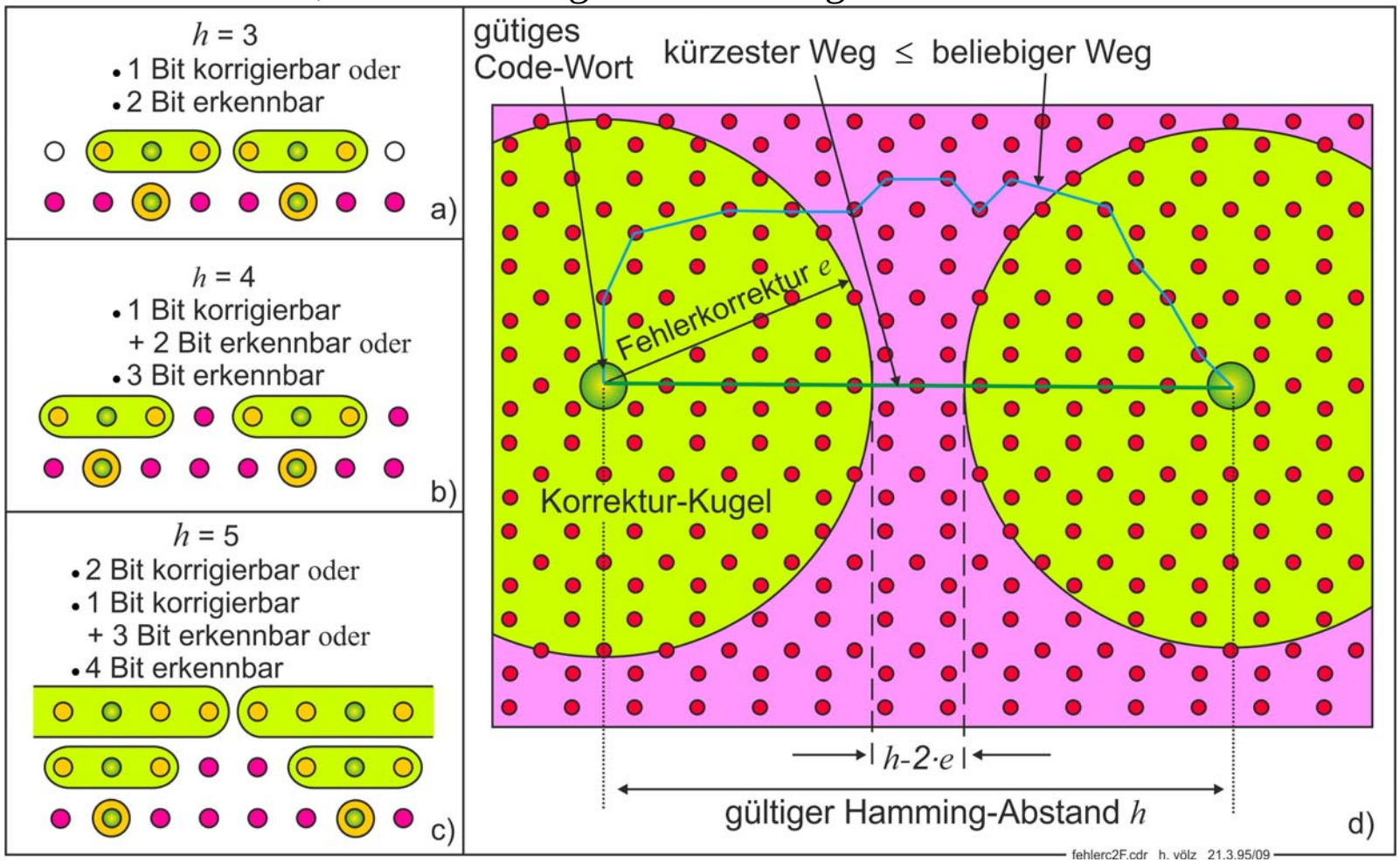


Hammimg-Abstand und Grenzen der Fehlerbehandlung

für einen bekannten Hamming-Abstand *h* lassen sich leicht Aussagen zur Leistungsfähigkeit angeben.
Beträgt *h* = 3 so sind ersichtlich 1-Bit-Fehler korrigierbar
oder wenn das nicht genutzt wird (ohne Korrektur) 2-Bit-Fehler erkennbar.
Falls ein 2-Bit- oder Mehr-Bit-Fehler auftrat, wird leider sogar falsch korrigiert.

Für *h* = 4 sind
2 „Betriebsweisen“
möglich:
1-Bit-Fehler werden
korrigiert und zusätzlich
2-Bit-Fehler erkannt
oder nur 3-Bit-Fehler
erkannt

Bei *h*=5 sind drei
Betriebsweisen möglich
usw.



Abschätzungen

Für den Zusammenhang zwischen der maximal erkennbaren Fehler f_{max} und dem *Hamming*-Abstand h gilt

$$f_{\max} = h - 1.$$

Für die Anzahl der maximal korrigierbaren dagegen

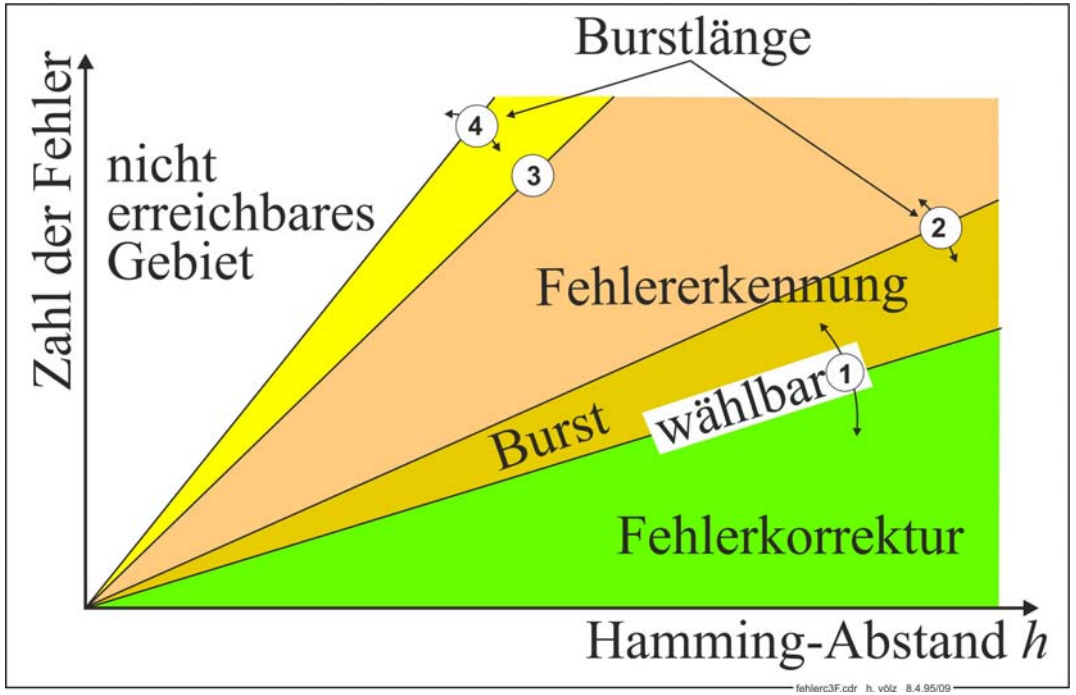
$$e_{\max} \le \mathrm{INT}\left(\frac{h-1}{2}\right).$$

Beide Größen nicht unabhängig: Werden nur $e < e_{\max}$ Fehler korrigiert, so sind nur noch Fehler erkennbar

$$f_{\max,e}(h, e) = h - e - 1.$$

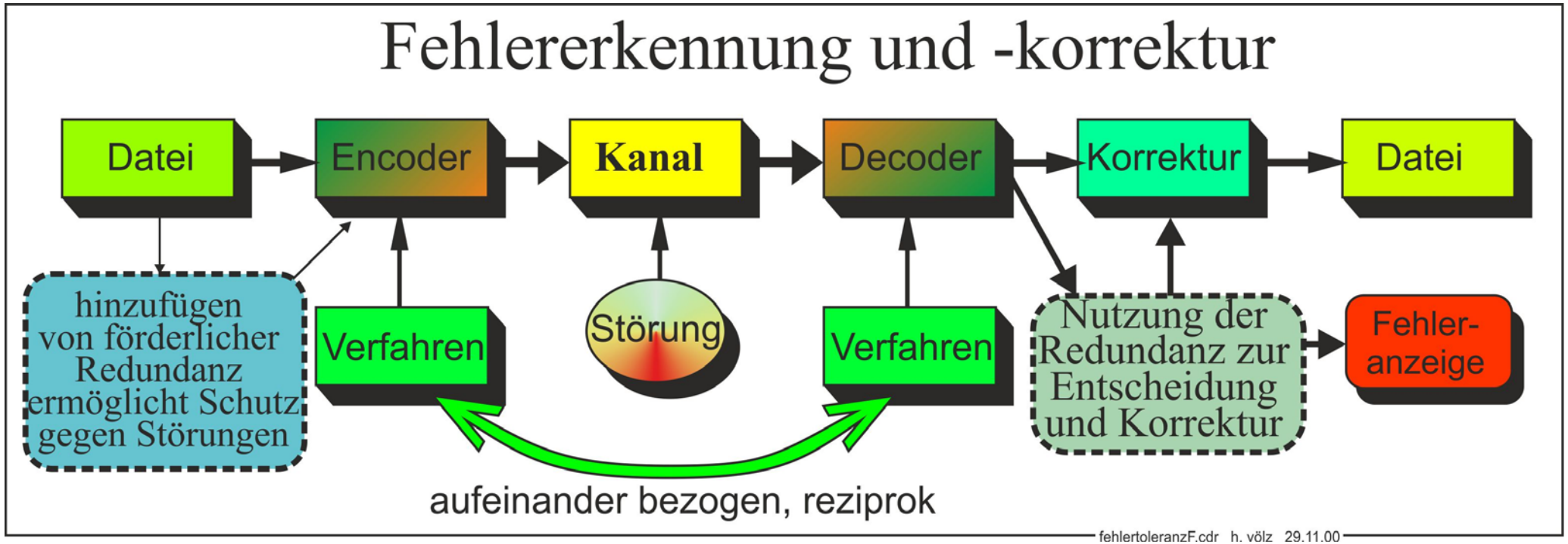
Daher sind Fehlererkennung und -korrektur oft gegeneinander abzuwägen.

Mit Berücksichtigung der zusätzlichen Möglichkeiten bei Bursts ergibt sich der folgende Überblick



Das Kanalschema

Den zu übertragenden Zeichen werden für den Fehlerschutz zusätzliche Bits als Redundanz hinzugefügt
Das steht ganz im Gegensatz zu den Fakten bei der Entropie und Komprimierung, realisiert vom Encoder.
Oft geschieht das aber nicht durch einfaches von Anhängen von Bits.
Meist erfolgt es aber auf komplexe Weise mit längeren gültigen Wörtern
Nach der Übertragung prüft der Decoder, ob ein ungültiges Wort vorliegt.
Dann wird entschieden, ob Fehlererkennung und/oder -korrektur erfolgen soll.
Bei der Fehlererkennung wird das ungültige Wort einfach verworfen.
Mehrfache Übertragung bringt keine Sicherheit (evtl. 2- oder mehrmals gleivh falsch).



Einige Grenzen

Die Zusammenhänge von Fehlererkennung und -korrektur lassen sich vertiefen.

Bei Blocklänge n aus i Informations- und k Kontroll-Bits ($n = i + k$) ist der „förderliche“ Redundanzfaktor

$$r = \frac{n-i}{n} = \frac{k}{n}.$$

Für den erreichbaren ***Hamming***-Abstand gilt

$$h \le \frac{2^n}{2^i} - 1 = 2^k - 1.$$

Die minimale Anzahl der erforderlichen Kontroll-Bit für die zu erkennenden Fehler f beträgt

$$k \ge \text{INT}(1 + \text{ld}\,(f + 1)).$$

Hier geht die Blocklänge n nicht ein – ganz im Gegensatz zur Fehlerkorrektur.

Für e korrigierbare Fehler ist die Berechnung deutlich komplizierter. Denn es müssen die einzelnen Fehlertypen mit den entsprechenden Distanzen aufaddiert werden:

$$k \ge \text{ld}\left(\sum_{x=1}^{e}\binom{n}{x}\right).$$

Der Wert ist nur iterativ zu berechen z. B. mit dem nebenstehenden Programm.

```
k1=1
FOR x = e-1 TO 0 STEP -1
  k1 = k1*(n-1)/x - x +1
NEXT x
k = log(k1)/log(2)
```

Einfache Verfahren

Bei den meisten Fehlerverfahren ist die Erzeugung der gültigen Wörter recht komplex.
Es gibt jedoch drei relativ „einfache“ Methoden

- Beim **Gleichen Gewicht**: wird eine Anzahl k < m der im Wort enthaltenen 1 festgelegt. z. B. für 4×1 in 110011, 101011, 110110, 10101010, 1111 usw. Das ermöglicht nur eine Fehlererkennung.
- Bei der **Symmetrie** sind die gültigen Wörter vor- und rückwärts gelesen gleich, z. B. 110011, 101101, 011110 usw. Der Code ist hoch redundant und ermöglicht teilweise auch Fehlerkorrektur. Für ihn gibt es keine geschlossene Theorie. Er wird u. a. beim klassischen Barcode benutzt.
- Bei der **Parity** (Parität lateinisch paritas Gleichheit) wird die Anzahl der im Wort enthaltenen 1 gezählt. Dabei gibt es zwei Varianten: Ihre Summe soll für gültige Wörter *gerade* oder *ungerade* sein. Damit das geforderte Ergebnis vorliegt, wird zusätzlich eine 1 oder 0 angehängt. Es werden hierbei nur 1-Bit-Fehler erkannt. 3-, 5- und 7-Bit-Fehler erscheinen als 1-Bit-Fehler.

Es gibt auch spezielle, für nicht-binäre Codierungen abgewandelte Paritäts-Codes,
z. B. für ISBN-Nummer von Büchern, Schutz für Geldscheine und IBAN-Werte für Konten.

Erweiterte Parity-Verfahren

Zuweilen werden auch ***mehrere Parity-Bit*** benutzt.
Dann sind mehrere Fehler zu erkennen und einzelne sogar zu korrigieren

Etwas komplexer die **Block-Parity**, 1973 von IBM für den 9-Spur-Magnetspeicher als GCR (Gruppencodierung) eingeführt, auch Längsquer-, Matrix-, oder Kreuz-Parity genannt
Mit 8 Spuren wurden 8×8-Bit-Blöcke gebildet und die Parity in beide Richtungen (längs und quer) gebildet.
Die neunte Spur erhielt die Werte der Quer-Parity.
Je Block kann ein 1-Bit-Fehler korrigiert und ein 2-Bit-Fehler erkannt werden.

Vereinfacht auf 6 Spuren und 4-Bit-Wörter zeigt es bei ungerader Parity die nebenstehende Tabelle.
Der rot unterlegte, durchgestrichene Parity-Wert 0 in der 7. Spur wird nicht benutzt.

Prinzipiell sind Block-Codes mit mehr Dimensionen möglich.
Ihre Effektivität ist jedoch geringer

Gesendet		1-Bit-Fehler			2-Bit-Fehler			Mehr-Bit-Fehler		
Wort	P	Wort	P		Wort	P		Wort	P	
1001	1	1001	1		1000	1	←	1000	1	←
0011	1	0011	1		0011	1		0011	1	
0100	0	0000	0	←	0100	0		0000	0	←
0101	1	0101	1		0101	0	←	1001	1	*
0010	0	0010	0		0010	0		0010	0	
0111	0	0111	0		0111	0		0111	0	
0001	~~0~~	0001	~~0~~		0001	~~0~~		0001	~~0~~	
		↑			↑	↑		↑*	↑	
		1 Fehler korrigierber			2 Bit nur erkennbar			* nicht mal erkennbar		

Kurzer Überblick

Es gibt mehrere komplexe und mathematisch recht aufwändige Verfahren der Fehlerbehandlung.

Namen oft nach den **Erfindern** benannt.

Das immer noch grundlegende Werk mit sehr **guter Einführung** ist [Pet67].
Darin fehlen allerdings **neue Codes**. Für sie sei [Fri96] empfohlen.
Verkürzte Beschreibungen enthalten u. a. [Völ96], S. 95ff., [Völ01], S. 427ff. und [Völ07], S. 109ff.

Häufig erfolgen die Untersuchungen zur Fehlerkorrektur mit den **Galois-Feldern** in der Matrizen-Methode.
Denn sie ermöglicht auch, dass nicht-binäre, also höherwertige Variablen nutzbar sind.

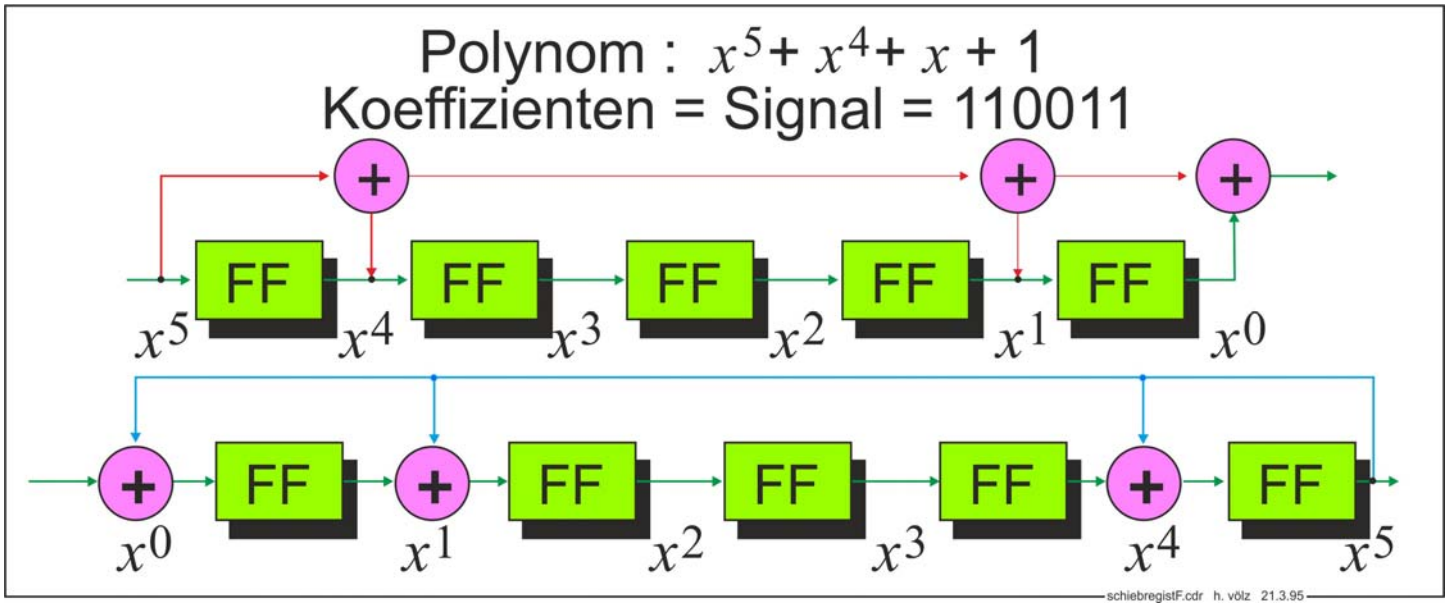
Hier wird nur die etwas übersichtlichere ***Polynom-Methode*** für binäre Wörter beschrieben.

Polynom-Methode

Ein typisches Polynom und seine wichtige, aber die verkürzte Schreibweise lautet

$$y = 1\cdot x^5 + 1\cdot x^4 + 0\cdot x^3 + 0\cdot x^2 + 1\cdot x^1 + 1\cdot x^0 \quad = \quad x^5 + x^4 + x + 1.$$

Für das „eigentliche“, rein mathematische Polynom können die ***Potenzen mit Faktor 0 entfallen***.
Ferner kann angenommen werden, dass die *Variable x* nur eine Hilfsgröße, ein dummy ist.
Dann entspricht dem Polynom auch dem **binären Wort** 110011.
eine gleichwertige Darstellungen erfolgt mittels ***XOR-rückgekoppelter Schieberegister-Ketten***.
Weil sie *vorwärts oder rückwärts* gerichtete Kopplungen besitzen können, existieren *zwei Varianten*.
die 4 Varianten vom Bild sind für die Fehlerbehandlung äquivalent.
Die formale „Übereinstimmung“ ermöglicht es, Polynome, Signale und Hardware ineinander umzurechnen.
So folgt aus der Mathematik sofort die dazu gehörende (elektronische) Schaltung.
Für die technische Entwicklung bringt das große Vereinfachungen (Obere Kette ist einfacher!)



Anwendung binäre Logik

Die Polynommethode benutzt die binäre Logik (Arithmetik) gemäß den folgenden Tabellen
Bei 2 Eingangsvariable logische Operatoren: XOR als Addition/Subtraktion, AND als Multiplikation, EQU.
Auf Code-Wörter werden sie bitweise ohne Übertrag angewendet.
Beispiele zeigen die Teiloperationen bei der Addition = Subtraktion, Multiplikation und Division
Die Polynom-Methode kann unmittelbar und gleichartig auf **Signale** als **Daten-Polynome** $D(x)$ und auf die **Hardware**, nämlich Schieberegister als Generator-Polynome $G(x)$ angewendet werden.
Signale und Schieberegister folgen dabei dem gleichen Takt.

Logik				
x1	0	0	1	1
x2	0	1	0	1
XOR = Addition = Subtraktion	0	1	1	0
AND = Multiplikation	0	0	0	1
EQU = Vergleich	1	0	0	1

Addition = Subtraktion

```
  10101
+11110
  01011
```

Multiplikation

```
10101 · 101
  10101
 00000
10101
1000001
```

Division

```
10000000000000 : 110101 = 111011001
110101                    Teiler        Ergebnis
 101010
 110101
  111110
  110101
    101100      zwei Stellen (00) holen
    110101
     110010
     110101
        111000      drei Stellen (000) holen
        110101
           1101     = Rest der Division
```

Rechnungen erfolgen ohne Übertrag!

Irreduzible Polynome

Die ganzzahlige Division hat die Besonderheit: Ein nicht weiter *ganzzahlig zu teilender Rest* kann auftreten
Im Tabellenbeispiel gilt x^{13} dividiert durch $x^5+x^4+x^2+1$ ergibt
$x^8+x^7+x^6+x^4+x^3+1$ mit dem Rest x^3+x^2+1.
Die Umkehrung dazu lautet

$$(x^8+x^7+x^6+x^4+x^3+1) \cdot (x^5+x^4+x^2+1) + (x^3+x^2+1) = x^{13}.$$

Ähnlich den *Primzahlen* gibt es auch Polynome, die ohne Rest nur durch x^0 (= 1) und sich selbst teilbar sind. Sie besitzen die Besonderheit der Symmetrie. Ist z. B. 10011 irreduzibel, so ist es auch 11001.

In Listen wird daher nur eine der beiden Varianten aufgeführt. Die ersten irreduziblen Polynome lauten:

11; 111; 1011; 10011; 11111; 100101; 101111; 110111.

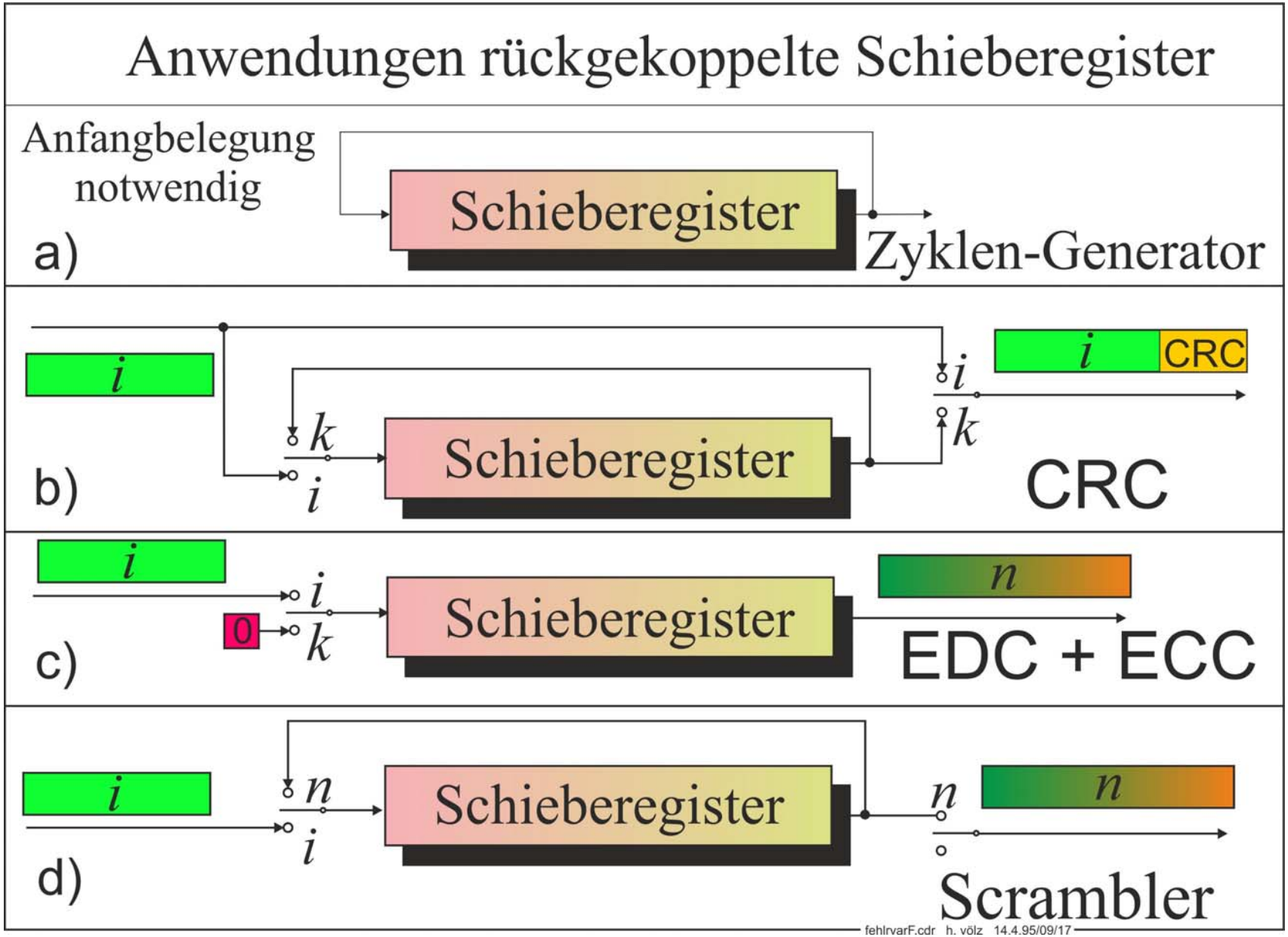
Anwendungen rückgekoppelter Schieberegister

Rückgekoppelten Schieberegister-Ketten bestehen aus in Reihe geschalten Flipflop mit gemeinsamen Takt
Er verschiebt die Inhalte jeweils einen Flipflop nach rechts, wobei vorn ein neues Bit übernommen wird.
Es existieren die vier Anwendungen vom Bild (nächste Seite).
Beim **Zyklen-Generator** a) wird kein Eingangswort benutzt.
Der Ausgangswort wieder zum Eingang → Generator
Beim **irreduziblem** Polynom werden alle möglichen Wörter (Flipflop-Anzahl = Polynomgrad) durchlaufen.
Bei **reduziblen** Polynomen treten je nach der Anfangbelegung mehrere, z. T. unterschiedliche Folgen auf.

Bei einem richtig ausgewähltem (Generator-) Polynom $G(x)$ und passender Anfangsbelegung können für die Fehlerbehandlung nur Wörter mit einem ***optimalen Hamming-Abstand*** erzeugt werden.
Spezieller ist die so auch mögliche Erzeugung von ***Zufallszahlen.***

2 mal Länge 8		**3 mal Länge 4**			**Länge 2**	**Länge 3**
00001	00111	00011	00101	01001	01111	10001
00010	01110	00110	01010	10010	11110	
00100	11100	01100	10100	10111		
01000	01011	11000	11011	11101		
10000	10110					
10011	11111					
10101	01101					
11001	11010					

Arten der Schieberegister



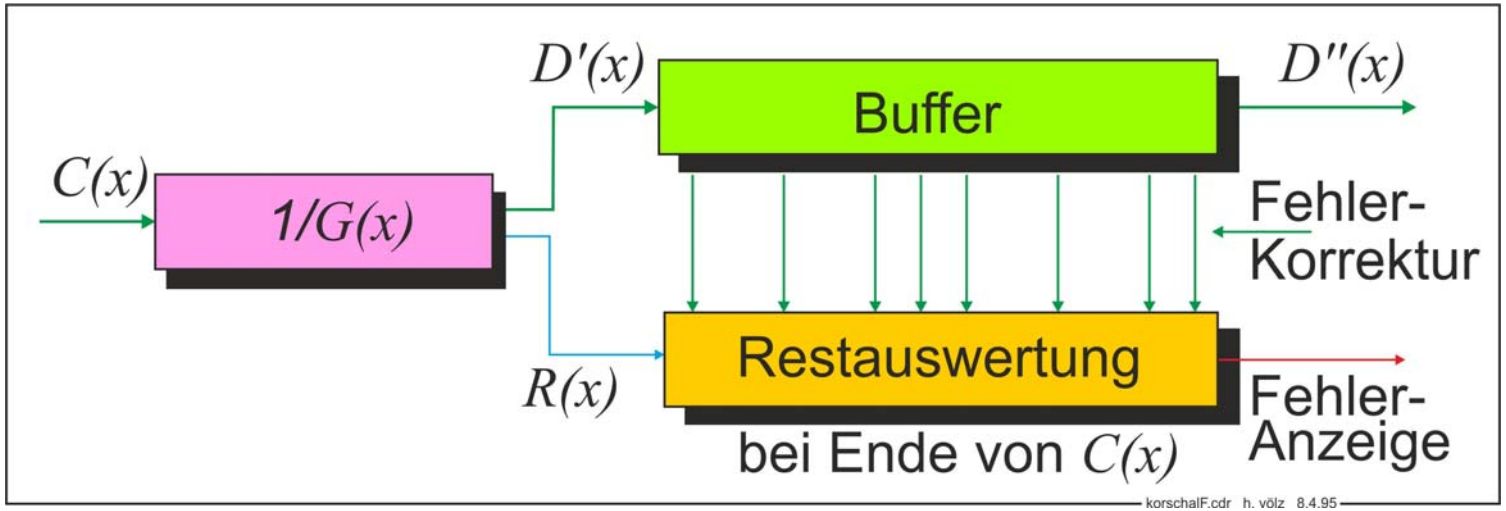
2. Anwendungen rückgekoppelter Schieberegister

Die zweite Anwendung ist die hocheffektive **Fehlererkennung** mit dem **CRC** (cyclic redundancy code).
Dabei wird auf der Sende- und Empfangsseite ein irreduzibles Polynom benutzt.
Vor dem Beginn jedes neuen Wortes werden alle Flipflop auf 0 gesetzt.
Das zu sendende (gültige) oder empfangene Wort wird mit dem Takt zum Eingang des Schieberegisters und zum Ausgang der Schaltung geleitet.
Beim Wortende steht im Schieberegister eine über das Polynom berechnete Bit-Folge, das CRC-Zeichen.
Beim Senden wird es ans (gültige) Wort gehängt, bei der Wiedergabe zur Fehlererkennung verglichen.
Besteht kein Unterschied, so ist entsprechend der Länge *n* des Polynoms (Länge des CRC-Zeichens) kein Fehler aufgetreten.
Der Zyklus eines irreduziblen Polynoms ist 2^n-1 lang und ergibt so eine Fehlersicherheit $1/(2^n\text{-}1) \approx 10^{-0,3 \cdot n}$
Praktische Bedeutung haben die der Tabelle, u. a vom CCITT und ISO genormten CRC-Polynome.

Länge	Fehler	Polynom
8 Bit	$4 \cdot 10^{-3}$	$x^8 + 1$
12 Bit	$2 \cdot 10^{-4}$	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
16 Bit	$2 \cdot 10^{-5}$	$x^{16} + x^{15} + x^2 + 1$; oder $x^{16} + x^{12} + x^2 + 1$; oder $x^{16} + x^{12} + x^5 + 1$
32 Bit	$2 \cdot 10^{-10}$	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

3. Schieberegister für Fehlerkorrektur und -erkennung

Auch bei der kombinierten Fehlererkennung und -korrektur (EDC + ECC) werden Flipflop auf gesetzt
Dann wird das Eingangssignal $D(x)$ Bit-weise ins Schieberegister geleitet.
Mit den XOR-Verknüpfungen des Generator-Polynoms $G(x)$ wird die Belegung der Flipflops geändert.
So entsteht das gültige Ausgangswort $C(x) = G(x) \cdot D(x)$.
Anschließend wird der Inhalt des Schieberegisters taktweise angefügt (Länge = Anzahl Kontroll-Bit.
Zur Decodierung muss die Division $C(x)/G(x)$ realisiert werden.
Bei Übertragungsfehlern entstehen verändertes Datenwort $D'(x)$ und Fehlervektor = Rest $R(x)$ = Syndrom
Der Rest ermöglicht eine Fehlererkennung.
Dazu ist ein Buffer mit der Länge des Codewortes notwendig. Die falschen Bit werden invertiert
Das Generatorpolynom $G(x)$ (Schaltung) bestimmt die mögliche Fehlererkennung und -korrektur
Typisch sind Hamming-, Fire-, Bose-Chaudhuri-Hocquenghem- (BCH-) und Reed-Solomon-Code
Es gibt aber bisher kein universelles Verfahren.



Scrambler = Verwürfler

Er arbeitet mit Pseudozufall und wird ***nicht zur Fehlerbehandlung*** benutzt.
Bei der **Speicherung** erzeugt Codewörter mit bestimmter Lauflänge (RLL)
Sie enthalten dann eine maximale Anzahl aufeinander folgende 0 bzw. 1
Es geschieht also eine Umcodierung vom NRZ- zum randomized RNRZ-Code.

Eine weitere Anwendung ist der **Datenschutz**, z. B. beim Pay-TV, um so die Bezahlung sicherzustellen.
Erst nach einem Descrambling ist ein brauchbares Bild wiederzugeben.
Wichtig sind hierbei reduzible Polynome, z. B. $1+x^{14}+x^{17}$; $1+x^{6}+x^{7}$ und $1+x^{18}+x^{23}$.

Das **GPS** der USA ist so absichtlich für die private Nutzung ungenauer gemacht.
Nur für das Militär ist die Rückwandlung in die exakten Signale per Descrambler möglich.

Matrix-Methode

Sie ist die zweitwichtigste Methode der Fehlerkorrektur.
Sie gestattet auch andere Zahlenbasen als 2, bei Galois-Feldern z. B. 2^n und Primzahlen [Fri96], [Pet67].
Die 2^i gültigen Codewörter werden aus dem *n*-dimensionalen Raumes aller Wörter erzeugt.
Mit den Basisvektoren g_l der Generatormatrix [*G*] gilt für die einzelnen Codewörter

$$v = c_1 \cdot g_1 + c_2 \cdot g_2 + c_3 \cdot g_3 + \cdots + c_i g_i.$$

$$[G] = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_i \end{bmatrix}$$

Die Skalare c_l entsprechen den zu übertragenden Wörtern;
Mit [*G*] können alle gültigen Wörter mittels einer Code-Eigenschaftsmatrix [*C*] und dem Einheitsvektor [*E*] erzeugt werden. Damit entsteht die Matrix-Schreibweise $[v] = [C] \cdot [G]$:

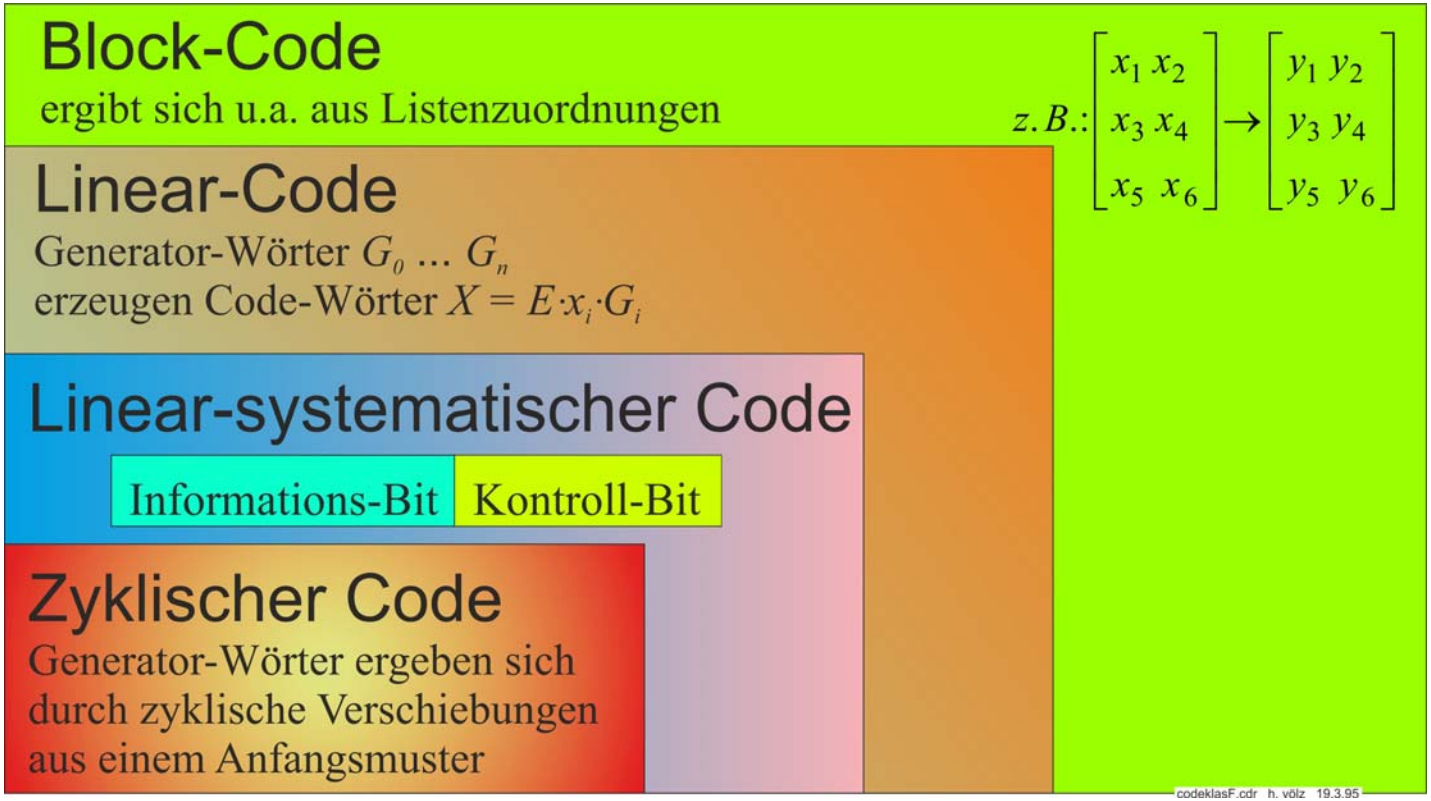
c_j	1 1 0 1 0 0 1 1 0 1 0 0 1 1 = Generatorwörter [*G*] 1 1 1 0 1 0 0
0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 1 1 = Codewörter [v_i] 1 1 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1 0

Nicht immer existiert [*G*] in der systematischen Gestalt. Dann ist [*E*] nicht direkt zu erkennen.
Die praktische Anwendung der Matrix-Methode verlangt:

1. Die *i* Zeilen der Generatormatrix müssen gespeichert vorliegen und
2. muss eine Schaltung zur Matrizen-Multiplikation existieren.

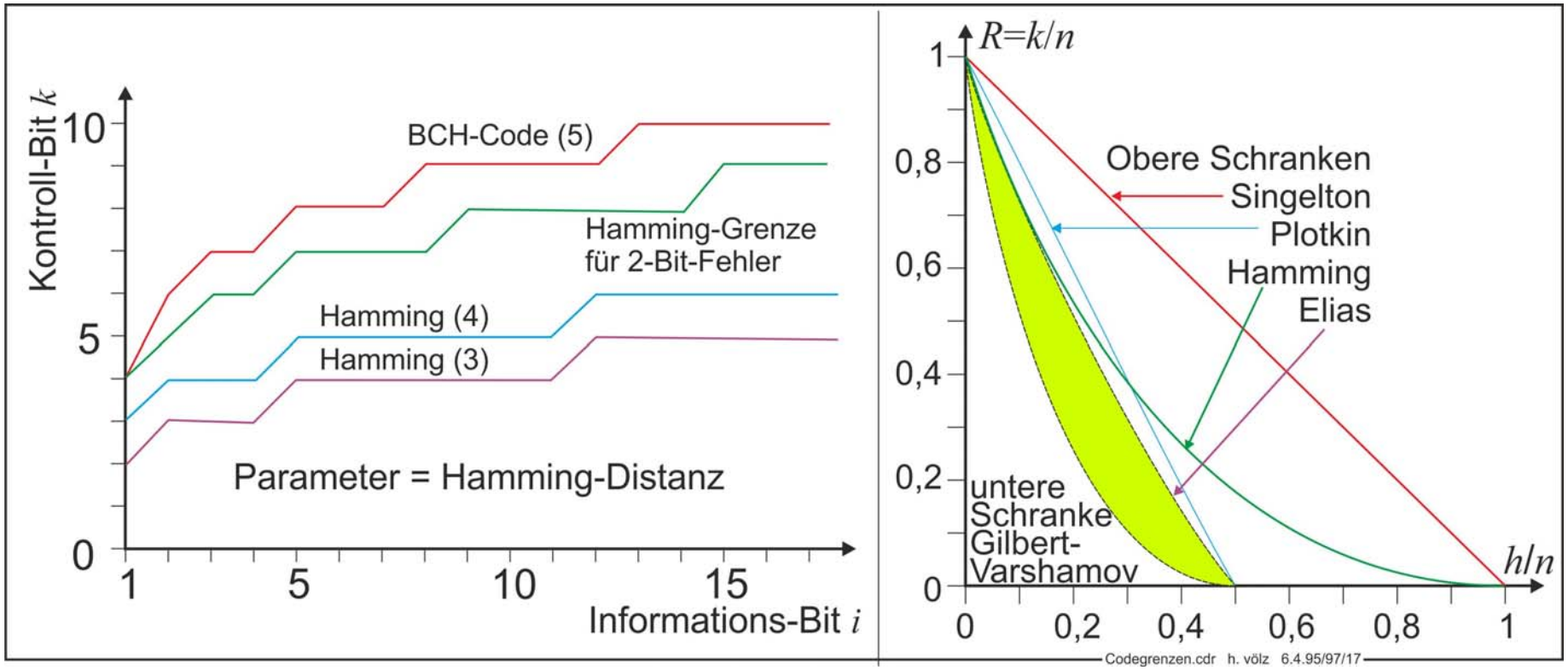
Systematik

Eine Einteilung der verschiedenen fehlersichernden Codes ist schwierig (vereinfacht s. Bild).
Die Block-Codes lassen sich oft durch eine Tabelle beschreiben.
Z. T. sind sie aber so groß, dass sie technisch nicht mehr effektiv nutzbar sind.
Dann sind die aufgeführten und weitere algorithmische Methoden notwendig.
Eine gewisse Einengung bezüglich der möglichen Codewörter stellt der linear-systematische Code dar.
Bei ihm sind die zu übertragenden Wörter noch in den gültigen Wörtern direkt erkennbar.
Die Kontroll-Bits sind dann einfach angehängt.
Spezieller ist der zyklische Code: Die gültigen Wörter entstehen durch zyklische Verschiebung



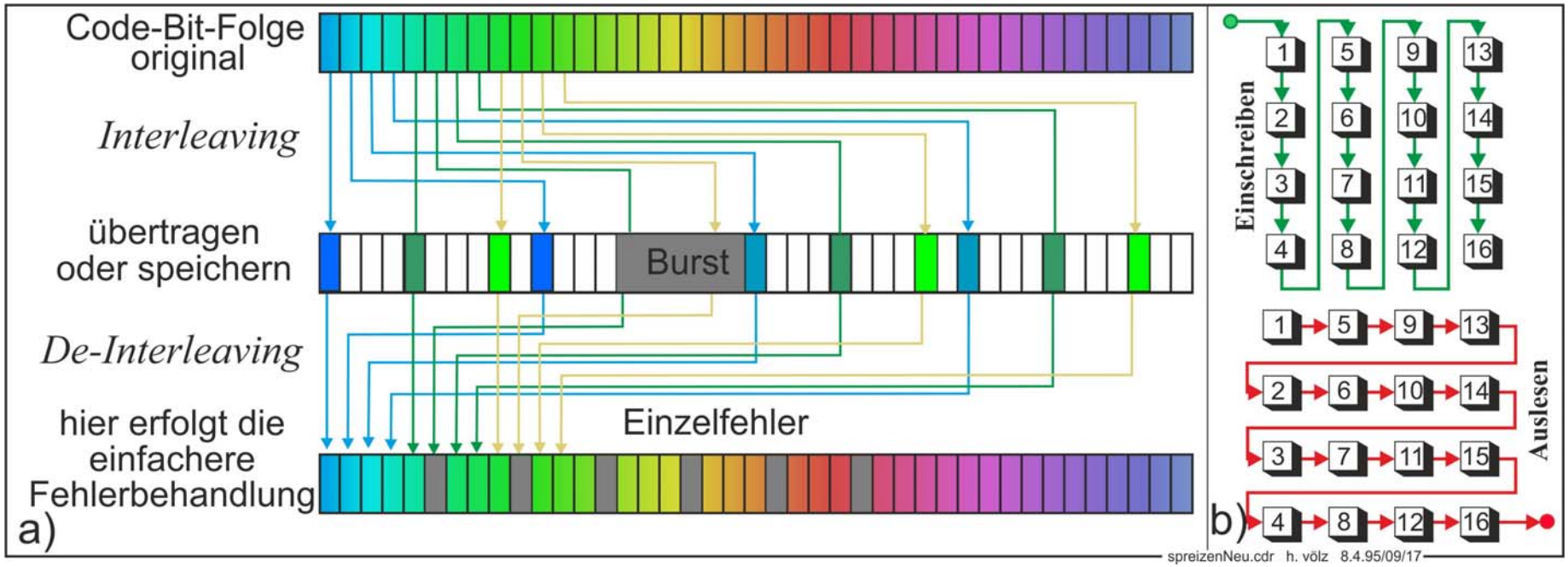
Grenzen

Das Verhältnis der *i*-Bit- zu übertragenden unn *n*-Bit-langen gültigen Wörter ist die Coderate $R = i/n$.
Sie soll möglichst groß sein. Einen Überblick zu erreichten und theoretischen Werten zeigt das Bild
R nimmt mit der Länge der zu übertragenden Wörter zu.
Leider nimmt damit aber auch die Fehler-Wahrscheinlichkeit zu.
Für jeden gibt es typisches Optimum, das auch vom angewendeten Verfahren abhängt.
Für die gerade noch zulässige Fehlertoleranz gibt es mehrere Abschätzungen,
u. a. die Schranken von Hamming, Elias, Singelton, Plotkin und Gilbert-Varshamov
Die Abzisse benutzt die Darstellung das Verhältnis aus Hamming-Distanz und Codewortlänge *h*/*n*.



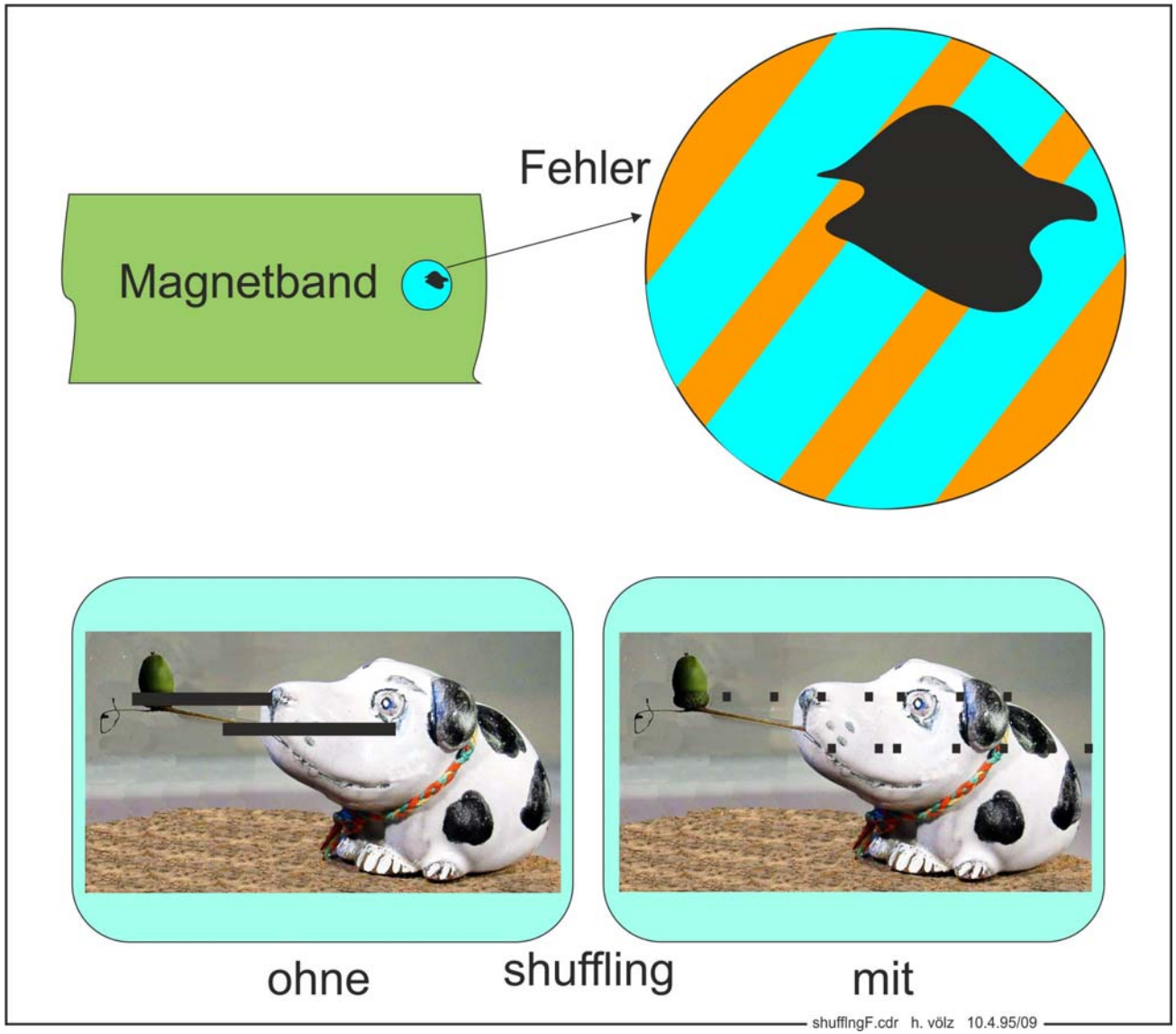
Spreizung

Sie ist vorteilhaft bei gehäuften Fehlern (Bulk) in einem Block.
Sie lassen sich oft relativ einfach zu Einzelfehlern in verschiedenen Blocks umwandeln.
Dazu dient spezielle Umsortierung. Als Einzelfehler sind sie dann leicht zu korrigieren.
Typisch ist die systematische Spreizung, auch Interleaving, Verschachtelung, Datenspreizung, Verwürfelung, Codespreizung, Überlappung oder Bit-Umordnung genannt.
Einzelne Bit werden über die Blockgrenzen hinaus um 0, 10, 20 und 30 Bit-Positionen verschoben.
Im Bild a ist das nur für die ersten 12 Bits dargestellt.
Und zwar für Übergänge: $1 \rightarrow 1$, $2 \rightarrow 11$, $3 \rightarrow 21$, $4 \rightarrow 31$, $5 \rightarrow 2$, $6 \rightarrow 12$, $7 \rightarrow 22$, $8 \rightarrow 13$, $9 \rightarrow 3$ usw.
Nach Korrektur erfolgt die Rücksortierung
Oft mittels Matrix-Zusatzspeicher spaltenweise eingeschrieben und nach Korrektur zeilenweise gelesen.



Erfolg beim Fernsehen

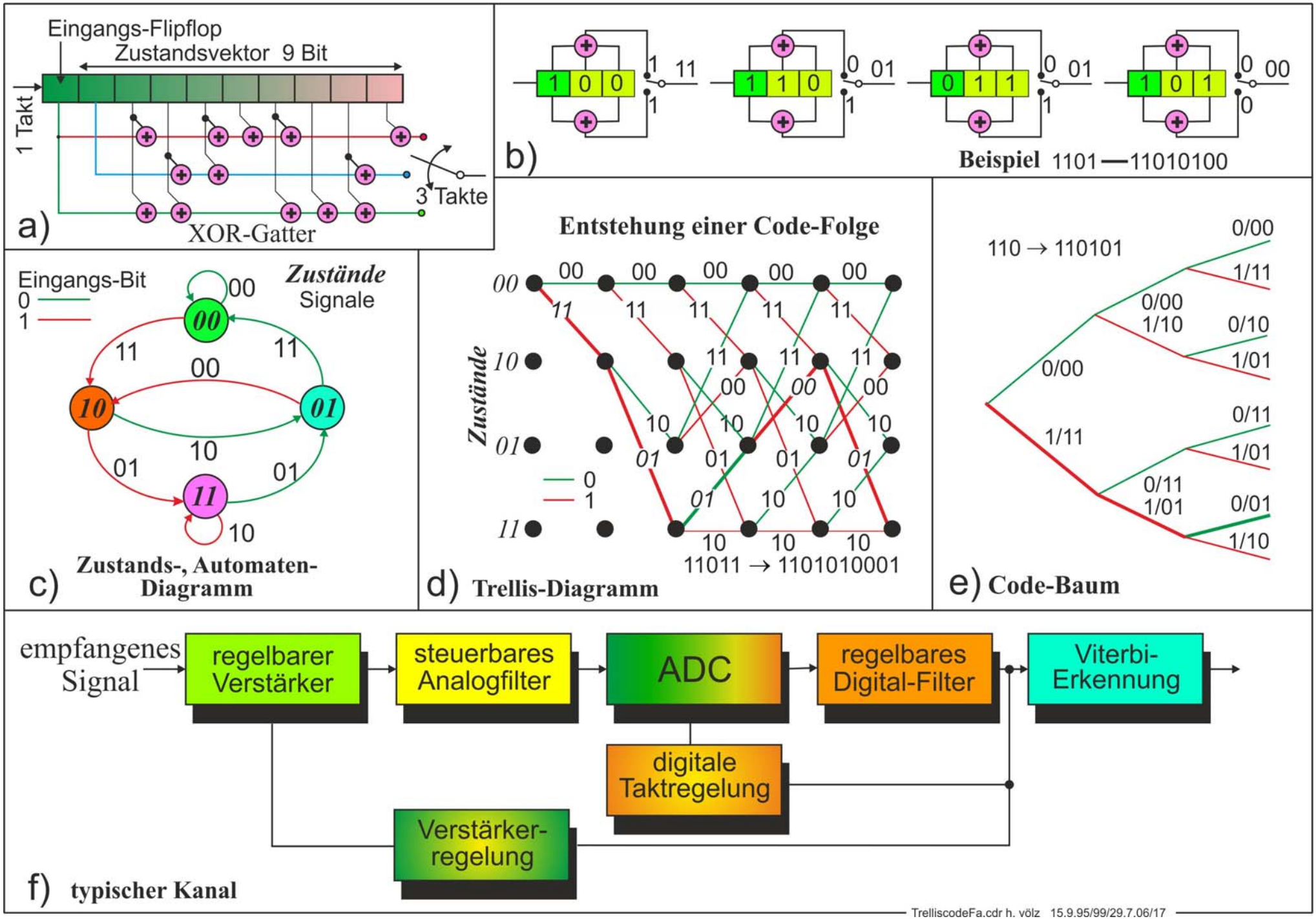
Hier wird meist von shuffling gesprochen. Den möglichen Gewinn zeigt das Bild



Faltungs-Codes 1

Die bisherigen Fehlerbehandlungen beruhen auf der Blockbildung der Daten.
Seit den 80er Jahre werden dagegen immer häufiger (auch für Speicherung) Faltungs-Codes
Sie codieren fortlaufend Bit für Bit. Auch wichtig für beliebig lange Datenströme z. B. Streaming.
Dazu werden aus jedem Signal-Bit 2, 3 oder mehr Code-Bits erzeugt.
Der Datenstrom ist daher ein ganzzahliges Vielfaches der gültigen Wörter bzw. des Signalstromes.
Bei der Wiedergabe kann über diese Redundanz eine Fehlerkorrektur erfolgen.
In der schaltumgstruktur existiert ein Schieberegister der Länge *n* (im Bldbeispiel Länge 9).
Die Daten werden taktgetreu hinein geschoben. Im Gegensatz zu Blockcodes existiert keine Rückkopplung.
Es besteht ein „Gedächtnis“ für die jeweils letzten *n* Bits.
Von ihnen werden durch logische Verknüpfungen *m* (meist 2 im Beisspiel 3) neue Bits gebildet.
Diese Werte werden mit der *m*-fachen Datenrate als Code-Wörter ausgegeben.
Einige Sonderfälle des Ablaufs zeigt das Bild
Aus dem Signal-Wort 1101 entsteht dabei das Code-Wort 11 01 01 00.
Die formale Beschreibung dieser Decoder ist in drei Varianten möglich.

Beim Zustands-; Automaten-Diagramm wird die Darstellung des Mealy-Automaten benutzt (c).
Je nach den Bits in den Registern existieren 2^{n-1},
Über die Automaten-Zustände (Kreise) werden an den Pfeilen stehenden Bit-Kombinationen ausgegeben.
Es ist auch eine Umsetzung in eine Automatentabelle möglich.
Beides sind besonders kurze Fassungen der Codierung, sind aber für die Fehlerkorrektur unwesentlich



TrelliscodeFa.cdr h. völz 15.9.95/99/29.7.06/17

Faltungs-Codes 2

Die zweite Darstellungsvariante ist ein Code-Baum, der mit der Startposition beginnt. Alle Register auf 0
Dann können 0 oder 1 als Signal-Bit auftreten und im Beispiel werden 00 oder 11 ausgegeben.
Das ergibt die beiden nächsten Knoten des Baumes. Mit dem folgenden Bit je zwei Verzweigungen möglich,
strebt zu vielen Verzweigungen, ist zur Darstellung nicht schnell terminierender Signalfolgen ungeeignet.
Für eine bestimmte Signalfolge wird aber nur eine Verzweigung durchlaufen.
Für die Signalfolge 110 ist sie dick gezeichnet.

Weitgehend durchgesetzt hat sich das Trellis-Diagramm (d) (englisch trellis Raster, Gitter).
Hierfür sind die Automatenzustände als Zeilen vorhanden.
ie Signaltakte sind in den Spalten als Punkte von links nach rechts aufgereiht.
Zum nächsten Automaten-Zustand gibt es jeweils zwei Möglichkeiten der Fortsetzung.
Im universellen Schema gibt es auch hier für eine Signalfolge nur einen Weg im Trellis-Diagramm.
Im Bild ist die Signalfolge 11011 wiederum fett hervorgehoben.
Im Gegensatz zur Baumdarstellung bleibt dabei (Bezug Automatenzustände) die Diagrammsbreite konstant.
Daher lassen sich auch gut recht lange Signalfolgen als Weg darstellen.
Das Trellis-Diagramm hat für die Fehler-Korrektur einen großen Vorteil.
Bei Übertragungsfehlern wird der Weg verändert. Meist treten Wege auf, die „unzulässig“ sind.
Mittels Rechnung kann dann der wahrscheinlichste, „ursprüngliche“ und zulässige Weg ermittelt werden.
Bei langen Signalfolgen können sich jedoch Fehler fortpflanzen, die eine Korrektur unmöglich machen.
Deshalb ist die Terminierung der Signalfolge nach einer bestimmten Länge sinnvoll oder gar notwendig.
Truncation (engl. Abstumpfung, Stellenunterdrückung) oder Tail-Biting (englisch tail = Ende, Schwanz)
Vorteil der Faltungs-Codes besteht in der Wahrscheinlichkeitssuche des vermutlich richtigen Weges.
Dabei ist Kopplung mit der Taktgewinnung und Signal-Erkennung möglich.

Faltungs-Codes 3

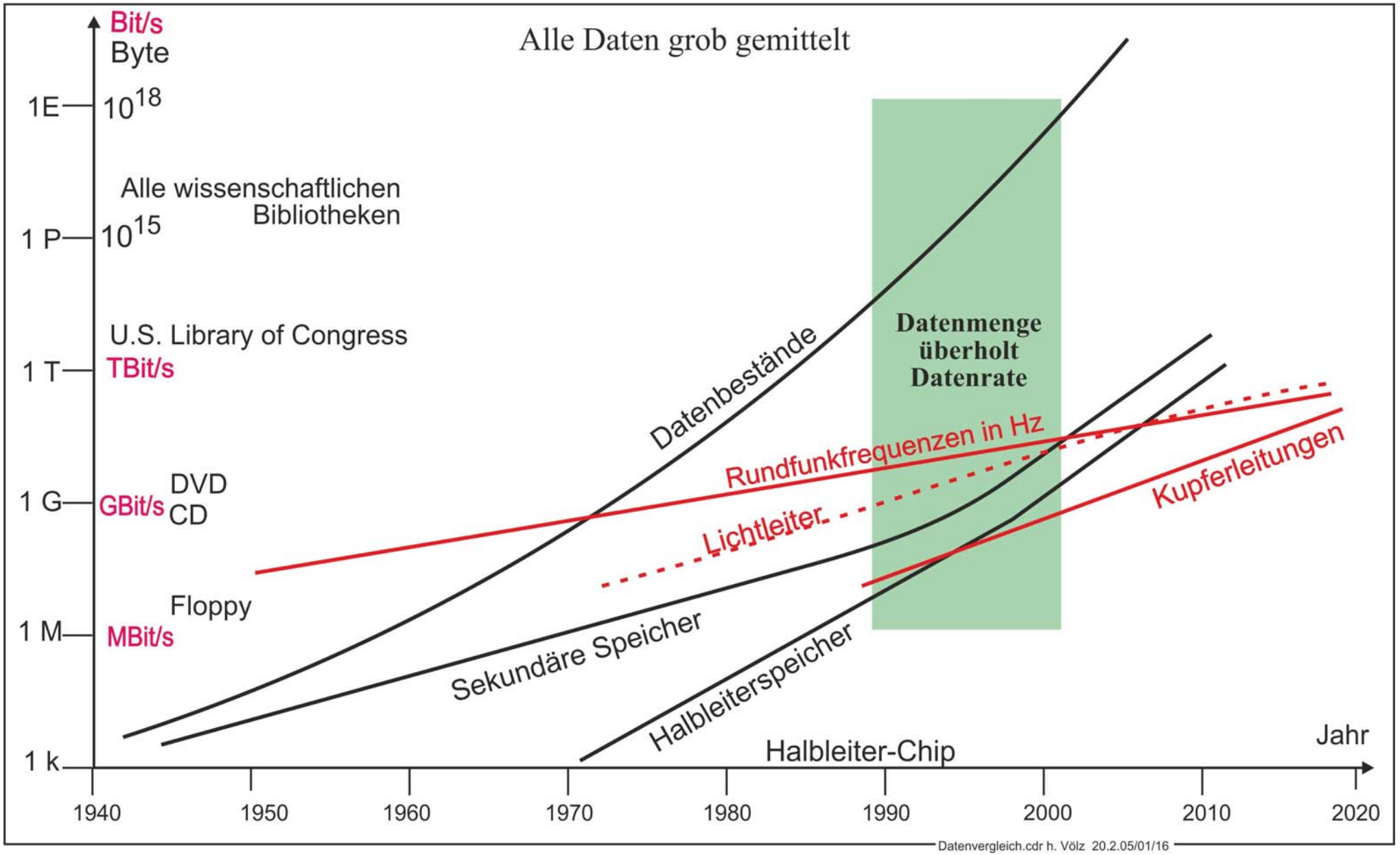
So ergibt Verfahren des PRLM (partial response likelihood maximum).
Für Faltungs-Codes gibt es verschiedene Decodierprinzipien.
Neben der Maximum-Likelihood-Decodierung auch sequentielle und algebraische Decodierung.
Häufig wird der Viterbi-Algorithmus verwendet.
Die Theorie der Faltungs-Codes ist deutlich schwieriger als die der Block-Codes.
Für die meisten Anwendungen wird sie aber nicht benötigt.
Durch die Nähe zum Automatenmodell wird ein günstiges Verfahren meist durch Rechnersuche gefunden.
Vorteil zu Block-Codes ist die Möglichkeit, große Blocklängen und große Gedächtnislängen zu nutzen.
Bündel-Fehler sind allerdings schwerer zu beherrschen, erfordern spezielles Interleaving.
Für einige Sonderanwendungen wurden Kombinationen aus Block- und Faltungs-Codes entwickelt.
Weitere Details zu den Faltungs-Codes enthält u. a. [FRI96].

Besonders komplexe und hochleistungfähige Verfahren werden bei der CD/DVD benutzt [Völ07] S. 525ff.

Komprimierungen

Lat. premere drücken, bedrängen, pressen und comprimere, compressum zusammen-, niederdrücken, niederpressen, -drängen, frei übersetzt verdichten.
Kompressor = Gerät zum Zusammendrücken von Gasen und Dämpfen, Kompresse = feuchter Umschlag
Kompression = Quetschung von Körper-Organs oder -Stelle,
Depression= Bedrückung, Niedergeschlagenheit, Ex- und Impressionismus, Express bei Bahn und Post.
Daher für Daten nicht Kompression sondern nur **Komprimierung** benutzen.

Fehlersicherung ist notwendig, dagegen scheint Komprimierung nicht unbedingt erforderlich.
Denn seit etwa 1995 verfügt praktisch jedermann im Prinzip reichlich Speicherkapazität
Doch beim Vergleich mit der Datenrate wird der Engpass deutlich erkennbar.
Sie ist bereits heute deutlich kleiner (Folge Wartezeiten)
Doch ebenfalls seit etwa 1995 wächst die Datenrate um Zehnerpotenzen langsamer als die Datenmenge.
Der Engpass nimmt also steil zu
Neue leistungsfähige Auswege können nur die Künstliche Intelligenz und Datenkomprimierung bringen
Was big data leisten wird ist noch unklar.


Bit/s
Byte
Alle Daten grob gemittelt
1E
10^{18}
Alle wissenschaftlichen Bibliotheken
1 P
10^{15}
U.S. Library of Congress
1 T
TBit/s
1 G
GBit/s
DVD
CD
Floppy
1 M
MBit/s
1 k
Datenbestände
Datenmenge überholt Datenrate
Rundfunkfrequenzen in Hz
Lichtleiter
Kupferleitungen
Sekundäre Speicher
Halbleiterspeicher
Halbleiter-Chip
Jahr
1940
1950
1960
1970
1980
1990
2000
2010
2020
Datenvergleich.cdr h. Völz 20.2.05/01/16

Die Brille

Korf liest gerne schnell und viel;
darum widert ihn das Spiel
all des zwölfmal unerbetnen
Ausgewalzten, Breitgetretnen.
Meistes ist in sechs bis acht
Wörtern völlig abgemacht,
und in ebensoviel Sätzen
läßt sich Bandwurmweisheit schwätzen.
Es erfindet drum sein Geist
etwas, was ihn dem entreißt:
Brillen, deren Energieen
ihm den Text - zusammenziehen!
Beispielsweise dies Gedicht
läse, so bebrillt, man - nicht!
Dreiunddreißig seinesgleichen
gäben erst - ein - Fragezeichen!

Problem Datensammeln

Der Unterschied von Datenmenge und -rate ist auch das akribische, z. T. sinnlose Datensammeln
Über den typisch menschlichen Sammeltrieb hat sich Twain in „Die Geschichte des Hausierers.“ belustigt
Wenn eine Sammlung vollständig zu werden scheint, bekommt er nicht das letzte, also entscheidende Stück
Schließlich sammelt er Echos. Sammlung scheint vollständig, dann wird ein „größtes Echo“ entdeckt.
Nach langem Streit über die zwei Berge tägt einer schließlich einen Beg ab.
Eine kuriosere Sammelleidenschaft von Schweigen schildert Böll in „Dr. Murkes gesammeltes Schweigen“
Auf rein ***inhaltlichen Komprimierungen,*** wie Schrift, Noten usw. wird später eingegangen.
Für die ***betont technischen*** Komprimierungen sind zwei Begriffe wesentlich (Bild):

- ***Relevanz*** (Lat. levare erleichtern, erheben und levis leicht)
  Verwandt mit *Elevator* als Heraus-, Emporheber, Aufzug sowie *Eleve* als Zögling, Schüler.
  Ferner wird Elevation als Erhöhungsgrad bei Sternen, Geschützen usw. benutzt.
  Verwandt damit ist das franz. Relief, jenes was sich hervorhebt.
  kennzeichnend sind Bedeutsamkeit, Wichtigkeit und Erheblichkeit
  Irrelevanz bezeichnet Unwesentliches, Überflüssiges.
- **Redundanz (**Lat. unda die Welle und Undulation die Wellenbewegung zurück.
  entspricht Zurückwogen. Verwandt damit ist das Wassermädchen, die Nixe Undine,
  Franz. *ondulieren* und *sondieren*, später auch die *Sonde*.
  Technisch verstanden vor allem Üppigkeit, Überfluss, Überreichlichkeit und Übermaß.

Redundanz

D*s *st *n T*xt

keine Vokale

Dies ist ein Text

es genügt die obere Hälfte

Störungen sind unwirksam

Irrelevanz
von Font und Auszeichnung

Dies ist ein Text

Dies ist ein Text

Dies ist ein Text

Dies ist ein Text

Dies ist ein Text

Dies ist ein Text

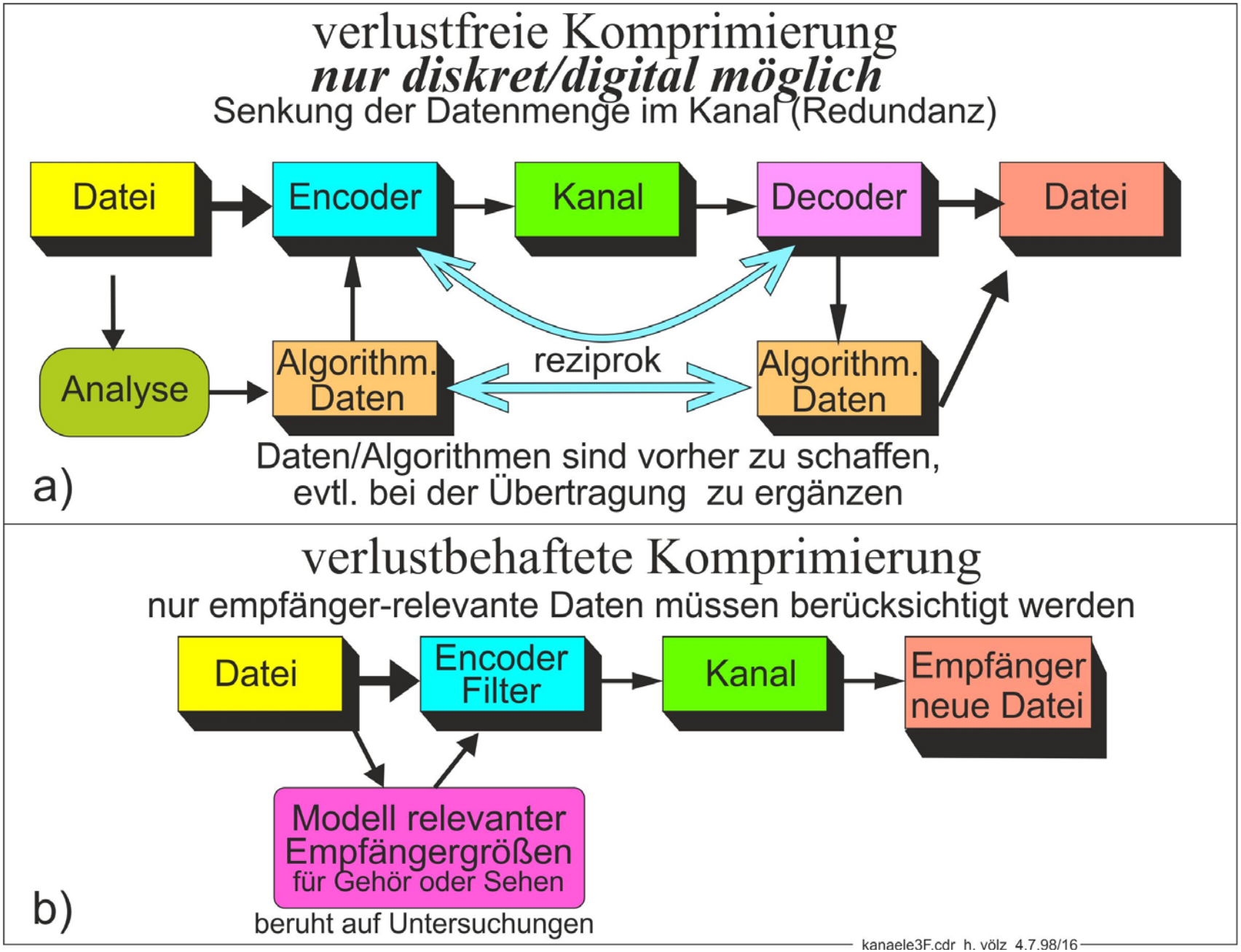
Dies ist ein Text

ocr2.cdr h. völz 30.1.94

Verlustfrei ⇔ verlustbehaftet

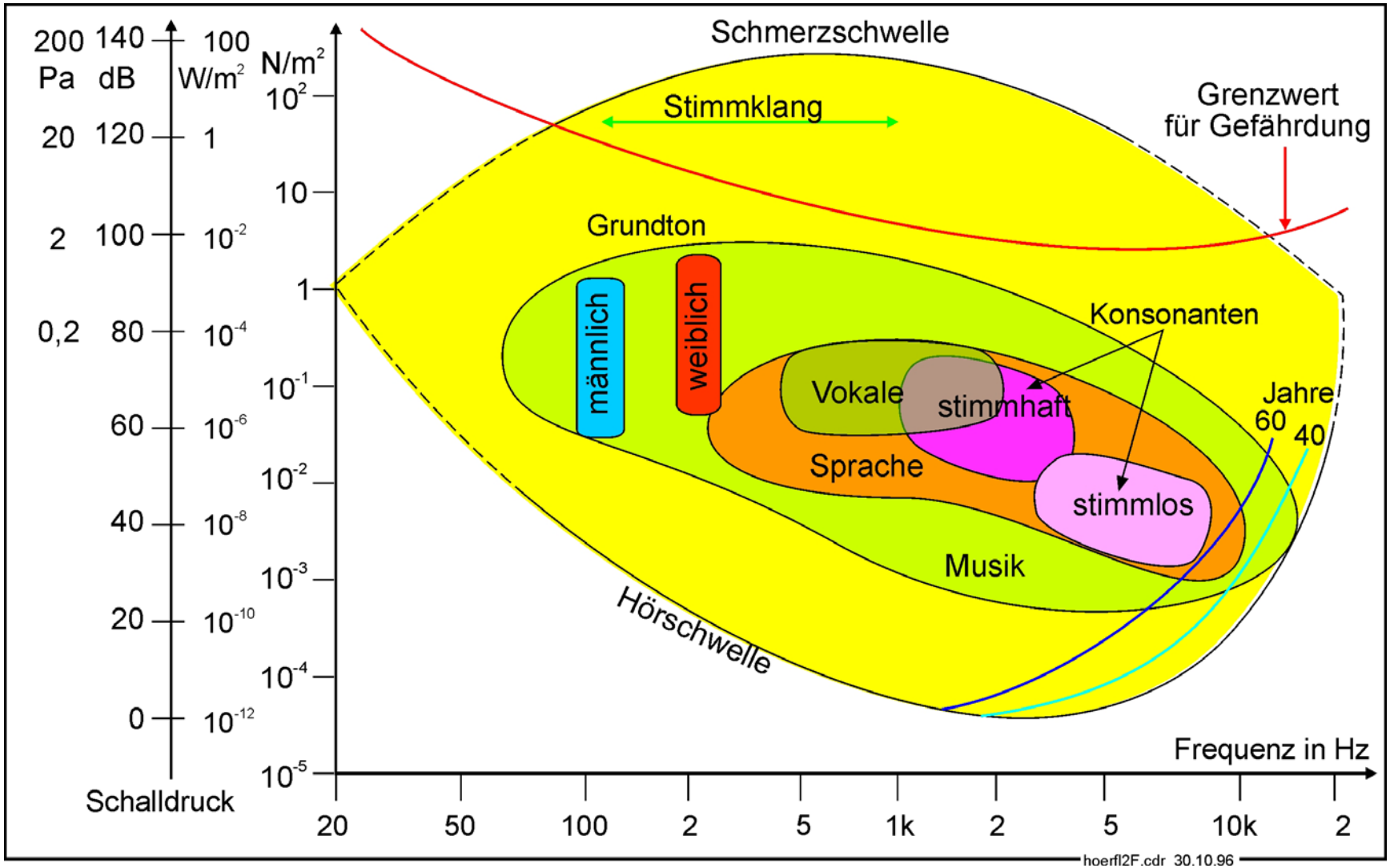
Irrelevanz, Redundanz haben beachtlichen Bezug zu Ockhams Rasiermesser, nicht Notweniges abschneiden dort auch Abstraktion und Reduktion.
Bei Fehlerbehandlung wird Redundanz in genau bestimmter Weise hinzugefügt wird
Bei der Komprimierung dagegen soviel wie möglich oder zulässig entfernt werden.
Dafür sind ***drei Methoden*** und ***zwei Kanalvarianten*** (Bild nächste Seite) zu unterscheiden:

1. ***Verlustbehaftete*** Methoden entfernen irrelevante Daten, Fakten und Erscheinungen.
   Vorwiegend jene, die der *Empfänger nicht benötigt oder benutzt*, z. B. *jemand nicht wahrnehmen kann*.
   Dazu muss ein geeignetes ***Modell des Empfängers*** geschaffen werden.
   Die Methode kann ***t-kontinuierliche*** und ***diskrete, digitale*** Daten angewendet werden
   Daher auch Oberbegriffe, Klassenbildung und Axiomatik bei der Z-Information anwendbar.
2. ***Verlustfreie*** Verfahren suchen nur teilweise nach redundanten Daten.
   Sie nutzen sie ***Umrechnungen, Algorithmen*** und ***Links*** auf im Daten, die im Sender vorhanden
   Vor der verdichteten Übertragung erfolgt die Umrechnung.
   Dazu ist meist zunächst mit einer ***Analyse*** die optimale Variante zu ermitteln
   Diese Verfahren sind nur auf ***diskrete, digitale*** Daten anwendbar.
   Theoretisch ist jede endliche Datei auf ***1 Bit*** zu reduzieren.
3. ***Inhaltliche*** Verfahren gibt es vor allem für die Sprache
   Dazu gehören z. B. ***Kurzfassungen, Inhaltsangaben, Annotationen*** usw.
   Bei der Musik sind es *Themen, Motive, Ouvertüren* usw.
   Dabei wirken ganze Texte oder kurze Musikstücke wie ein einziges Zeichen.

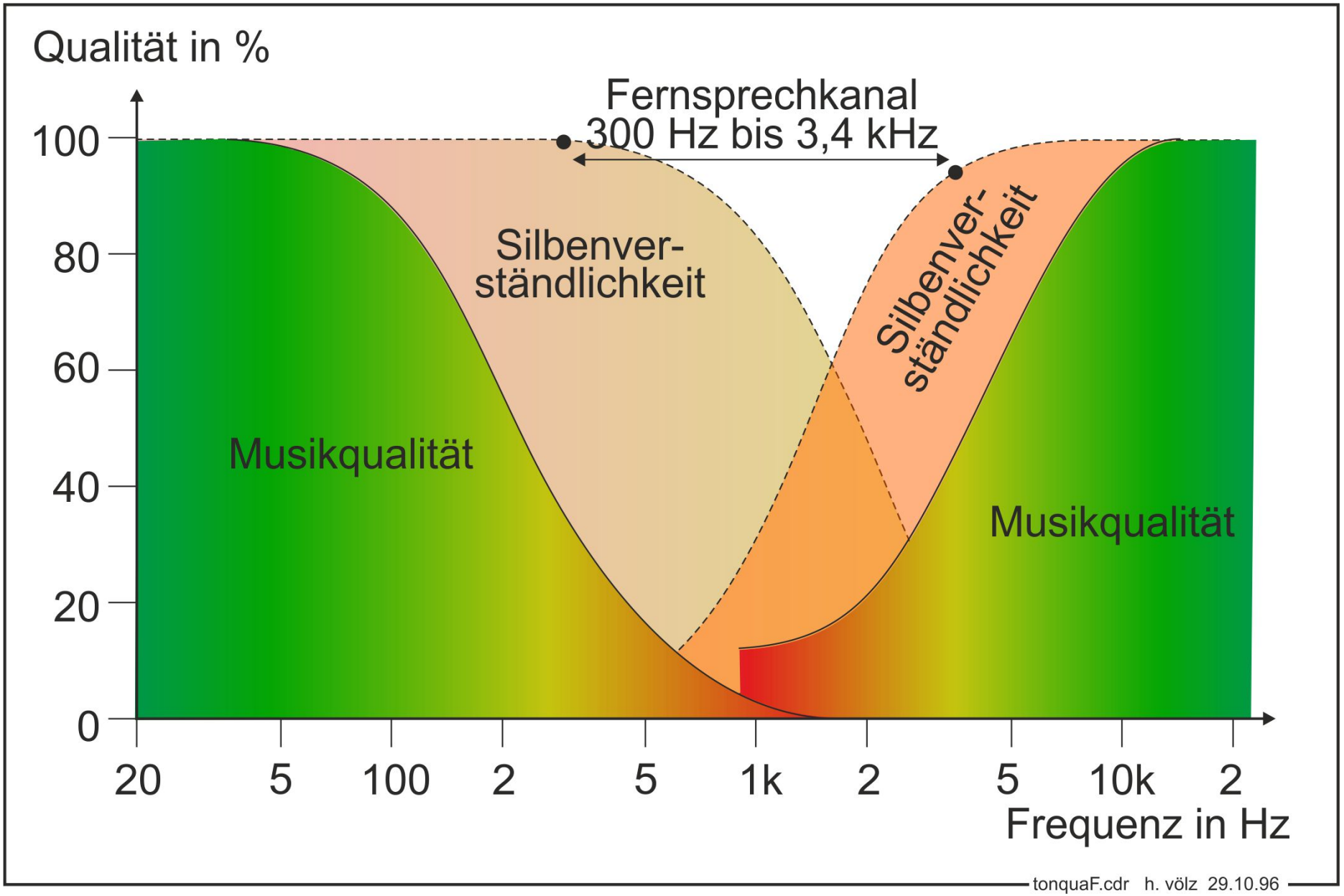

verlustfreie Komprimierung
nur diskret/digital möglich
Senkung der Datenmenge im Kanal (Redundanz)
Datei
Encoder
Kanal
Decoder
Datei
Analyse
Algorithm. Daten
reziprok
Algorithm. Daten
Daten/Algorithmen sind vorher zu schaffen,
evtl. bei der Übertragung zu ergänzen
a)
verlustbehaftete Komprimierung
nur empfänger-relevante Daten müssen berücksichtigt werden
Datei
Encoder Filter
Kanal
Empfänger neue Datei
Modell relevanter Empfängergrößen
für Gehör oder Sehen
beruht auf Untersuchungen
b)
kanaele3F.cdr h. völz 4.7.98/16

Modell für Hören

Für verlustbehaftete Komprimierung sind Modelle für unser Hören und Sehen wichtig.
Bei akustischen Übertragungen je nach Anwendung Frequenzbereich und Dynamik.



Klangqualität

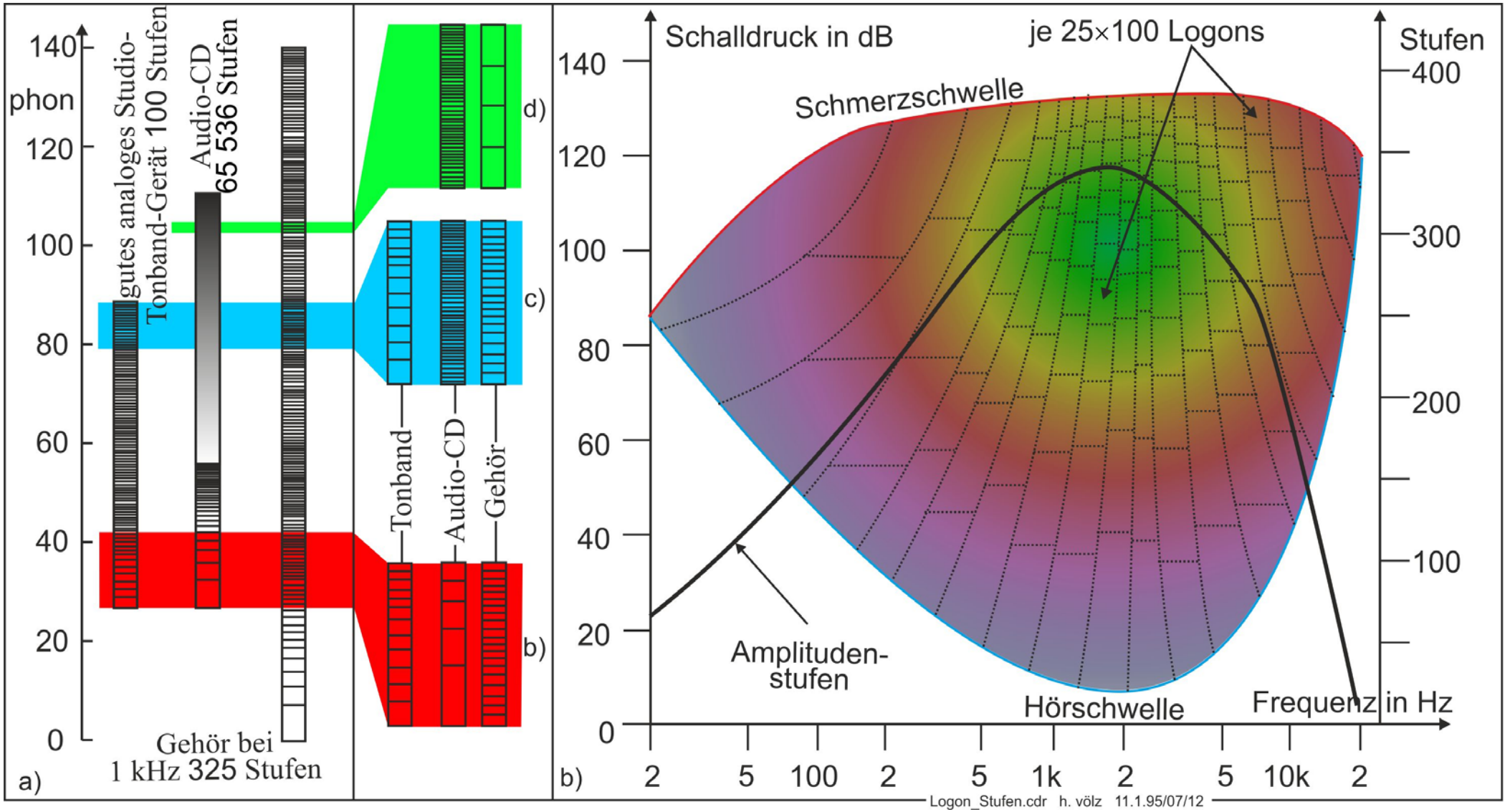


Amplitudenstufen ⇔ Logons

Der Klirrfaktor muss meist klein gehalten werden. Ansonsten gilt das logarithmische Weber-Fechner-Gesetz
Ähnlich auch Logons (2-fach p-kontinuierlich)
In ihrer Fläche sind keine Änderungen der Frequenz und/oder Lautstärke hörbar.
Für die Darstellung im folgenden Bild sind jeweils 25×100 Logons zusammengefasst.
Sie sind um 1000 Hz und 90 dB Schalldruck besonders klein und dicht (Abhörlautstärke!)
Bei 1000 Hz Verteilung der unterscheidbaren Amplitudenstufen

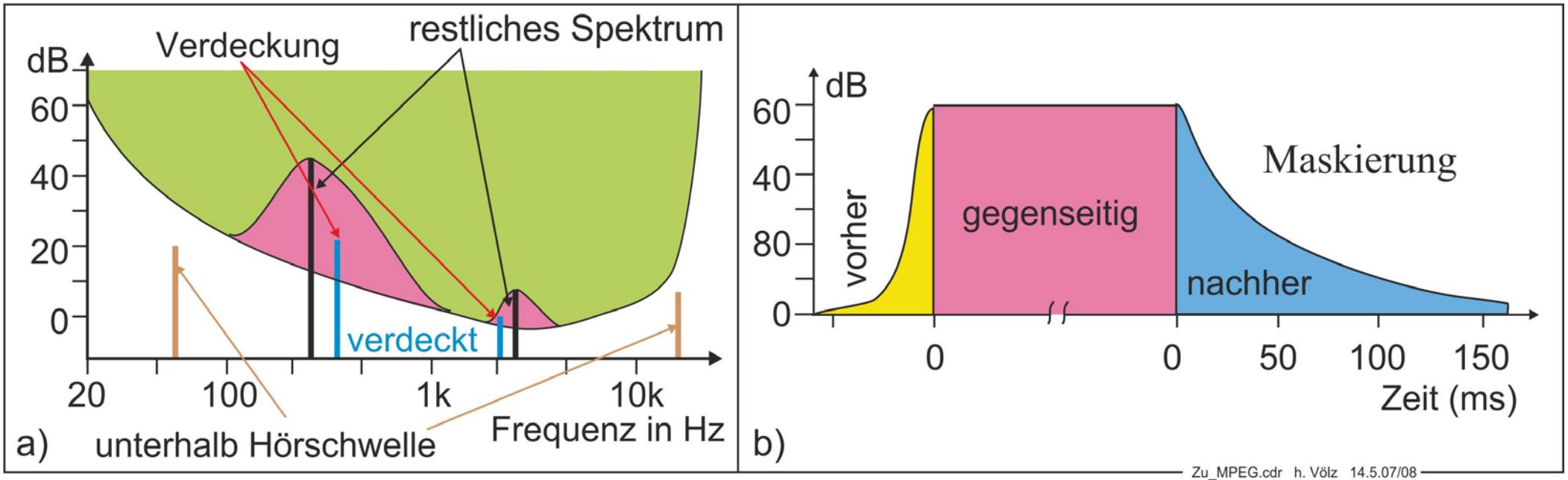
Die Auswirkungen zeigen sich deutlich beim Vergleich von Gehör, Audio-CD und Tonband
Deshalb können Studio-Magnetband-Aufzeichnungen teilweise die Qualität der CD (SCD) übertreffen.

Bei großen Lautstärken können aber unnötig viele kleine Stufen zu einer zusammengefasst werden.
Anwendung u.a. bei MP3 (einstelbar s.u.) , bei Mobiltelefonen sogar nur **5** Lautstärkestufen.


a)
phon
140
120
100
80
60
40
20
0
gutes analoges Studio-Tonband-Gerät 100 Stufen
Audio-CD 65 536 Stufen
Gehör bei 1 kHz 325 Stufen
d)
c)
b)
Tonband
Audio-CD
Gehör
b)
Schalldruck in dB
je 25×100 Logons
Stufen
400
300
200
100
Schmerzschwelle
Amplituden-stufen
Hörschwelle
Frequenz in Hz
2
5
100
2
5
1k
2
5
10k
2
Logon_Stufen.cdr h. völz 11.1.95/07/12

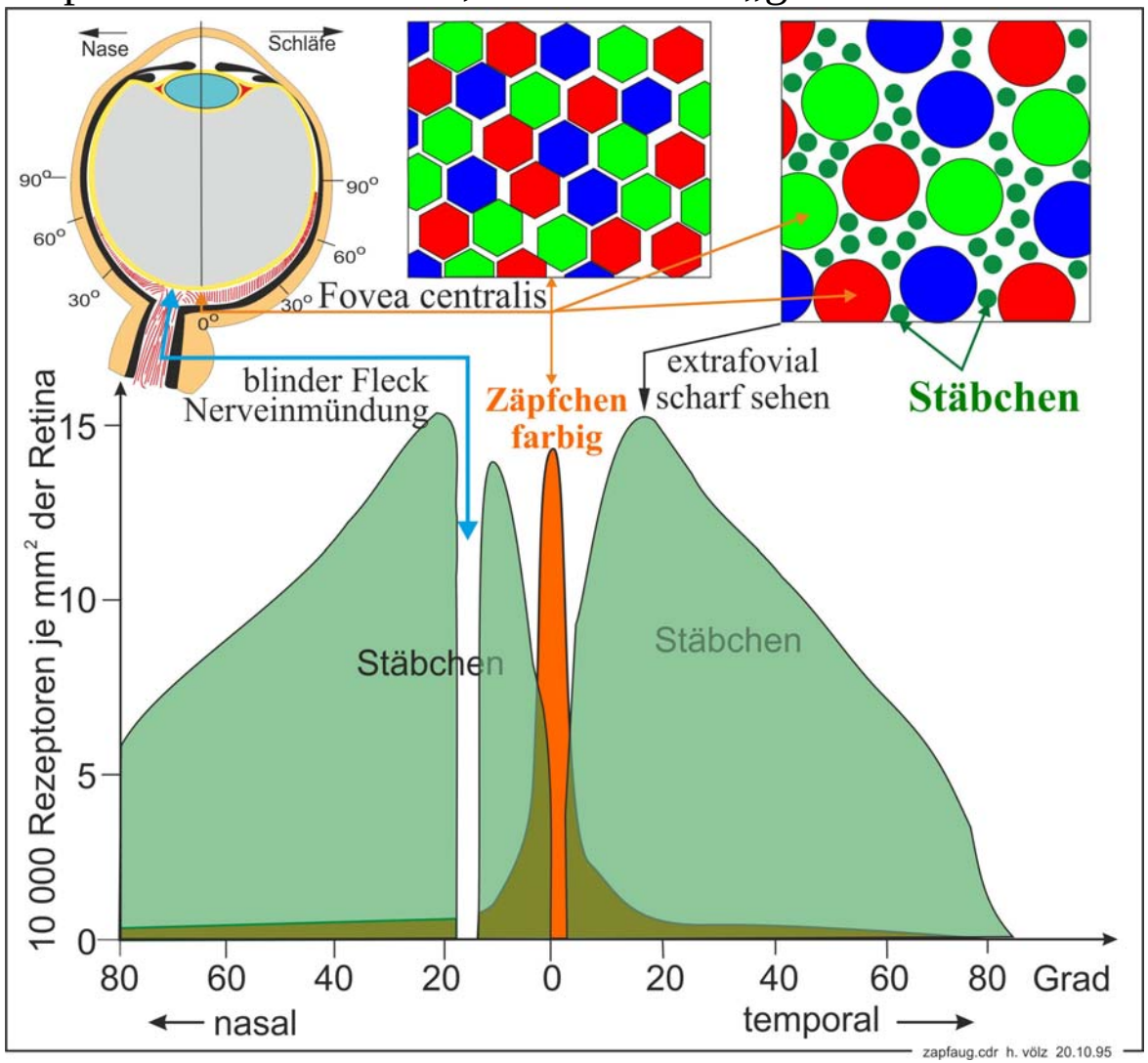
Weitere Varianten

Neben Amplituden- und Frequenzstufen sind Maskierung und Verdeckung geeignet MP2 und MP3.

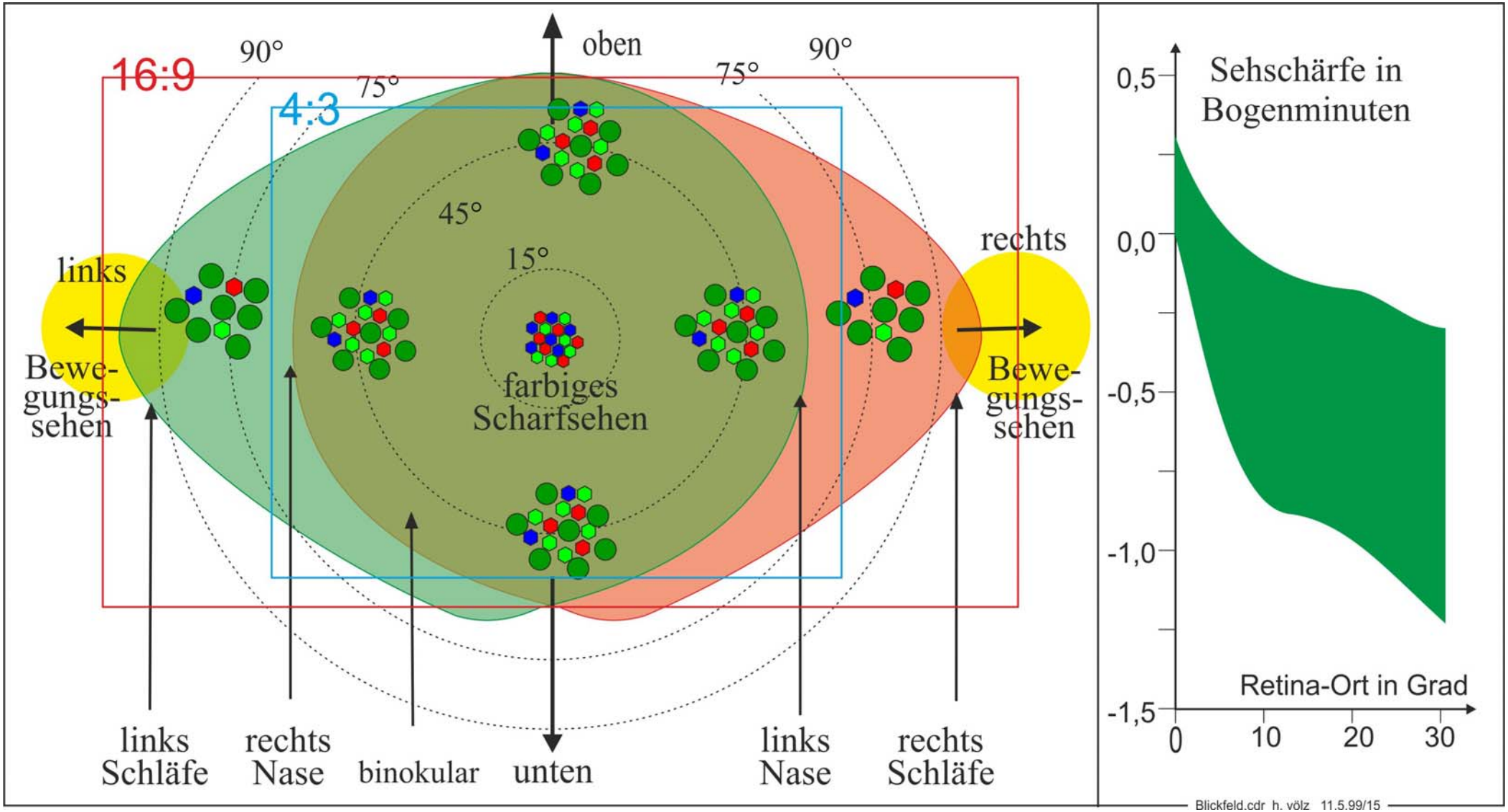


Modell Sehen

Beim Sehen gelten deutlich andere Zusammenhänge.
Nur in einem kleinen Raum-Bereich sehen wir farbig, noch kleiner bei feinen Details.
Dichteverteilung der Zäpfchen für Farbsehen, Stäbchen nur „grau“ und sehr klein für optimal scharf

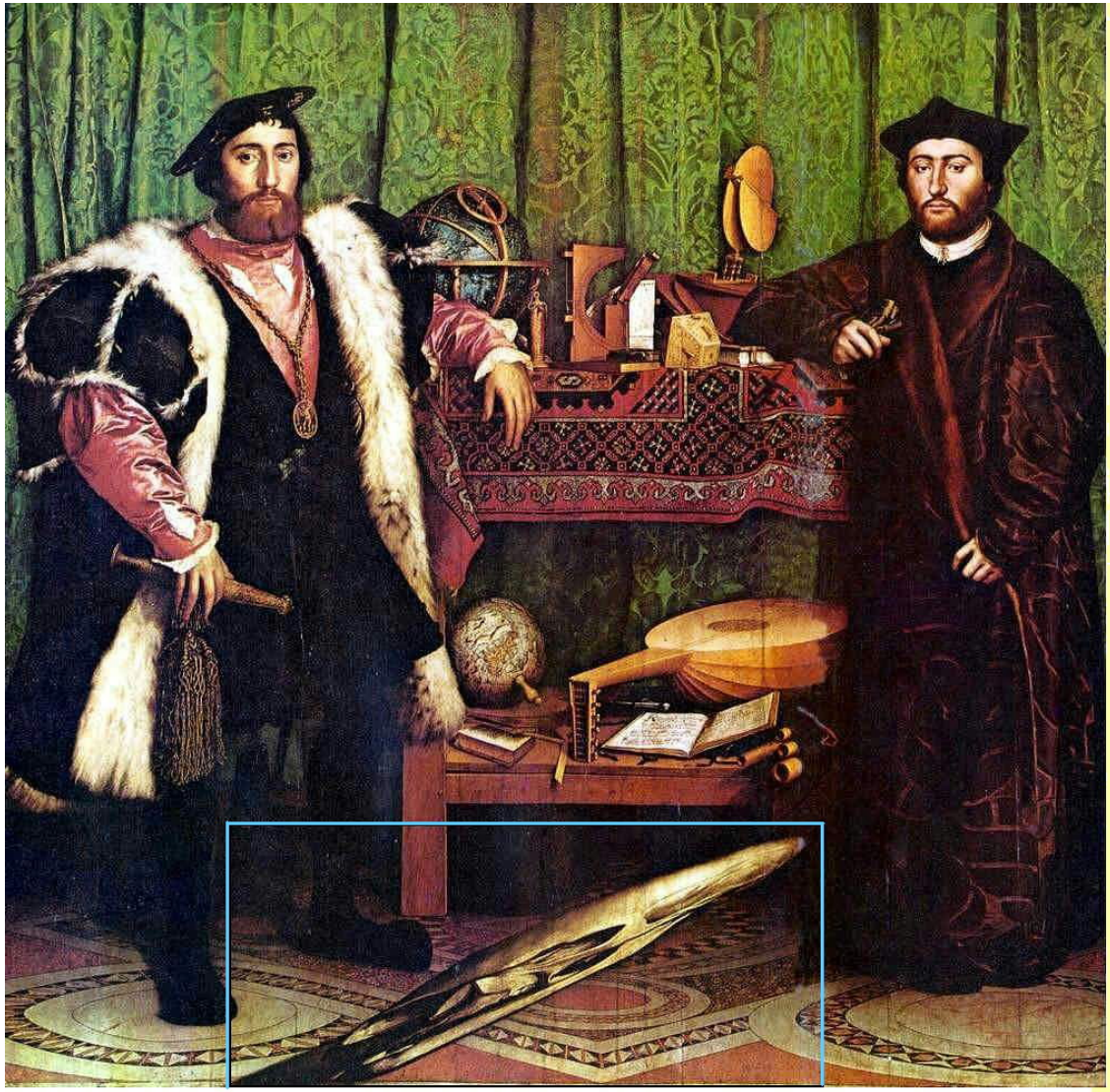


Unser Sehfeld

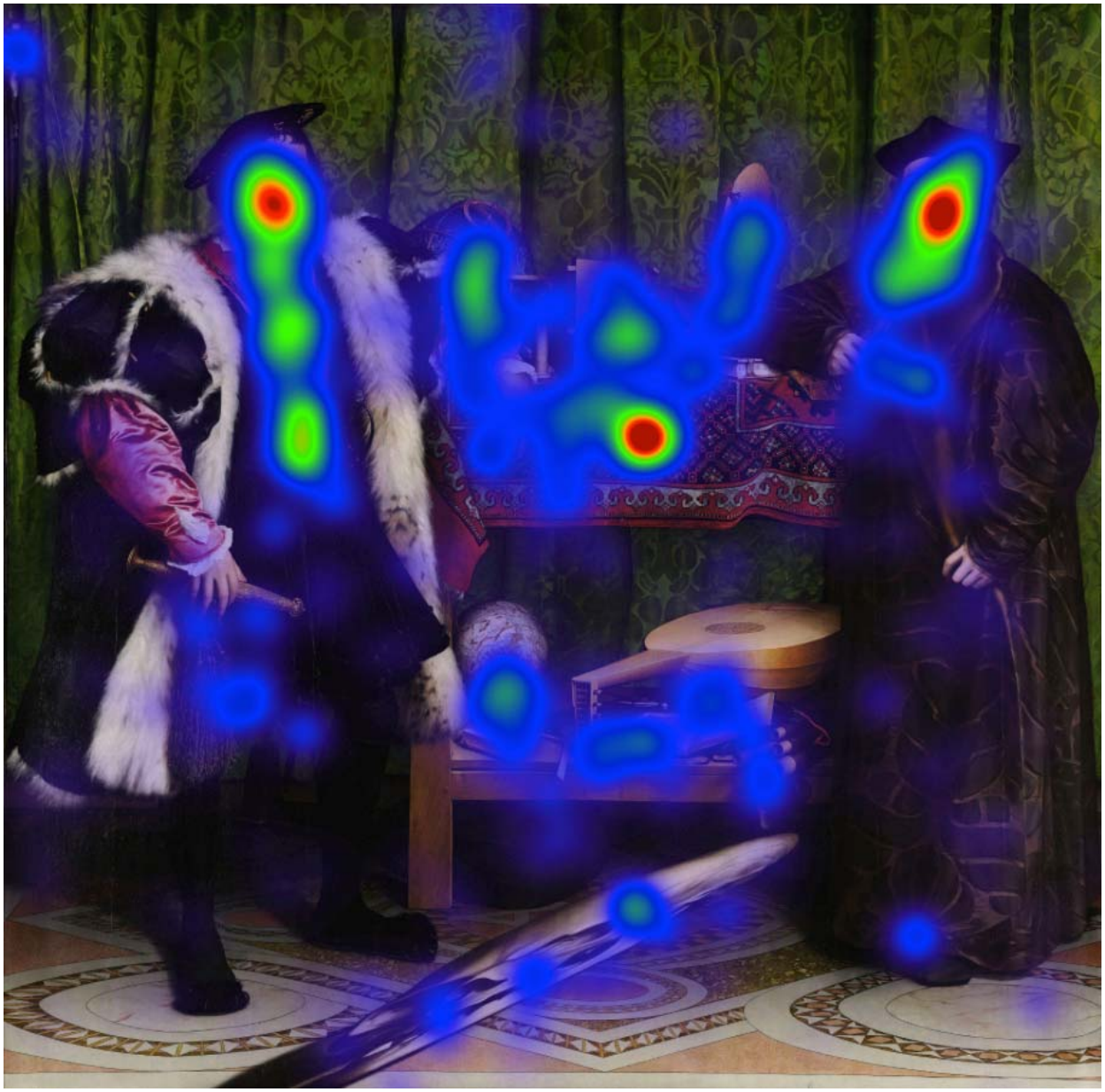


Abfolge beim Sehen

Bei einem neuen Bild verschaffen wir uns zunächst mit dem großen Blickfeld einen Überblick
Dan erfolgt schrittweises analysieren von Details mit dem scharf sehenden Farbbereich.



Feinanalyse



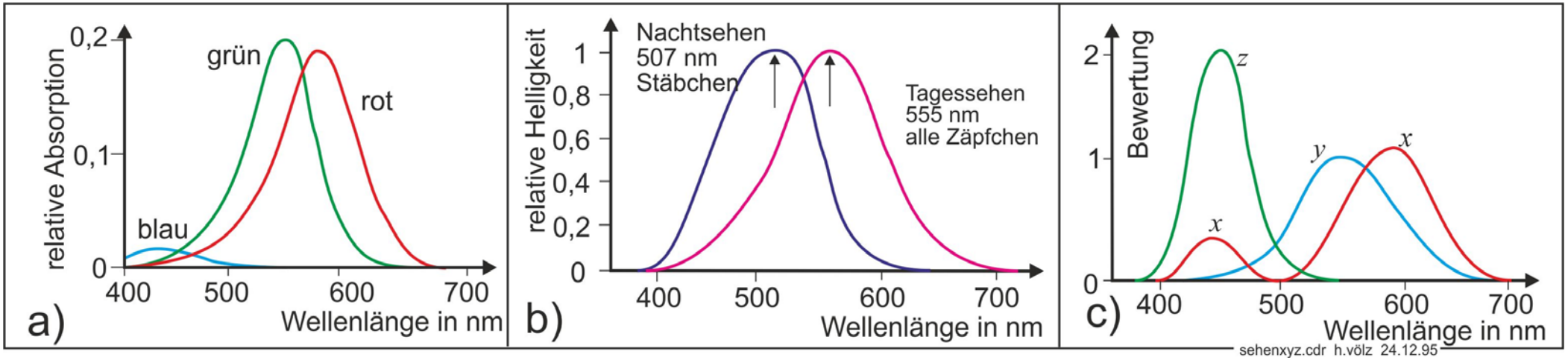
Holbeins d. Jüngere: „Die Gesandten“

Französischer Gesandte Jean de Dinte-Ville und seinen Bevollmächtigter und Freund, der Bischof Georges de Selve.

Am Boden – oben blau umrandet – stark verzerrt das Bild eines Totenschädels, der erst bei der Entzerrung (rechts) gut zu erkennen ist. Vermutlich spielte Holbein damit auf den bevorstehenden Tod de Dintevilles an.
[Zho15].

Gestaffelte Betrachtung verlangt Rechteckbilder konstanter Auflösung. Hierbei keine Redundanz möglich!

Farbwahrnehmung



Sie ist weitgehend durch die drei Zäpfchenarten mit den Absorptionskurven bestimmt.
Auffällig ist die geringe Absorption der blauen Zäpfchen(a).
Der wahrgenommener Helligkeitsverlauf beim Nachtsehen mit Stäbchen ist gegenüber der Summenkurve aller Zäpfchen bei Tageslicht leicht in Richtung blau verlagert (b).
Die einzelnen Zäpfchen wirken für gesehene Farbe zusammen (c).
Zu den Primärfarben x (Rot), y (Grün) und z (Blau) gehören die Gewichten g_x, g_y und g_z. Daher Farbvalenz:

$$f = g_x \cdot x + g_y \cdot y + g_z \cdot z.$$

So kann jede Farbvalenz als Ort in einem 3D-RGB-Raum dargestellt werden.
Weil das aber recht aufwändig ist, begann bereits in den 1920er Jahren die CIE (Commision Internationale de l'Éclairage) eine einfachere Darstellung zu entwickeln.
1931 für CIE-RGB-Modell drei physikalisch nicht existierende, virtuelle Primärfarben eingeführt:

$$\bar{x}(\lambda),\ \bar{y}(\lambda),\bar{z}(\lambda)\ .$$

CIE-RGB-Modell

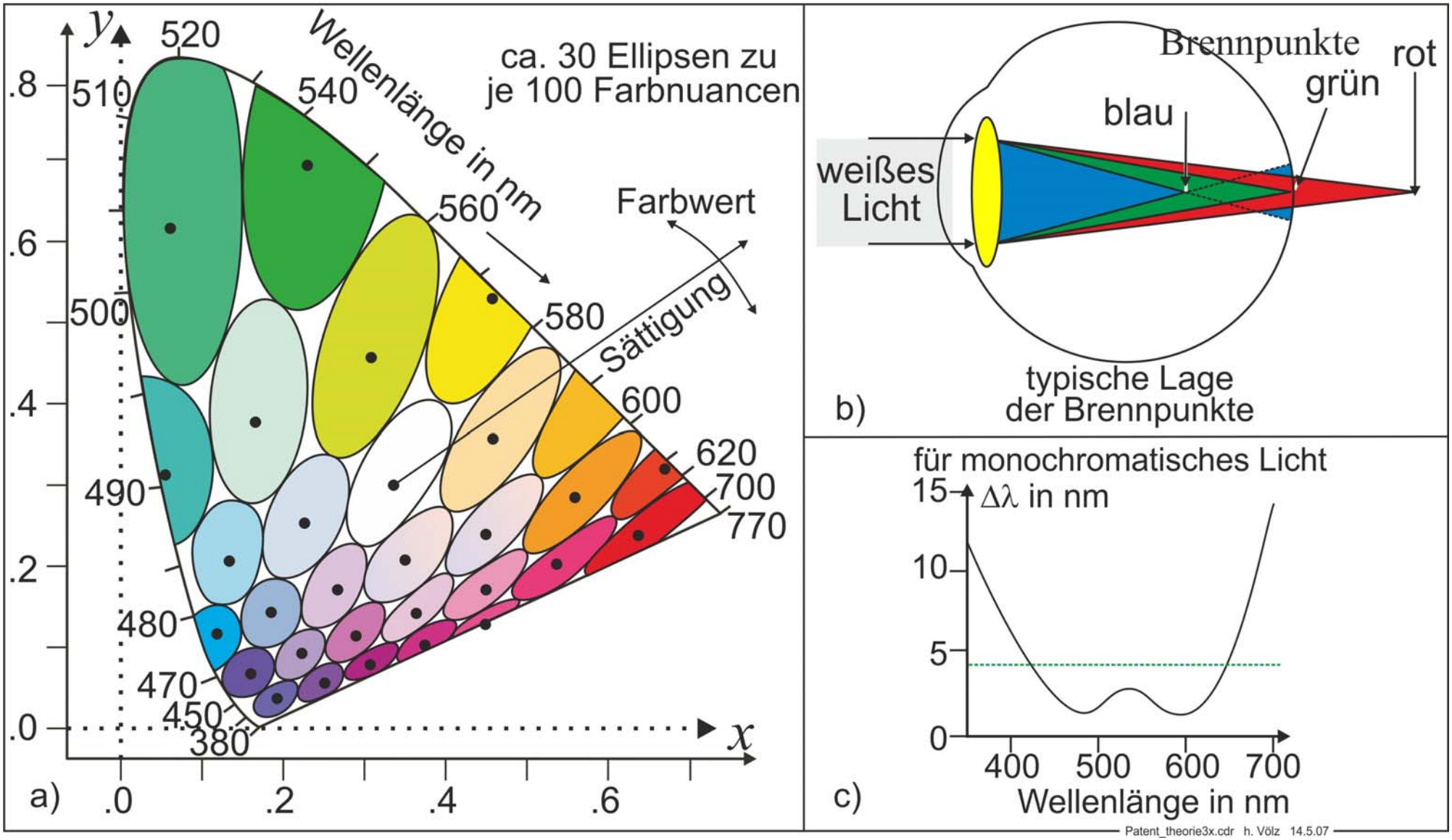
Mit dem Spektralverlauf $S(\lambda)$ folgen durch Integration die Normalfarbenwerte des Modells:

$$X = k \cdot \int S(\lambda) \cdot \bar{x}(\lambda) \cdot d\lambda \quad Y = k \cdot \int S(\lambda) \cdot \bar{y}(\lambda) \cdot d\lambda \quad Z = k \cdot \int S(\lambda) \cdot \bar{z}(\lambda) \cdot d\lambda \,.$$

Ihre 3D-Darstellung ist unhandlich. Deshalb wird die absolute Helligkeit $A = X + Y + Z$ eingeführt.
Damit ergeben sich die Normfarbanteile zu

$$x = X/A; \quad y = Y/A \text{ und } z = Z/A.$$

Ihre Werte reichen von 0 bis 1. Ihre Summe ist immer 1, deshalb genügen 2 Werte für alle Farbvalenzen.
So entsteht das CIE-Farbdreieck (nächstes Bild). Dem „Mittelpunkt“ entspricht weißes (unbuntes) Licht.
Alle etwa 150 monochromen (= gesättigten) Farben liegen auf dem hufeisenförmigen Rand.
Auf unteren Geraden befinden sich die Mischfarben aus rot und blau, u. a. violett.
Im Innern liegen alle möglichen Mischfarben, die in Richtung zu Weiß pastellartig werden.
Auf der Sättigungsgeraden vom Mittelpunkt bis zum Rand sind etwa 20 Werte unterscheidbar.
Dadurch entstehen (ähnlich dem Schall) ellipsenförmige Logons nicht wahrnehmbare Farbänderungen
In Richtung der Helligkeit kann das Dreieck 600 Stufen mit einem Kontrast von 1 : 200 annehmen.


y
520
510
540
500
560
580
600
620
700
770
490
480
470
450
380
.8
.6
.4
.2
.0
x
.0
.2
.4
.6
a)
Wellenlänge in nm
ca. 30 Ellipsen zu
je 100 Farbnuancen
Farbwert
Sättigung
Brennpunkte
rot
grün
blau
weißes
Licht
typische Lage
der Brennpunkte
b)
für monochromatisches Licht
15
Δλ in nm
10
5
0
400
500
600
700
Wellenlänge in nm
c)
Patent_theorie3x.cdr h. Völz 14.5.07

Folge für Farbfernsehen

Für das Farbfernsehen musste auf Schwarzweiß-Fernsehern ein leidliches Grau-Bild entstehen.
Dazu wurde ein Helligkeitssignal (= Luminanz *Y;* ~~nicht yellow~~) bestimmt ≈ Helligkeitsverlauf $V(\lambda)$
Wir unterscheiden Farben weniger gut als Helligkeiten → farbabhängige Brennpunkte unserer Augenlinse.
Deshalb wird für die beiden Farbdifferenzsignale (Chroma) C_b und C_r nur die halbe Bandbreite benutzt.

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \frac{1}{256} \cdot \begin{bmatrix} 77 & 150 & 29 \\ 151 & -110 & -21 \\ 44 & -87 & 131 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

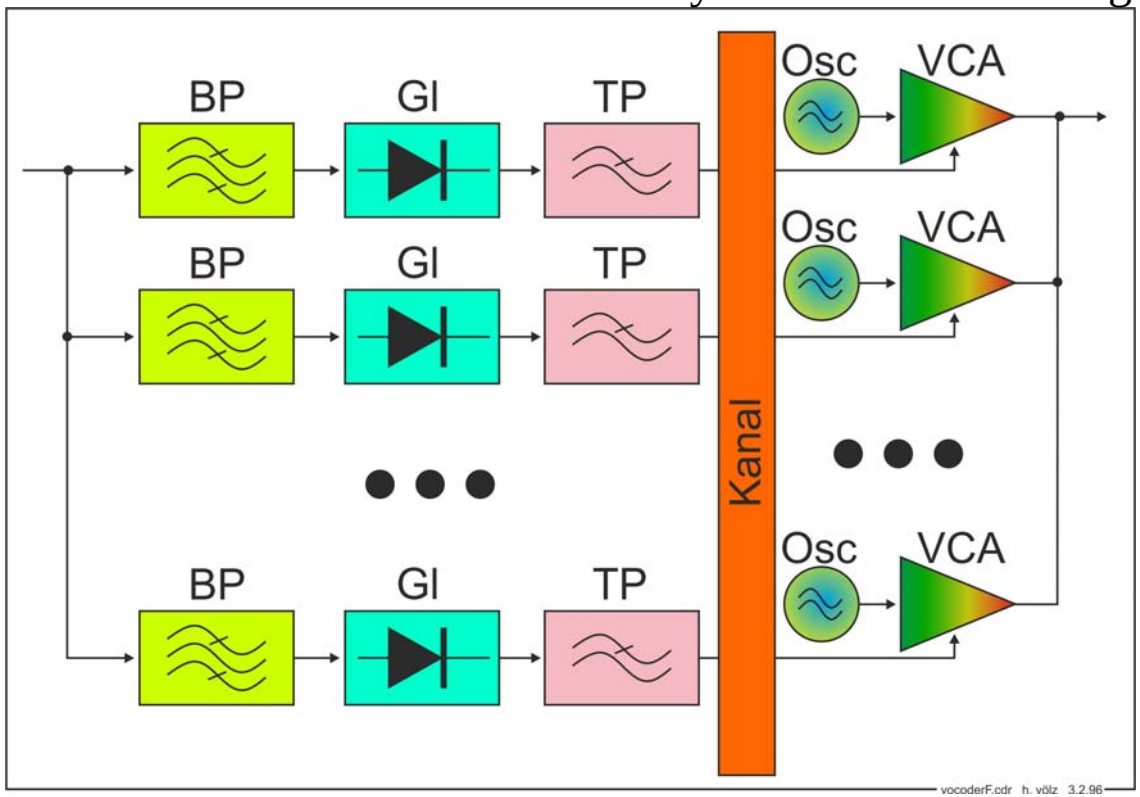
Für die kontinuierliche Übertragung gilt folgt ähnlich das YUV-Signal mit den Auflösungen 4:1:1.

$$Y = 0{,}299 \cdot R + 0{,}587 \cdot G + 0{,}114 \cdot B,\ U = 0{,}493 \cdot (B - Y) \text{ und } V = 0{,}877 \cdot (R - Y).$$

Mehr Details [Völ99] S. 26ff.

Verlustbehaftete Schall-Komprimierungen

Eine erste hohe Sprachkomprimierung ist der bereits 1920 entstandene Vocoder.
Dabei werden mehrere (z .B. 20) Frequenzkanäle mit Bandbreite ≈ 200 Hz und 30 dB Dynamik angelegt
Es werden nur deren zeitlich Amplituden übertragen.
Bei der Wiedergabe steuern sie die Amplitude der entsprechenden Oszillatoren.
So entsteht eine Komprimierung von >300 : 1. Dennoch wird eine Silbenverständlichkeit ≥ 90 % erreicht.
Dabei geht jedoch jede Individualität und Interpretation des Sprechers verloren.
Insbesondere ist keine Sprecher-Erkennung möglich.
Heute wird die Technik nur noch für Musik z. B in Synthesizern als Effektgerät benutzt.



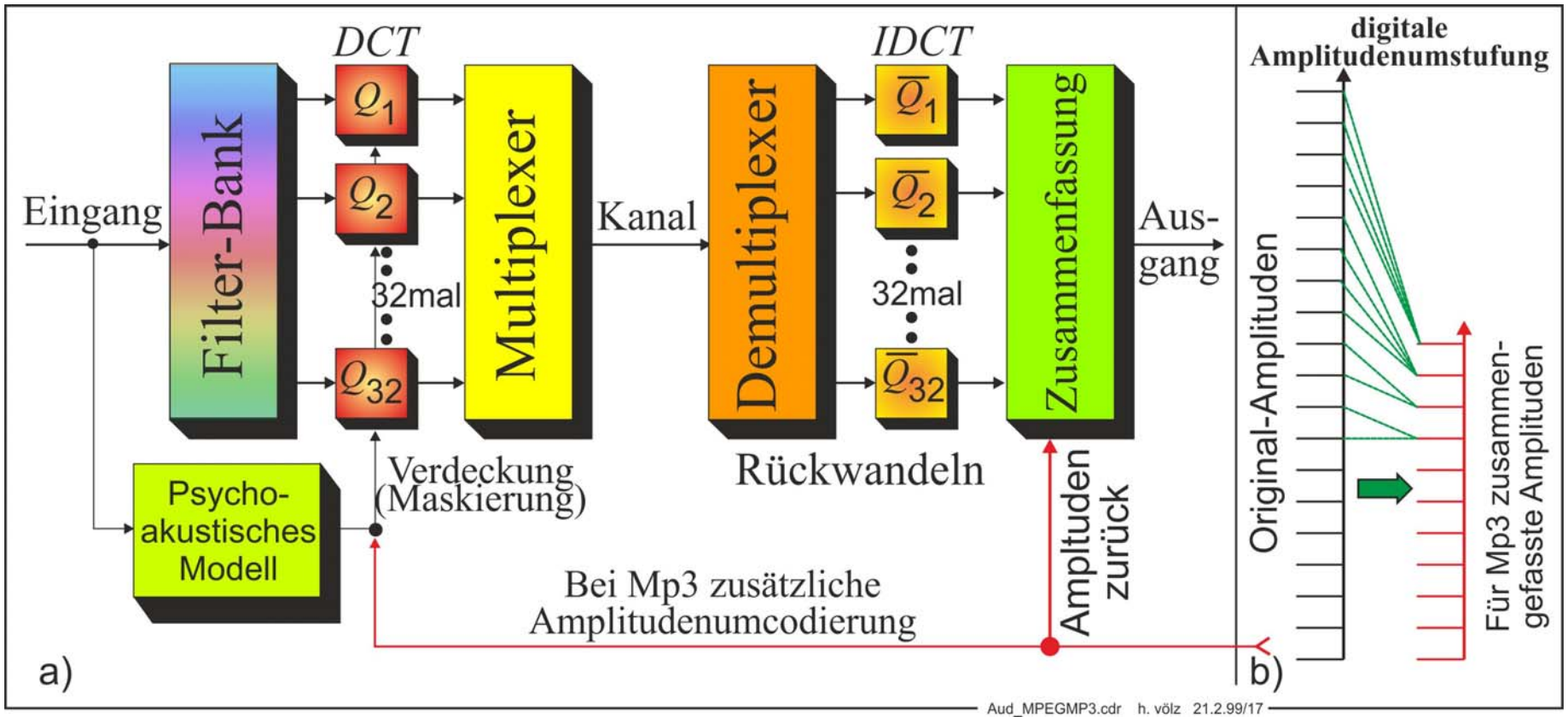
Dolby- und ähnliche Verfahren

Bei hochwertigen Musikaufnahmen steuert der Tonmeister den Aufzeichnungspegel mittels der Partitur.
entsprechend der geringeren technischen Dynamik sollen die originalen Dynamiksprünge erhalten bleiben.
So wird vor dem Lautstärkanstieg den Pegel unhörbar langsam abgesenkt u. nachher wieder erhöht.
Für Speicher (Schallplatte, Heimtonbandgerät) wird das z. T. mit automatischen Dynamikregelungen ereicht
Dazu entstanden ab den 70er Jahren mehrere Dolby-Verfahren mit Com- und Expandern.
Prinzip ≈ reziproke Dynamikregelung [Völ58a]).

Ein deutlich anderes Verfahren ist DNL (dynamic noise limiting).
Bei der Wiedergabe geringer Lautstärke werden die Höhen (die dann ja unhörbar sind) und damit auch das Rauschen abgesenkt [Völ07].

MPEG

(moving picture expert group) entstand primär für Video ab Ende der 1980er Jahre.
Dabei waren auch Audio-Komprimierungen notwendig, die getrennt von Video als mp2 bezeichnet werden.
Sie nutzen vor allem die Verdeckungen durch laute Frequenzen und z. T. auch Maskierungen(s.o.)
So ergibt sich das Prinzip des Bildes, mit dem zunächst nur Verdichtungen um 1:4 möglich waren.
Mp3 ist eine Weiterentwicklung durch Zusammenfassunf von Amplitudenstufen
Für die Frequenzkanäle wurde zunächst die übliche Fourier-Transformation eingesetzt.
Mit der Digitalisierung tritt dann die DCT (discrete-cosine-transformation) an ihre Stelle.



ASCII-Code

Code: lat. cauda Schwanz, Schleppe; caudex Baumstamm, Strafblock, Buch, Bibel, in Frankreich im Sinne von Gesetz, u. a. Code Napoleon, Code de commerce, Coda musikalischer Schluss.

Schall gibt es vielfältig: Sprache, Gesang, Musik, Oper, Hörspiel, Audioart, Warnsignale, Tierlaute usw.
Hierfür sind die Komprimierungen des vorigen Abschnitts grundsätzlich geeignet.
Jedoch in zwei Fällen schuf sich der Mensch effektivere Lösungen:
1. entstand für die Kommunikation die ***Sprache*** und zu ihrer dauerhaften Speicherung die ***Schrift***
2. Ähnliches geschah für ***Gesang*** und instrumentale ***Musik*** mit den ***Noten***.
Dabei werden nur spezielle Schalle ausgewählt, dann ***Redundantes ausgesondert***.
Von der t-kontinuierlichen Sprache werden nur wenige Laute diskret ausgewählt,
und mit Buchstaben zu einfachen ***Zeichen*** (Z-Information) umgesetzt, z.B. in **ASCII-Code** (american standard code for Information interchange): 7-Bit-Code mit 128 Zeichen, erstmals 1967 veröffentlicht und zuletzt 1986 für alle Sprachen aktualisiert.
Dabei gehen alle Besonderheiten, wie Lautstärken, Betonung, Tonhöhe (männlich, weiblich, kindlich), Melodie, Pausen und Rhythmen verloren. Etwas genauer ist die *phonetische Schrift*.
Bei erneuter Wiedergabe (Lesung usw.) müssen sie als individuelle *Interpretation hinzugefügt* werden.
Besonders auffällig ist das bei technisch, *elektronisch generierten Stimmen*.
Dafür ermöglichen die starken Einschränkungen eine sehr hohe Komprimierung.
Ein 2-Minuten-Text als Audio-Signal benötigt mehr als **1 MByte**
Die entsprechenden A4-Seite als ASCII-Code weniger als **2 KByte**. (1:500)
Dabei ist sogar noch der Schriftfont wählbar, geht bei OCR (optical character recognition) wieder verloren.
Weniger gut funktioniert das bei der Spracherkennung.

MIDI

Für Musik war die Entwicklung des **Moog-Synthesizer**s von 1964 wesentlich.
Bald entstanden ***Keyboard-Varianten*** und elektronisch **gesteuerte Rhythmus-** und **sonstige Instrumente**.
1982 entstand MIDI (musical instrument digital interface) zunächst für **Datenaustausch** (Kommunikation) zwischen Computer und Keyboards, Synthesizer, Rhythmusinstrumente usw.
das **Interface** V 24 bzw. RS 232 mit 31,25 kHz (1/32 MHz) mit 8 Daten-Bit + 1 Stopp-Bit wurde benutzt.
Drei Modi: *Omni*: an alle Geräte; *Poli*: nur adressierte Geräte; *Mono*: für Änderungen in einem Gerät.
1991 wurden Instrument-Adressen exakt festgelegt.
Jedoch wurde die **Musikkomprimierung** mit **Notenbezug wichtiger**.
Es wurden folgende Bezüge festgelegt:

Daten-Byte	
Tonhöhe	60 mittleres C (Klavier 24-108)
Lautstärke	64 „normal“, 0 aus, 1 ppp, 127 fff. Sie wird durch die Anschlagsdynamik als Δt zwischen zwei Kontakten bestimmt.
Status-Byte	
Bit 0- 3	Für 16 Kanäle, d. h. unterschiedliche Keyboards, Geräte.
Bit 4 - 7	Z. B.: 1000 = Note aus, 1001 = Note ein, 1010 = polyphon, 1100 = Programm-Änderung.

Ferner können Klangbänke für etwa hundert Instrumente ausgewählt werden.
Vor allem durch vorhandene Obertöne sowie Ein- und Ausschwingvorgänge gekennzeichnet.
Spezieller Soundsamples nähern sich z. B. ganz spezielle Konzertflügel.
Auch hierdurch sehr hohe Komprimierungsrate,

Welte-Mignon

1904 schufen Welte und Sohn Berthold das **Reproduktionsklavier** Welte-Mignon (franz. mignon allerliebst, niedlich, reizend).

Es überträgt das Spiel eines Pianisten auf ein Lochband, die **Notenrolle**.

Dabei werden (fast) alle **Feinheiten der Interpretation** (Tempo, Anschlagstärke, Pausen usw., also die gesamte Agogik festgehalten (grie. agon Wettkampf, Anstrengung, etwa frei, individuell gestaltet.)

Heute stehen so Notenrollen der **Interpretationen vieler Musiker in Originalqualität** zur Verfügung.

Für die Ausbildung usw. hat es einen weiteren großen Vorteil

Bei der Wiedergabe sind so erstmalig, **Interpretationsmerkmale** (Lautstärke, Agogik, Betonung usw.) **ein- bzw. ausschaltbar.**

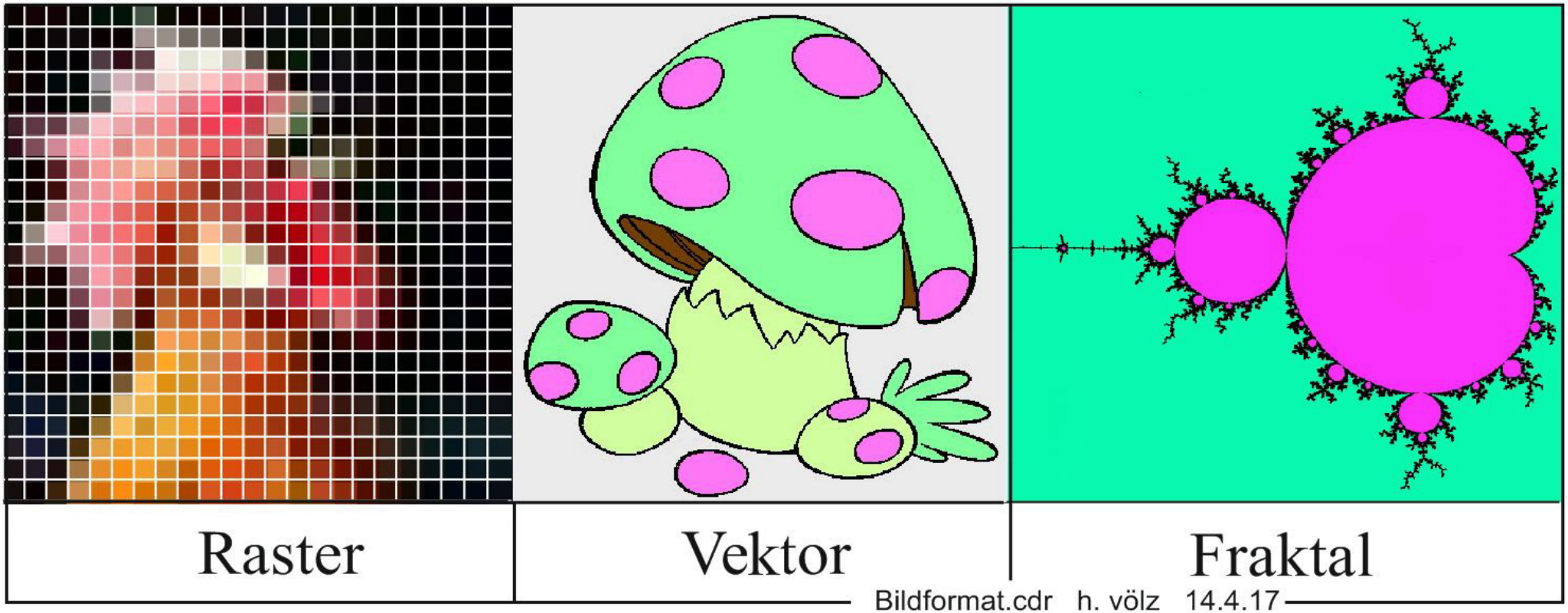
Selbst ein **Komponist** kann in seinen Noten **nicht so viele Details** festlegen

Ab etwa 1970 ist auch der automatisierte Übergang **zum MIDI** gelungen (Details [Völ05], S. 189ff.+592ff.). Leider ist ähnliches für Oper, Lied usw. nicht vorhanden.

Erwähnt sei noch, dass nach diesen Prinzipien auch bereits eine **Notenerkennung** von Musik leidlich funktioniert.

Bildkomprimierungen

Im Gegensatz zum Schall sind bei Bildern immer betont als zweidimensionale Signale vorhanden.
Z. T. werden aber auch dreidimensionale Gebilde (für Konstruktionen) dargestellt.
In jedem Fall sind komplizierte Behandlungen und Komprimierungen erforderlich.
Außerdem gibt es kein einheitliches Format.
Generell werden die frei Formate des Bildes unterschieden



Rasterbilder

Hier ist eine t-kontinuierliche 1D-Bearbeitung bei der früheren diskreten Zeilenabtastung möglich.
Mit den heute üblichen CCD-/CMOS-Sensoren entsteht immer eine 2D-Rasterung.
Meist werden 2D-verknüpfte Rasterbilder mit quadratischen Pixeln erzeugt.

Das vorige Bild zeigt vereinfacht das 20×20 Rasterbild eines Hahnenkopfes
Jedes einzelne Pixel trägt dann Werte für Helligkeit und Farbe, eventuell auch Durchsichtigkeit.
Sie werden mit diskret, mit Farbauflösungen von meist 24, auch 2, 8, 16 oder 32 Bit. gespeichert
Hochaufgelöste Rasterbilder enthalten viele Millionen Pixel.

Erst wenn die Pixel nicht mehr erkannt werden scheint wieder das kontinuierliche Bild vorzuliegen
Viel benutzt werden bmp, gif, ico, img, jpeg, jpeg2000, mac, pcx, pic, pmg. raw (unkomprimiert) und tif.
[Hol94] und [Str02].

Vektorbilder

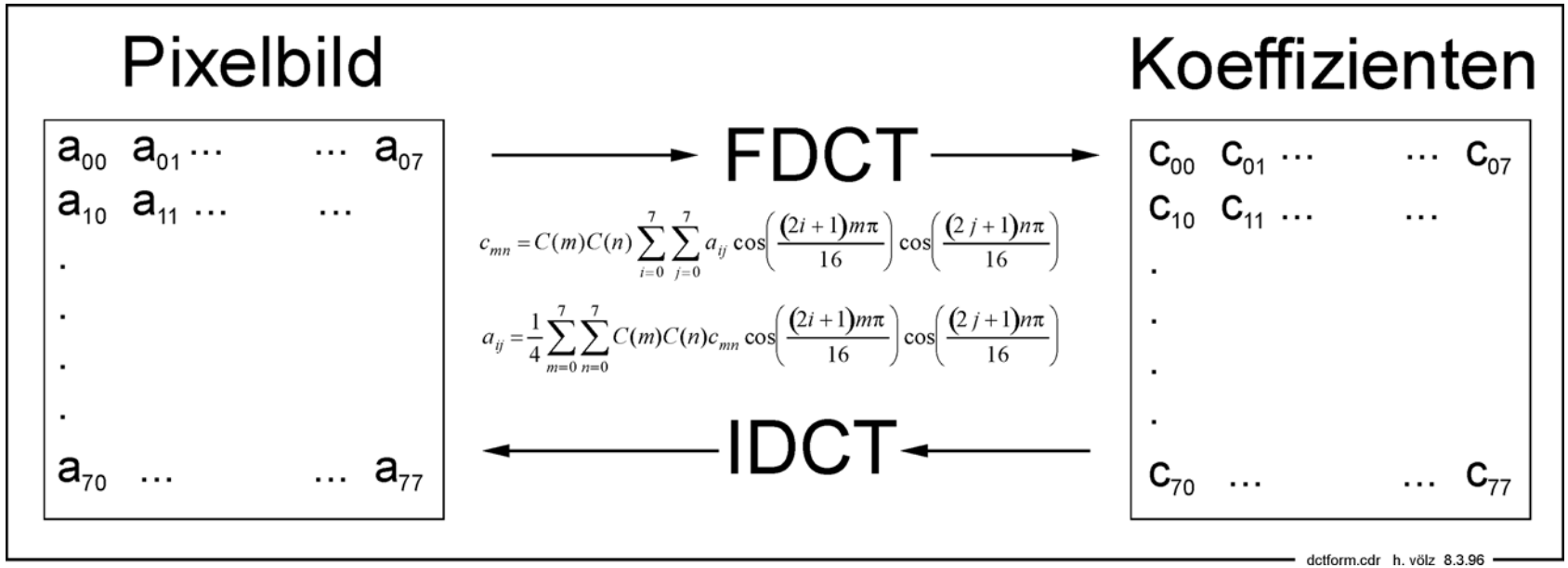
Vektor-Bilder entstehen bei einer erweiterter Euklidischen Geometrie mittels spezieller Programme.
Primär entstehen sie aus mathematisch generierten Linien mit wählbarer Strichstärke und -farbe.
Dabei werden auch Figuren wie Kreise Ellipsen, Vielecke usw. benutzt.
Abgeschlossene Flächen können dann mit (teilweise gleitender) Farbe ausgefüllt werden.
Solange keine zu komplexen Bilder notwendig sind, liegt dabei eine hoch verdichtete Datei vor.
Das Bild muss aber immer erst aus den digitalen Daten erzeugt werden.
Z.T. lassen sich Vektordateien für **räumliche** 3D-Darstellungen erzeugen
Sie ermöglichen dann Darstellungen aus verschiedenen Richtungen und erlauben Drehungen.
Prinzipiell lassen sich diese (Zahlen-) Daten noch zusätzlich digital mit obigen Methoden komprimieren.
Doch der Gewinn bleibt dabei gering.
Für das 3D-Vektor-Fomat gibt es nur dxf.
Für 2D-Vectorbilder existieren ai, cdr, cgm, dwg, dwf, emf, eps, gem, ps und wmf.

Fraktale

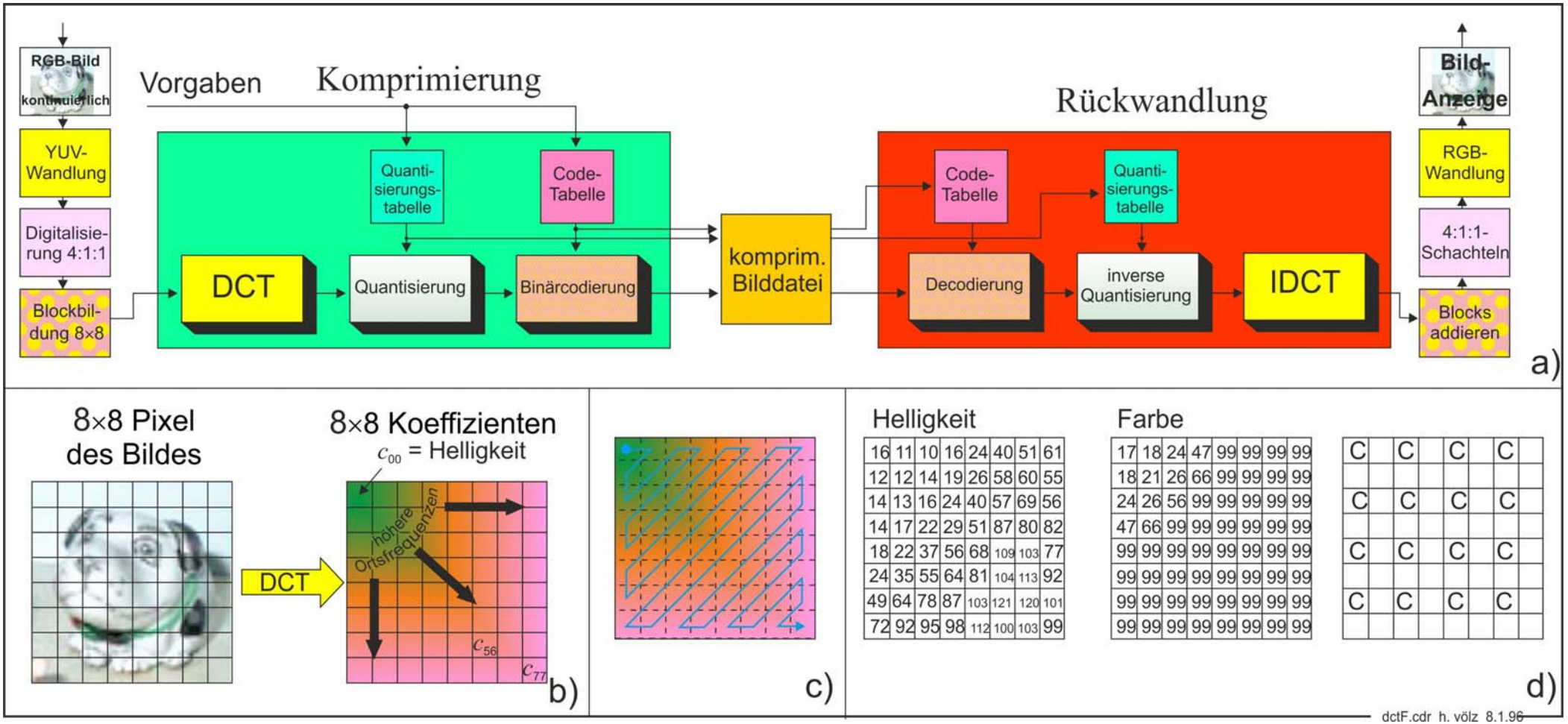
Sie können nur mittels zigfacher Rekursion erzeugt werden und verlangen leistungsfähige Rechentechnik
Siehe daher V-Infomation

Verlustbehaftete Bildkompression

Für ihre Entwicklung sie entstand 1986 das internationale Gremium JPG (joint photo group).
1991schuf sie das ziemlich komplizierte, aber recht leistungsfähige jpg-Format (Schema folgendes Bild)
Das rechteckigen Pixelbild wird zunächst in das YUV-Modell mit 4:1:1-Verdichtung überführt.
Dann werden Teilblöcke aus jeweils 8×8 Pixel gebildet.
Mit DCT (discrete cosinus transformation) wird die 8×8 Koeffizientenmatrix getrennt für Helligkeit und Farbe gebildet. Sie enthalten oben links den Wert c_{00} für die mittlere Helligkeit bzw. Farbe.
Weitere Koeffizienten erfassen deren Ortsfrequenzen.
Die Matrizen werden mit Tabellenwerten (d) multipliziert. Dann wird sequentiell abgetastet(c).
Sobald die Werte eine wählbare untere Grenze (bestimmt Qualität des komprimierten Bildes) unterschreiten, wird der Vorgang beendet und die komprimierte Bilddatei kann übertragen werden.
Die Rückwandlung in ein Bild erfolgt reziprok a).

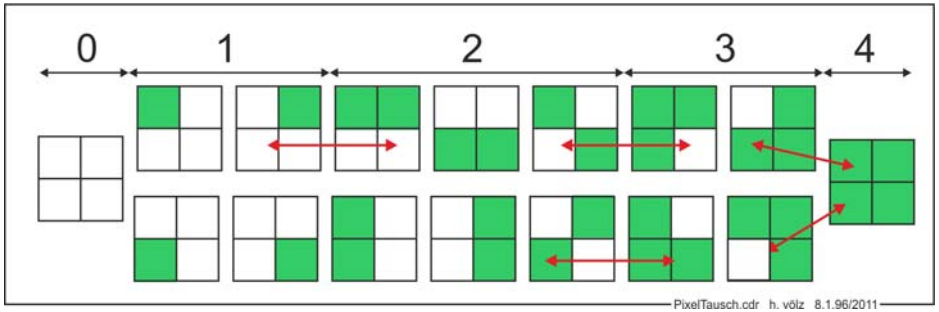


Jpg-Ablauf

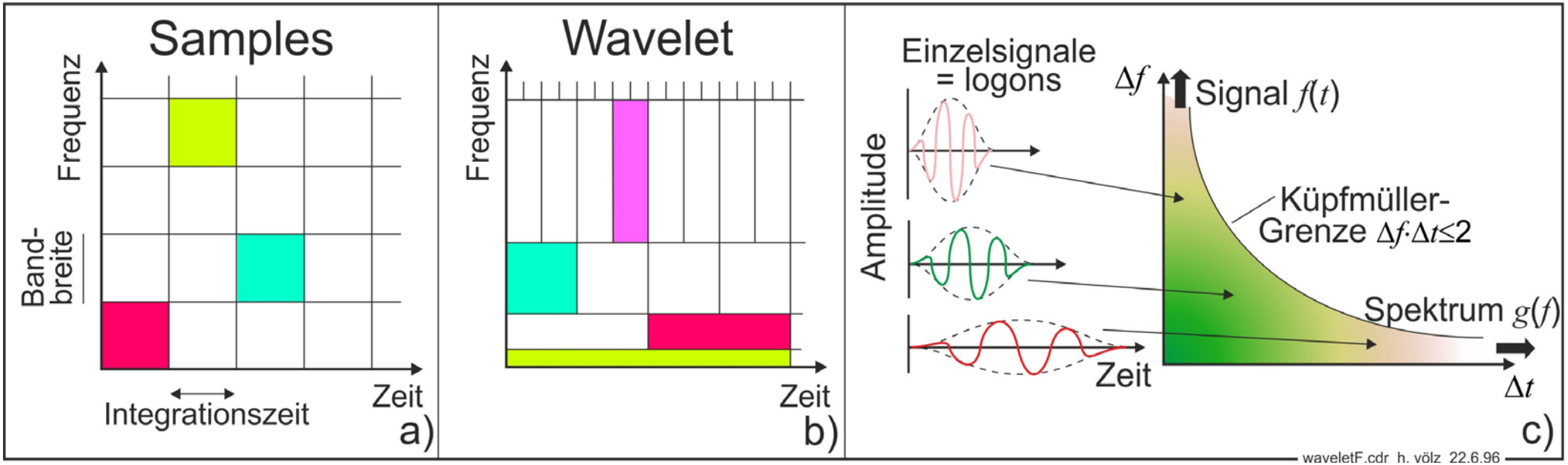

RGB-Bild
kontinuierlich
YUV-Wandlung
Digitalisierung 4:1:1
Blockbildung 8×8
Vorgaben
Komprimierung
Quantisierungstabelle
Code-Tabelle
DCT
Quantisierung
Binärcodierung
komprim. Bilddatei
Rückwandlung
Code-Tabelle
Quantisierungstabelle
Decodierung
inverse Quantisierung
IDCT
Bild-Anzeige
RGB-Wandlung
4:1:1-Schachteln
Blocks addieren
a)
8×8 Pixel des Bildes
8×8 Koeffizienten
c_{00} = Helligkeit
höhere Ortsfrequenzen
DCT
c_{56}
c_{77}
b)
c)
Helligkeit
16 11 10 16 24 40 51 61
12 12 14 19 26 58 60 55
14 13 16 24 40 57 69 56
14 17 22 29 51 87 80 82
18 22 37 56 68 109 103 77
24 35 55 64 81 104 113 92
49 64 78 87 103 121 120 101
72 92 95 98 112 100 103 99
Farbe
17 18 24 47 99 99 99 99
18 21 26 66 99 99 99 99
24 26 56 99 99 99 99 99
47 66 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99
C C C C
C C C C
C C C C
C C C C
d)
dctF.cdr h. völz 8.1.96

Ergänzungen

Eine zusätzliche Komprimierung geschieht zuweilen noch mit den 8×8 = 64 möglichen Matrixmustern
Einige (häufige)werden ausgewählt und durch ähnliche, aber selten auftretende Muster ersetzt.
Auf eine 4×4-Matrix reduziert zeigt es das Bild rechts.
Der Austausch erzeugt typische, störenden jpg-Artefarkte
(lat. ars Kunst, Geschick; factum das Gemachte).



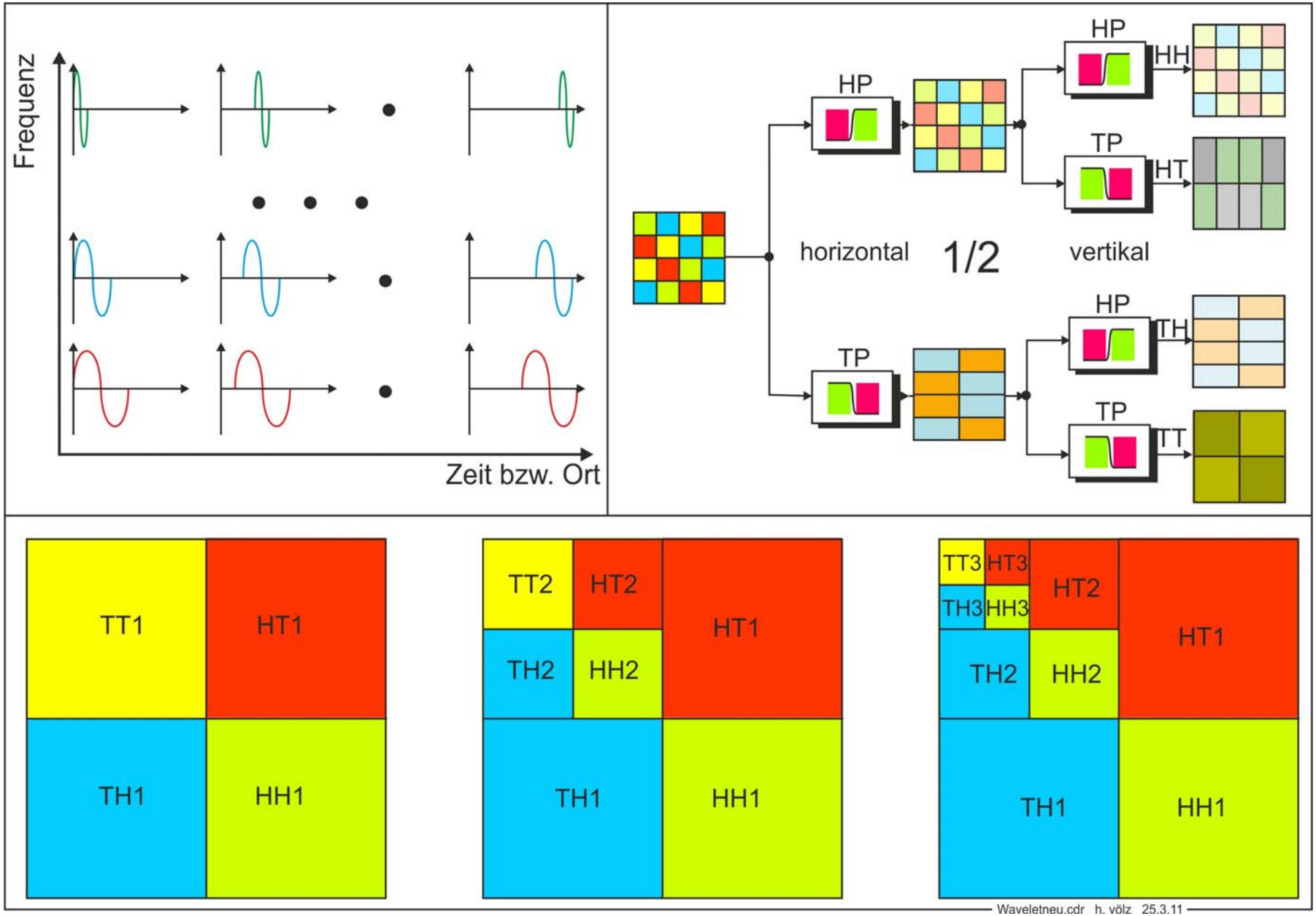
Um ihr Auftreten zu verringern wurden auch andere Transformationen versucht.
Möglichkeiten leiten sich vor allem aus dem Verhältnis von der Bandbreite Δf und Abtastzeit Δt ab



Wavelets

Sie benutzen eine einzelne (kurze) Schwingung (Wellchen), die immer noch nicht genormt ist.
Bisher wurde nämlich keine bestmögliche Variante gefunden (nächstes Bild).
Je nach ihrer Lage in der Zeitachse und ihrer Frequenz erfolgt die Wavelet-Transformation. Es entstehen hierarchisch gestaffelte Hoch- und Tiefpassfilterungen der Pixel-Teilmatrizen (s.u.).
Ähnlich wie bei der Fourier-Transformation werden Koeffizienten für die zu übertragende Datei berechnet
Hiermit kann ein besseres Originalbild erzeugt werden. Angewendet beim selten benutzten JPEG 2000.
Neben weiteren Varianten gibt es ein verlustloses jpeg-ls. Details zu jpg usw. u. a. in [Str02], S. 163, 221.

Es gibt noch viele weitere Bildformate, darunter auch Stereobilder, Hologramme und Fraktale. Für sie ist jedoch primär die Speicherung wichtig.


Frequenz
Zeit bzw. Ort
HP
TP
HH
HT
TH
TT
horizontal
1/2
vertikal
TT1
HT1
TH1
HH1
TT2
HT2
TH2
HH2
TT3
HT3
TH3
HH3
Waveletneu.cdr h. völz 25.3.11

Videoformate

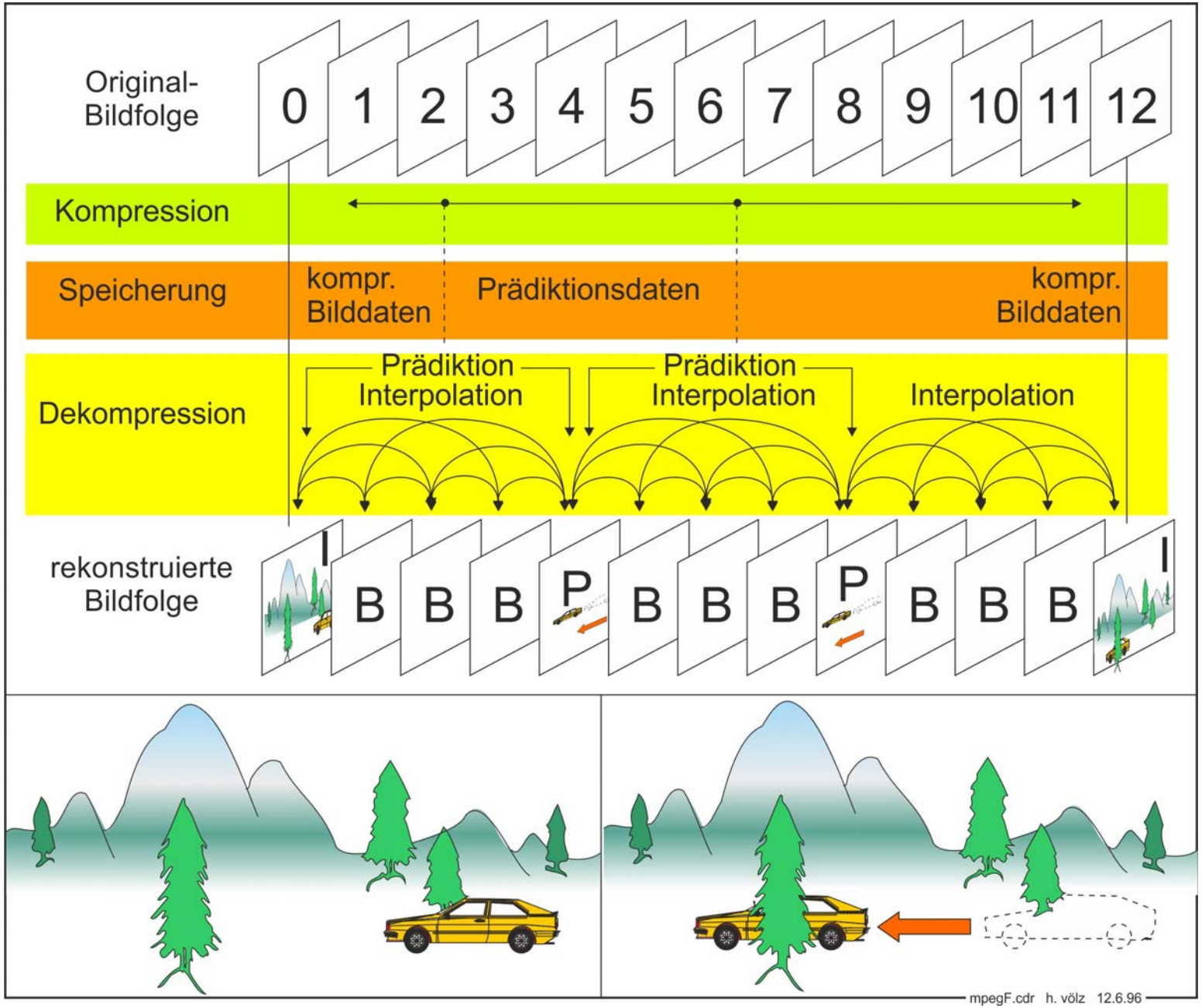
Für Videoformate ist jpg usw. ungeeignet: von Bild zu Bild ergeben such stark verändernde Artefarkte
Sie führen zu erheblich störenden Pixelrauschen. So entstand das mpeg-Format (moving peg).
Im folgenden Bild ist es vereinfacht dargestellt.
Eine aufeinander folgende Gruppe von meist 12 Bildern wird nach 3 unterschiedlichen Methoden bearbeitet.

1. Nur jedes zwölfte Bild wird als ***I*** (intra-frame) vollständig z. B. per jpg codiert.
2. dazwischen gibt es zwei Prädiktionsbilder ***P***. Über mehrere Zeilengruppen werden die sich bewegenden Bildteile abgeleitet, z. B. das durch den roten Pfeil gekennzeichnete Auto.
3. Mit den so gewonnenen Daten werden die 8 restlichen Zwischenbilder ***B*** mittels Interpolation berechnet

Diese Struktur besitzt erhebliche Nachteile für das Cuttern.

Erst bei den rein digitalen Videoformaten kommen neue leistungsfähige Möglichkeiten hinzu.
Dabei wurde schrittweise die Pixelzahl von 352×480 bei mpeg-1 auf 1 920×1 080 bei mpeg-3 erhöht.
Mit mpeg-4 wurde zusätzlich eine Version für interaktive Anwendungen mit 174×144 Pixel entwickelt.
Die Komprimierungsraten reichen dabei von etwa 80 : 1 bis 180 : 1.

viele Details zur Videocodierung enthält [Str02] ab S. 188. im Buch sind mehrere Quellcodes abgedruckt.


Original-
Bildfolge
0 1 2 3 4 5 6 7 8 9 10 11 12
Kompression
Speicherung
kompr.
Bilddaten
Prädiktionsdaten
kompr.
Bilddaten
Dekompression
Prädiktion
Interpolation
Prädiktion
Interpolation
Interpolation
rekonstruierte
Bildfolge
I B B B P B B B P B B B I
mpegF.cdr h. völz 12.6.96

Über einen Grafikcode

Für Sprache und Musik gibt es die sehr effektiven Komprimierungen mittels ASCII und MIDI.
Ähnliche Elementargebilde, aus denen sich alle möglichen Bilder erzeugen lassen, sind leider nicht bekannt.
Die analytische Geometrie ermöglicht Vieles mit Formeln.
Das Vektor-Format bietet zusätzliche Möglichkeiten
Jedoch die Vielfalt ist bereits deutlich größer als die Kenngrößen des ASCII- oder MIDI-Codes.
Daher sind für fotografische Bilder keine ähnlich großen Komprimierungsrate zu erreichen.
Eine Ursache könnte sein, dass die Realität immer 3-dimensional statt der 2-dimensionalen Abbildungen ist.
Aber auch für das 3-dimensionalen sind keine entsprechenden Elementareigenschaften bekannt
Das zeigen schon die Probleme für Stereo-Bilder und Holografie (s. d.) und das einzige 3D-Vektor-Format.
Es gibt aber noch die Fraktale, doch darauf kann erst bei der V-Information eingegangen werden.
Deshalb sei eine insgesamt ergänzende Analyse der Fakten für eine Komprimierung bzgl. 3 Fakten versucht.

- Für die Erzeugung, Herstellung der einzelnen Produkte ist immer ein geistiger, technischer, materieller und zeitlicher ***Herstellungs-Aufwand*** erforderlich.
- Es ist eine jeweils typische ***Speicherkapazität*** notwendig, um die geschaffenen Produkte aufzubewahren und verfügbar zu halten.
- Die Rezeption der Produkte erfolgt über unsere Sinne (vorrangig sehen und hören). Das geschieht immer nur für eine gewisse Zeitdauer. Danach erlischt deutlich das Interesse. So ergibt sich ein typischer mittlerer ***Nutzungs-Aufwand***

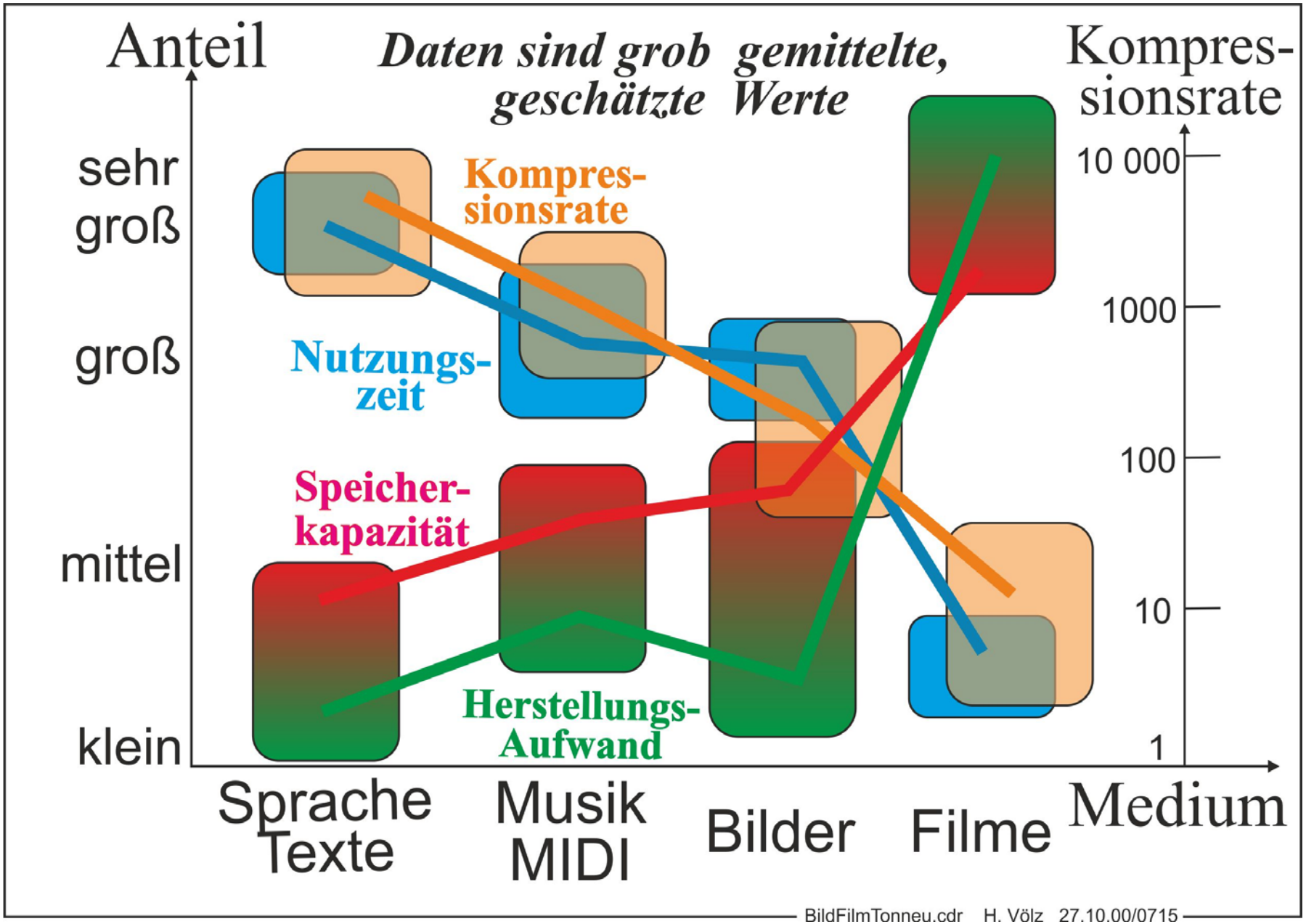
Zur Abschätzung

Für die Rezeption sind unsere physiologischen Eigenschaften entscheidend.
Das sind die Wahrnehmungsraten unserer Sinne und die Gedächtniskapazität.
Wenn wir alles vom jeweiligen Produkt hinreichend erkannt und gespeichert haben, erlischt eben unser Interesse.
Die Werte hängen erheblich von individuellen Interessen ab und ändern sich oft deutlich mit dem Alter.
Ein erster Überblick ergibt sich aus zwei Parametern:

1. Die Dauer in der das Produkt interessant ist und daher vollständig gehört und/oder betrachtet wird.
2. Wie oft und lange das wiederholt wird. Hierfür sind drei Aussagen recht gut gesichert:

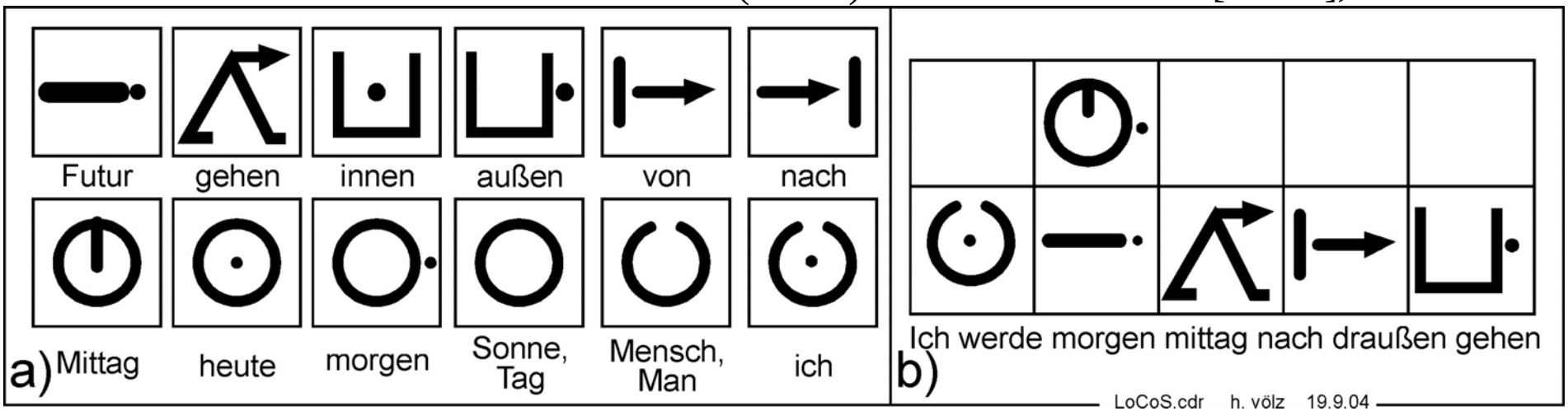
- Viele **Musikstücke** hören wir immer wieder, nicht selten mehr als hundertmal.
- **Filme** sehen wir dagegen nur wenige Male an, in Sonderfällen um dreimal.
- **Bilder** betrachten und **Texte** lesen liegen etwa zwischen beiden Extremen.

Eigentlich sollten unsere ***technischen Methoden*** im Sinne einer kybernetischen Anpassung und Analogie „optimal“ bezüglich unserer Physiologie entwickelt und konstruiert sein.
Dann müssten die Verhältnisse zwischen den drei Größen für alle Produkte etwa gleich sein.
So ergibt sich das grob das Folgende.
für Bilder besitzen wir noch keine auf unsere Sinne gut angepasste Technologie.
Beachtlicher Fortschritt, deutliche Vereinfachung brachte teilweise die elektronische Fotografie erreicht.
Dennoch ist der ***Abstand zu Sprache und Musik*** beachtlich.
Vielleicht liegt das an einem **fehlenden Code**.
Noch größer und z. Z. unveränderlich gilt die ***große Abweichung für Filme/Videos***.
Hier müssten also in der Zukunft noch ***revolutionäre Entwicklungen*** erfolgen.


Anteil
Daten sind grob gemittelte, geschätzte Werte
Kompres-sionsrate
sehr groß
groß
mittel
klein
Kompres-sionsrate
Nutzungs-zeit
Speicher-kapazität
Herstellungs-Aufwand
10 000
1000
100
10
1
Sprache Texte
Musik MIDI
Bilder
Filme
Medium
BildFilmTonneu.cdr H. Völz 27.10.00/0715

LoCoS

(lovers communication system) System: Kommunikationssystem für Liebende)
Dieser interessante Ansatz stammt von Yukio Ota (*1939). Weitere Details u. a. [Völ05], S. 335.

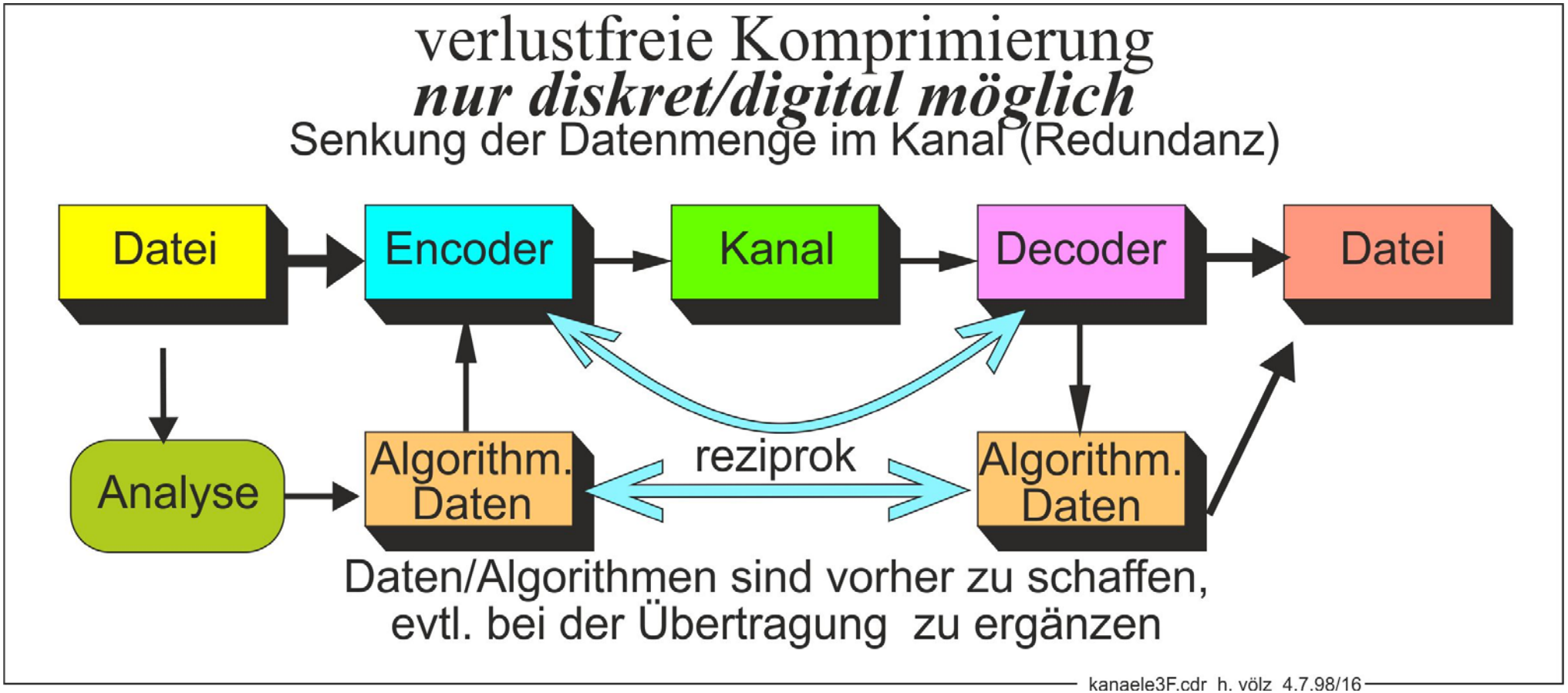


Diese Pseudosymbole (codes) sind in jeder Sprache schnell erlernbar. Ein Beispiel zeigt das Bild.
Falls trotz der genannten Probleme dennoch in absehbarer Zeit einer der gewünschten Codes gefunden werden sollte, ergäben sich bestimmt neue Probleme.
Er müsste von den Anwendern langfristig und gründlich gelernt und gut beherrscht werden.
Negative Bespiele zeigen heute bereits: Nur wenige beherrschen Musiknoten oder Kurzschrift (Steno).
Noch deutlicher zeigt das die Tanzschrift, die wohl kein Choreograf oder Ballettmeister beherrscht.
Stattdessen sind zum aufschreiben spezielle Notatoren erforderlich.
Ihre 2-bis 3-jährige Ausbildung erfolgt nur im Choreologischen Institut in London
Nur so kann die gemeinsam mit den Tänzern gefundene Choreografie aufgeschrieben werden.
Wegen der Spiegelbildlichkeit sind Videos wider Erwarten so gut wie nicht nutzbar. [Völ05], S. 210ff.

Verlustfreie Komprimierungen

Dem Encoder und Decoder werden gleiche Algorithmen und Daten zugeordnet.
Seit geraumer Zeit gibt es dabei für Daten und Komplexität der Algorithmen kaum Grenzen. Es muss gelten:

- Die Datenmenge muss endlich sein. Das gilt immer automatisch, denn selbst ein Stream hört mal auf.
- Die Algorithmen müssen sicher terminieren, denn sonst würde die Übertragung nie enden.
- Der Kanal und die Datei müssen muss diskret (digital) arbeiten.



Die zwei wichtigsten Verfahren

1. ***Algorithmische*** Codierung: Auf beiden Seiten befinden sich die zueinander reziproken Algorithmen.
   Der Encoder untersucht die zu übertragende Datei und findet darin Gesetzmäßigkeiten.
   Dafür wählt er den am besten geeigneten Algorithmus zur Komprimierung aus.
   Je nach Dateigröße und der Anzahl verfügbarer Algorithmen kann das einige Zeit benötigen.
   Damit sie beim Decoder entfällt, wird die Auswahl als Header zuerst mitgeteilt.
   Dann werden die zu übertragenden digitalen Daten mit dem Algorithmus komprimiert und übertragen.
   Nach dem Empfang erzeugt der Decoder mittels Header wieder die Originaldaten.
2. ***Link***-Codierung: Für beide Seiten wird festgelegter Datenvorrat benutzt,
   z. B. für Juristen Gesetzbücher, für Theologen die Bibel, Koran usw.
   Sie sind in Dateien, Paragrafen, Abschnitte usw. eingeteilt, durch Namen bzw. Zahlen gekennzeichnet
   Übertragen werden nur die Namen bzw. Zahlen
   Zumindest theoretisch ist so jede , Auswahl usw. mit nur 1 Bit zu übertragen.
   Im Prinzip sind so ***auch kontinuierliche Dateien verlustlos*** zu übertragen.

In der Praxis wird vorwiegend eine Kombination beider Varianten benutzt.
Dann ist immer mehr als 1 Bit erforderlich und die kontinuierlichen Daten entfallen.
Teilweise ermöglicht die algorithmische Codierung auch die Komprimierung unendlicher Reihen in endliche Daten (Programme).
Es gilt der Satz von Chaitin von 1975 [Cha75]:
Von allen unendlich langen Ketten lassen sich endlich viele verkürzen, aber unendlich viele nicht.
Insbesondere sind Zufallsfolgen nur selten zu komprimieren verkürzen.
Dieser Satz gilt aber nicht für endliche Dateien, die lassen sich immer irgendwie verkürzen.

Unendliche Reihen auf endlich verkürzen

Typische Beispiele sind:

33333…	⇒ beginnen mit 3 und Anhängen von 3 ständig wiederholen
01010101…	⇒ dasselbe für 01
01001000100001…	⇒ Startkette s_0 = 01, s_1 = 0 & s_0; dann fortwährend $s_{n+1} = s_n$ & s_0
1 4 2 5 0 3 6 1 4 7 2...	⇒ s_0 = 1; $s_{n+1} = (s_n + 3)$ Mod 8
2 3 5 7 11 13 17...	= Primzahlfolge ⇒ Primzahl-Algorithmus
π	⇒ Algorithmus (Formel) für π

Zwei Grundsätze

Es werden nur die wichtigsten, bedeutsamen oder häufig benutzten Verfahren beschrieben.

Zunächst sind dafür aber noch zwei Arten der algorithmischen Codierung zu unterscheiden.

1. Jede ***Quellen-Codierung*** geht von den Wahrscheinlichkeiten der Zeichen, Wörter usw. aus und erzeugt dafür den optimalen ***Präfixcode***.

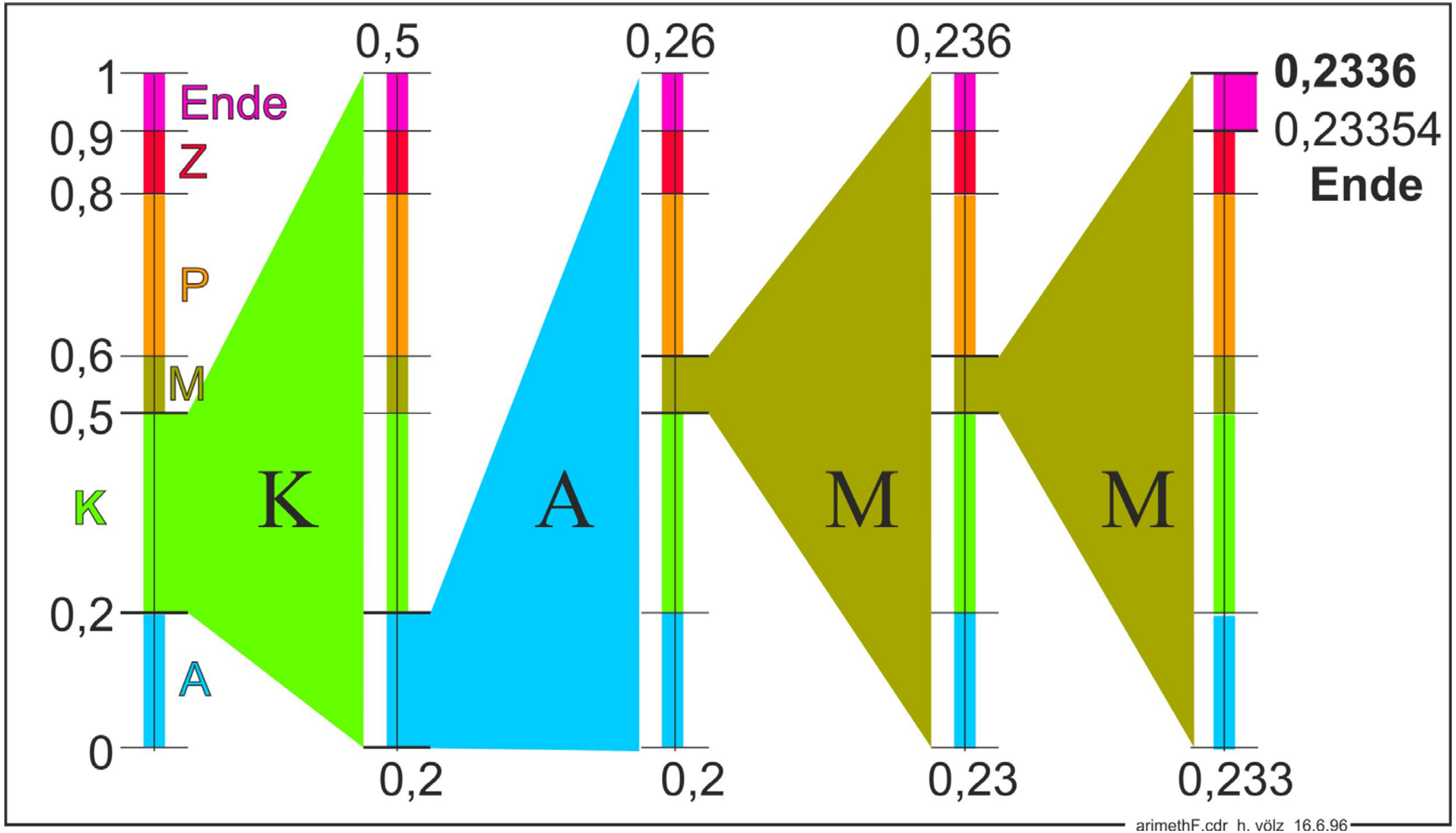
2. Die ***Kanal-Codierung*** betrifft dagegen immer einzelne, fest gegebene Dateibestände
   Wenn darin die Zeichen oder begrenzte Abschnitte bestimmbare Häufigkeiten besitzen,
   dann können auch die Verfahren der Quellen-Codierung gut für die Kanal-Codierung sein.
   Deshalb wird z. B. manchmal die Huffmancodierung als Abschluss-Codierung bei jpg angewendet.

Nur für die Kanal-Codierung folgen jetzt Beispiele.

Arithmetische Codierung

Sie wurde ab 1975 von P. Elias bei IBM entwickelt und dann patentiert [Wit87].
Z. T. brigt sie bessere Ergebnisse als der Huffman-Code, wegen der Lizenzkosten nur selten eingesetzt.
Sie arbeitet mit fortlaufenden, durch Häufigkeiten festgelegten Intervall-Schachtelungen.
Das Ergebnis ist schließlich nur eine Zahl als Codierung. Leider codiert sie deutlich langsamer.
Ein stark vereinfachtes Beispiel in dezimaler (üblich ist binär) Schreibweise benutzt das Wort KAMM.
Die einzelnen Zeichen besitzen die Häufigkeiten der Tabelle.
Den Ablauf der Codierung zeigt das Bild nächste Seite.
Nach jedem Codierschritt ist das weiter zu benutzende Intervall verkleinert (Tabelle und Bild).
Dabei wird oft eine endgültige Ziffer festgelegt (rot gekennzeichnet). Sie wird sofort gespeichert, gesendet.
Das Ende der Codierung erfolgt durch ein Sonderzeichen „!“

Zeichen	Häufigkeit	1. Intervall	Text-folge	Folge-Intervall	gültige Ziffern
A	0,2	0,0 - 0,2	K	0,2 - 0,5	0
K	0,3	0,2 - 0,5	KA	0,20 - 0,26	0,2
M	0,1	0,5 - 0,6	KAM	0,23 - 0,236	0,23
P	0,2	0,6 - 0,8	KAMM	0,233 - 0,2336	0,233
Z	0,1	0,8 - 0,9	Ende	0,23354 - 0,2336	0,2335
!	0,1	0,9 - 1,0	Ausgabe	**0,23358**	0,23358


0,5
0,26
0,236
0,2336
0,23354
Ende
1
0,9
0,8
0,6
0,5
0,2
0
Ende
Z
P
M
K
A
K
A
M
M
0,2
0,2
0,23
0,233
arimethF.cdr h. völz 16.6.96

Lauflängen-Codierung

(**RLE** Run Length Encoding) wird u. a. bei Bildern (z. B. pcx) für sich wiederholende Pixelwerte benutzt.
In der komprimierten Datei folgen dabei nacheinander je ein Zähl- und ein Pixel-Byte.
Statt „CCCCCCAABBBBAAAAEE“ wird dann 6C2A4B4A2E übertragen.
Bei geringen Wiederholungen kann so die Datei sogar größer werden, dann wird die Original-Datei benutzt.

Pointer-Verfahren

Es ist teilweise bei ZIP und ARC implementiert.
Es benutzt Verweise auf Orte, wo die aktuelle Zeichenfolge bereits vorher aufgetreten, gesendet wurde.
Der Verweis benötigt 2 Byte, je eines für den Ort und die Länge der Zeichenkette. Bei der Beispielkette

ABRABRIKADABRA

wiederholt sich ab dem vierten Buchstaben ABR. Stattdessen genügt dafür der Verweis 1, 3.
Unter Beachtung des nochmaligen ABR folgt somit für die gesamte Komprimierung

ABR 1, 3 IKAD 1, 3 A.

Die Effektivität des Verfahrens sinkt, wenn die Verweisvektoren zu groß werden.
Daher wird meist mit gleitendem Fenster gearbeitet.
Auch begrenzte Blocklängen sind vorteilhaft.

Code-Erweiterung

Wurde 1977 von J. ZIV; A. LEMPEL (LZW, PKZIP) eingeführt [Ziv77], 1984 von Welch erweitert [Wel84].
Dabei wird mit den 256 Zeichen des 8-Bit-ASCII-Codes begonnen.
Für häufige Zeichen-Kombinationen werden neue Symbole als 9- oder gar 10-Bit-Zeichen eingeführt.
Für die optimale Codierung müsste die Datei erst vollständig analysiert werden. Das dauert oft zu lange.
Stattdessen werden je Zeichen drei Vorgänge benutzt:

1. ***Ausgabe des aktuellen*** Symbols/Codes in die komprimierte Datei,
2. ***Erweiterung des Symbolsatzes*** durch eine neues Kombinationszeichen (z. T. auch Reduzierung)
3. **Verwendung** des erweiterten Symbolsatzes.

Als Beispiel sei verwendet

wieder□diese□Kinder□.

Dabei steht „□“ für das Leerzeichen. Zunächst wird w ausgegeben und wi = 256 generiert.
Dann folgt i und ie = 257 wird generiert; Weiter folgt die Ausgabe von e und neu ed = 258 usw.
Nach diesen Schritten existiert bereits ie und es wird als 257 ausgegeben. Zusätzlich ies = 264 erzeugt (alle niedrigeren sind inzwischen vergeben). Am Ende der Zeichenkette existieren die neuen Codes:

256 (wi); 257 (ie); 258 (ed); 259 (de); 260 (er); 261 (r□); 262 (□d); 263 (di); 264 (ies);
265 (se); 266 (e□); 267 (□K); 268 (Ki); 269 (in); 270 (nd); 271 (der).

Ausgegeben wird so die von 20 auf 16 Zeichen verkürzte Kette

wieder□d 257 se□Kin 259 261

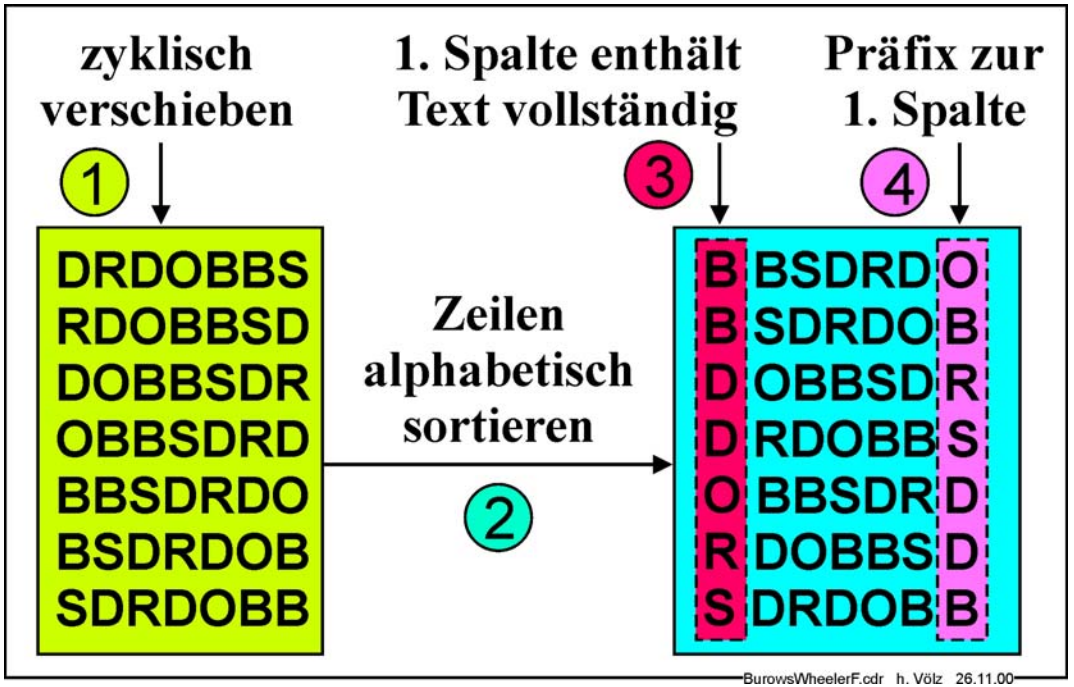
Dafür ist die Code-Länge jedoch von 8 auf 9 Bit/Zeichen angestiegen. Zusätzliche Maßnahmen ermöglichen es, die erweiterte Codebasis (bis 512 = 9 Bit) möglichst vollständig zu nutzen.

Burrows-Wheeler-Transformation

BWT wenig bekannt: 1983 nur intern von M. Burrows und D. J. Wheeler als ungewöhnliche Transformation publiziert [Bur83].
1994 ist sie dann als B2Zip an einem einfachen Beispiel gut verständlich vorgestellt worden [Kru92].
Dabei wird der kurze Text DRDOBBS benutzt. Ihre Wirkungsweise läuft damit in den folgenden Stufen ab:

- Zunächst wird der Text ***in einer Matrix zyklisch verschoben*** angeordnet ①.
- Dann werden die Zeilen der Matrix **alphabetisch sortiert** ② und erzeugen so die neue Matrix (rechts).
- Erste Spalte ③ enthält vollständigen Text, ist alphabetisch sortiert. gut per Lauflänge komprimierbar.
- Die letzte Spalte ④ enthält den jeweiligen Präfixbuchstaben zur ersten Spalte.
- Ihre Abfolge wird zusätzlich mittels Zahlen übertragen und dienen später zum Dekomprimieren.

Infolge der hohen Komplexität ermöglicht B2Zip (auch BZ2, BZ/P2) bis zu zehnfache Kompressionsrate.



Hilberg-Codierung

Sie existiert nur für Texte, ist extrem leistungsfähig, nutzt die Syntax, z. T sogar die Grammatik aus.
Auf jedes Wort kann nur eine Auswahl aus wenigen Wörtern folgen.
Der Zusammenhang muss aus umfangreichen Texten extrahiert und im En- und Decoder abgelegt werden.
(gespeicherte linke Tabelle). Jedes Wort erhält ***2 Eingangs-Vektoren*** x_i, y_i und 1 ***Ausgangsvektor*** z_i.
Mit nur 4 Bit (y_i, $3*x_i$) können so die 16 Sätze der rechten Tabelle generiert werden [Hil84, 90, 00]. Die Codierung für den ersten Satz 0000: „Rotkäppchen trifft die Großmutter“ ist deutlich rot hervorgehoben.

Mit dem Verfahren gelang es Meyer [Mey89] alle Dissertationen der Nachrichtentechnik Darmstadt mit nur 65 Bit vollständig zu codieren. Das entspricht einer Entropie ≈0,012 Bit/Buchstabe bzw. ≈1,8 Bit/Textseite. Diese Aussagen wurden vielfach bezweifelt, z. T wurde sogar Fälschung, Scharlatanerie oder gar Betrug angenommen. Jedoch lassen sich mit **65 Bi**t $2^{65} \approx 3{,}7 \cdot 10^{19}$ unterschiedliche Arbeiten generieren.

Meine Versuche mit Zufalls-Bit-Kombinationen nicht vorhandener Arbeiten zeigten immer einen „höheren Unsinn“.
Sie besaßen stets eine korrekte Syntax und Grammatik.
Für die Kapazität des Datenspeichers waren beim Decoder jedoch viele MByte (als Tabelle) erforderlich.
Im Prinzip sind auf dieser Basis viele neue Ansätze möglich.
Hilbert hat Ansätze zu einer fast idealen Übersetzung zwischen Sprachen erprobt.
Außerdem glaubt er, so eventuell eine Art Gedanken-Code schaffen zu können.
Bei mir hat auf dieser Grundlage ein Diplomand eine hocheffektive Eingabe für Behinderte programmiert [Rie13].

Gespeicherte Daten für Hilberg-Codierung

$y_i x_i$	Teil 1 oder 2	
0 0	Rotkäppchen	1
0 1	der Wolf	1
1 0	trifft	0
1 1	erkennt	0

$y_i x_i$	*Teil* für Satzende
0 0	den Jäger
0 1	die Großmutter
1 0	den Hänsel
1 1	die Gretel

$y_1 x_1 x_2 x_3$	mögliche Sätze
0 0 0 0	Rotkäppchen trifft den Jäger
0 0 0 1	Rotkäppchen trifft die Großmutter
0 0 1 0	Rotkäppchen erkennt den Jäger
0 0 1 1	Rotkäppchen erkennt die Großmutter
0 1 0 0	der Wolf trifft den Jäger
0 1 0 1	der Wolf trifft die Großmutter
0 1 1 0	der Wolf erkennt den Jäger
0 1 1 1	der Wolf erkennt die Großmutter
1 0 0 0	trifft Rotkäppchen den Hänsel
1 0 0 1	trifft Rotkäppchen die Gretel
1 0 1 0	trifft der Wolf den Hänsel
1 0 1 1	trifft der Wolf die Gretel
1 1 0 0	erkennt Rotkäppchen den Hänsel
1 1 0 1	erkennt Rotkäppchen die Gretel
1 1 1 0	erkennt der Wolf den Hänsel
1 1 1 1	erkennt der Wolf die Gretel

Kolmogorow-Komplexität

Komprimierungen stehen im engen Zusammenhang mit dem Satz von Chatain und der ***Kolmogorow-Komplexität***. Sie ist außerdem für die V-Information wichtig und verlangt

- eine universelle Turing-Maschine M
- ein Programm p
- eine auf dem Speicherband stehende Inschrift i

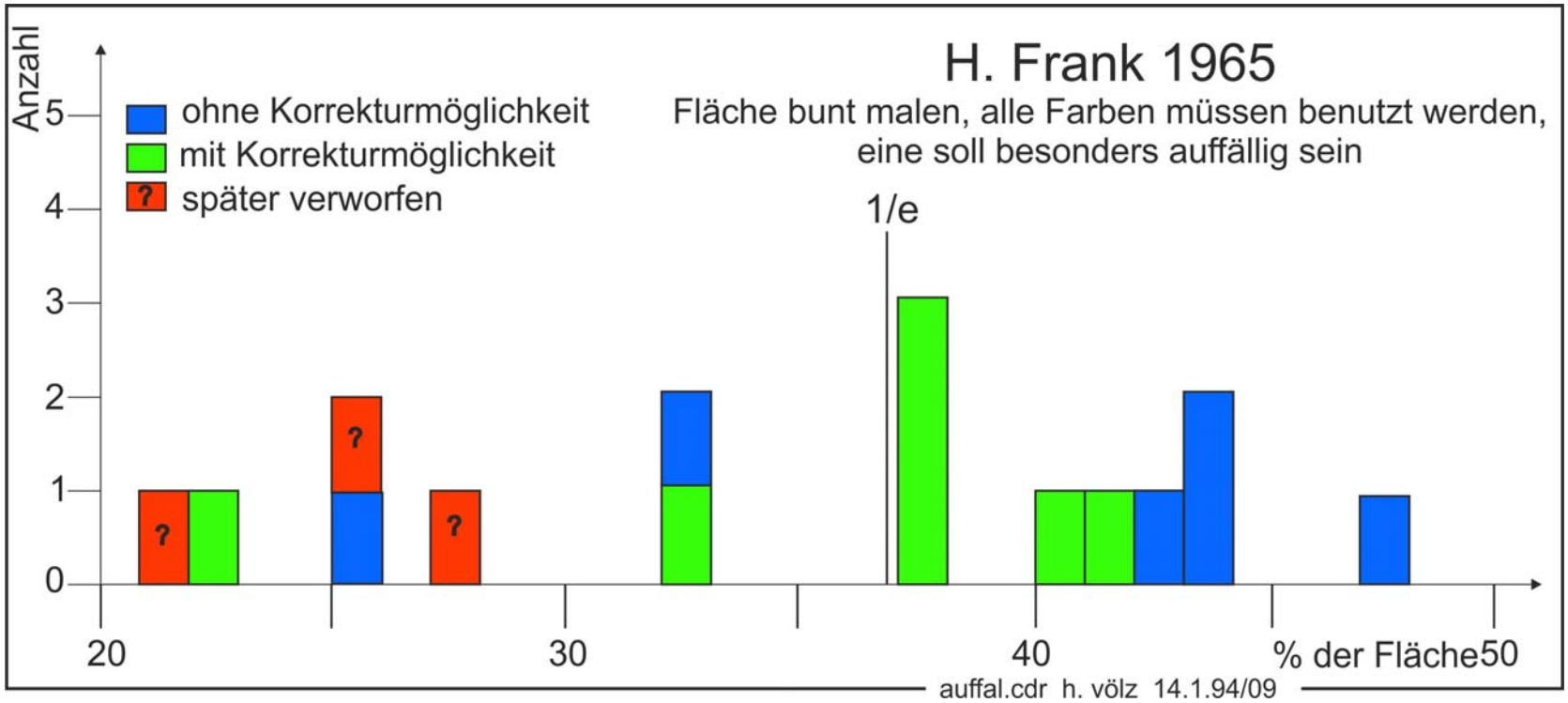
Sie erzeugen die Folge $s = M(p, i)$.
Es seien alle Programme p_i bekannt, die auf dem Speicherband s erzeugen.
Hierunter ist p_k das kürzeste (bekannte) Programm.
Seine Länge $L_M(s)$ ist die (aktuelle) Kolomogorow-Komplexität bezüglich s.
Wird auf einer anderen universellen Turing-Maschine U dann M simuliert, so gilt: $L_M(s) \leq L_U(s) + K_{M,\,U}$

Kurze Zeittafel zur Komprimierung

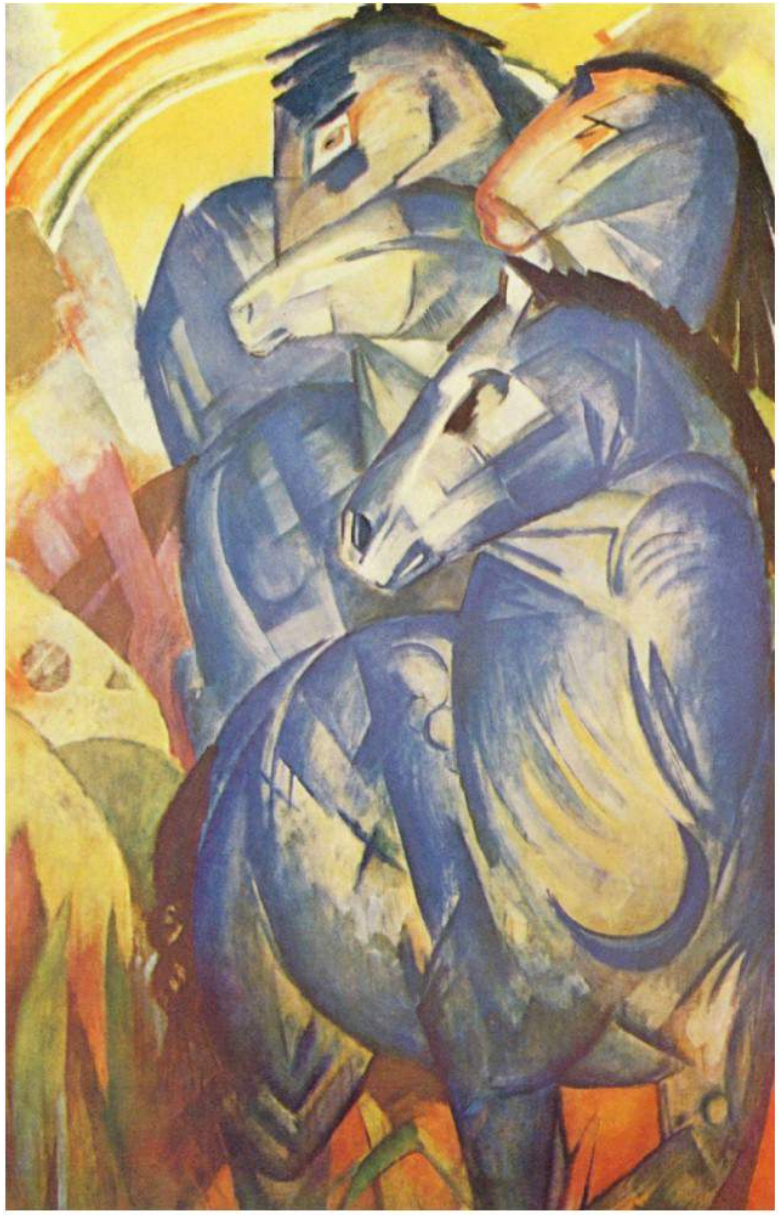
1949 Informationstheorie, Claude Shannon
1949 Shannon-Fano-Codierung
1952 Huffman-Codierung
1964 Konzept der Kolmogorow-Komplexität
1975 Integer Coding Scheme nach Elias
1975 Satz von Chatain
1977/78 Lempel-Ziv-Verfahren LZ77, LZ78
1979 Bereichs-Codierung als Implementierung der arithmetischen Codierung
1982 Lempel-Ziv-Storer-Szymanski (LZSS)
1983 Idee von Burrows-Wheeler
1984 Beginn der Hilberg-Codierung + Lempel-Ziv-Welch-Algorithmus (LZW)
1985 Apostolico, Fraenkel, Fibonacci Coding
1986 Move to front (Bentley et. al., Ryabko)
1989 Meyer speichert alle Darmstädter Dissertationen mit nur 65 Bit
1991 Reduced Offset Lempel Ziv (ROLZ, auch LZRW4, Lempel Ziv Ross Williams)
1994 Burrows-Wheeler-Transformation B2Zip (BZ2)
1996 Lempel-Ziv-Oberhumer-Algorithmus (LZO)
1998 Lempel-Ziv-Markow-Algorithmus (LZMA)

Anwendung außerhalb der Nachrichtentechnik

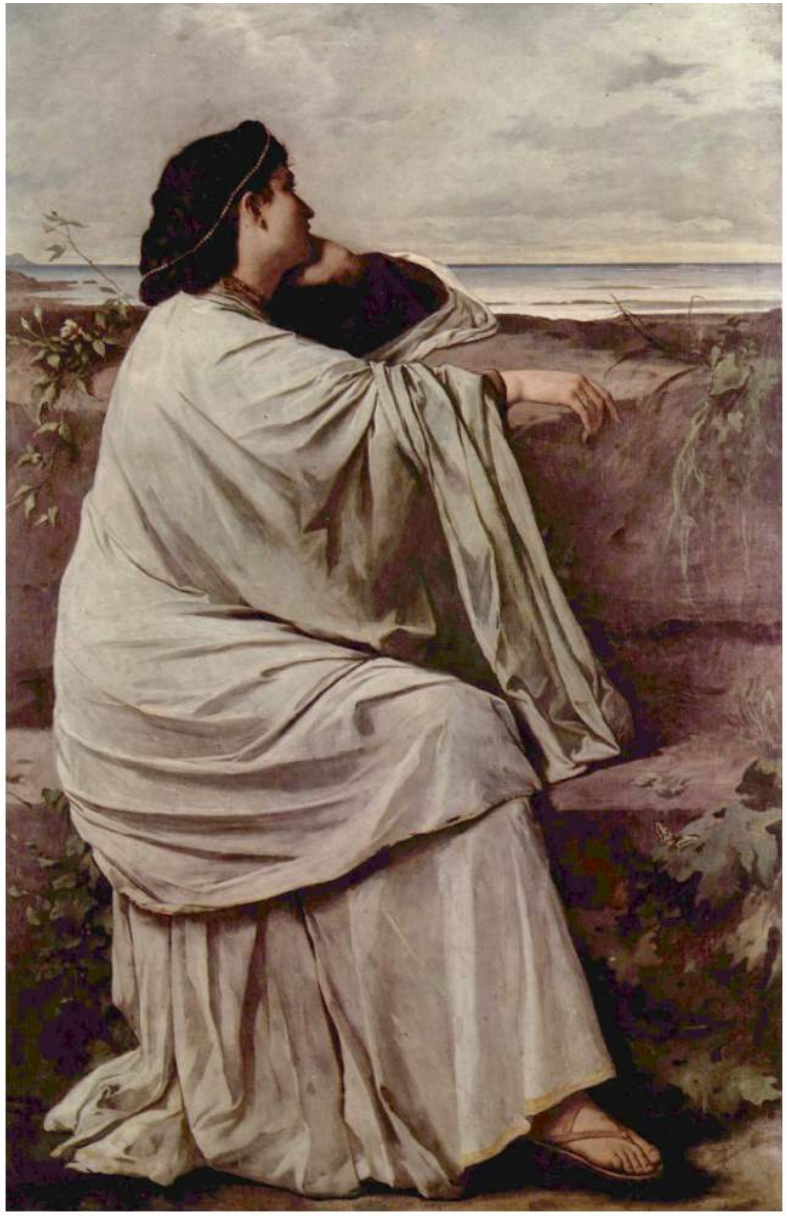
S-Information betrifft primär die bestmögliche technischen Übertragung.
Das Prinzip ist aber auch auf unsere Wahrnehmung anwendbar.
Damit hat sich u. a. H. Frank ausführlich beschäftigt.
Er führte um 1960 die **Auffälligkeit** des Maximums von $-p\mathrm{ld}(p)$ ein. Es liegt bei $1/e \approx 37\ \%$ [Fra69].
In einem Versuch mussten Probanden eine Fläche so mit 7 Farben füllen, dass eine auffällt.



Flächenanteile von Bildern genügen auch oft der Auffälligkeit.
z. B. der Turm der blauen Pferde von Marc 1913 und das Weiß bei der Iphigenie von Feuerbach (37 %)



Franz Marc: Turm der blauen Pferde 1913, 200* 130 cm



Amseln Feuerbach: Iphigenie II; 1871 200*132 cm

Bei Texten, Rhythmen und Kaufhaus

Weiter analysiert er eine **Gedicht**zeile aus „The Bells“ von Poe.
Gesprochen fällt das betonte „**e**“ deutlich auf. Es kommt darin achtmal vor, was grob 33 % entspricht.

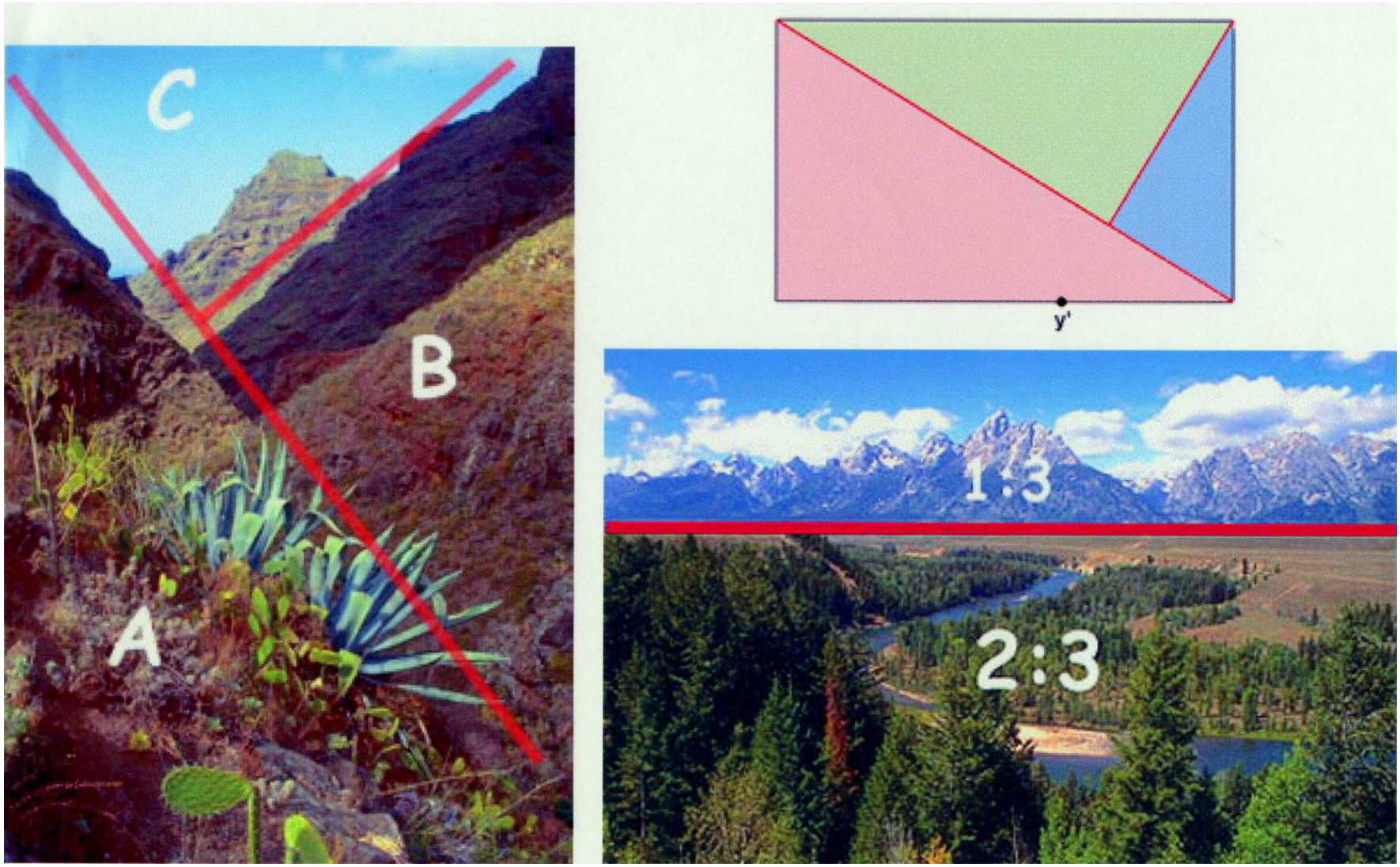
Hear the sl***e***dg***e***s with the b***e***lls, silver b***e***lls!
What a world of m***e***rrim***e***nt their m***e***lody foret***e***lls!

In der **Musik** fand er den Zusammenhang mit den Synkopen.
Im Jazz kommen sie so häufig vor, dass sie kaum Beachtung finden.
Jedoch im 3. Satz des 5. Brandenburgischen Konzerts von Bach sind sie mit 124 von 310 Takten (≈40 %) besonders auffällig.

Sehr ungewöhnlich ist eine Anwendung auf die Verkaufstrategie von Kaufhäusern.
Frank und Moles saßen eines Abends um 1973 mit dem Leiter eines großen Berliner Kaufhauses beim Bier und diskutierten dabei auch über die Auffälligkeit.
Dabei entstand die Idee, diesen Fakt auf die Preispolitik anzuwenden.
Zum nächsten Quartal wurden bei etwa ein Drittel der Produkte die Preise extrem knapp über ihren Einkaufswert festgelegt.
Für die anderen wurden sicherheitshalber etwas überhöhte Preise gewählt.
Bereits nach einem Monat war der Umsatz deutlich gestiegen. Eine repräsentative Befragung ergab:
Das Kaufhaus sei besonders preisgünstig. So wird das offensichtlich heute für die „Schnäppchen“ genutzt.

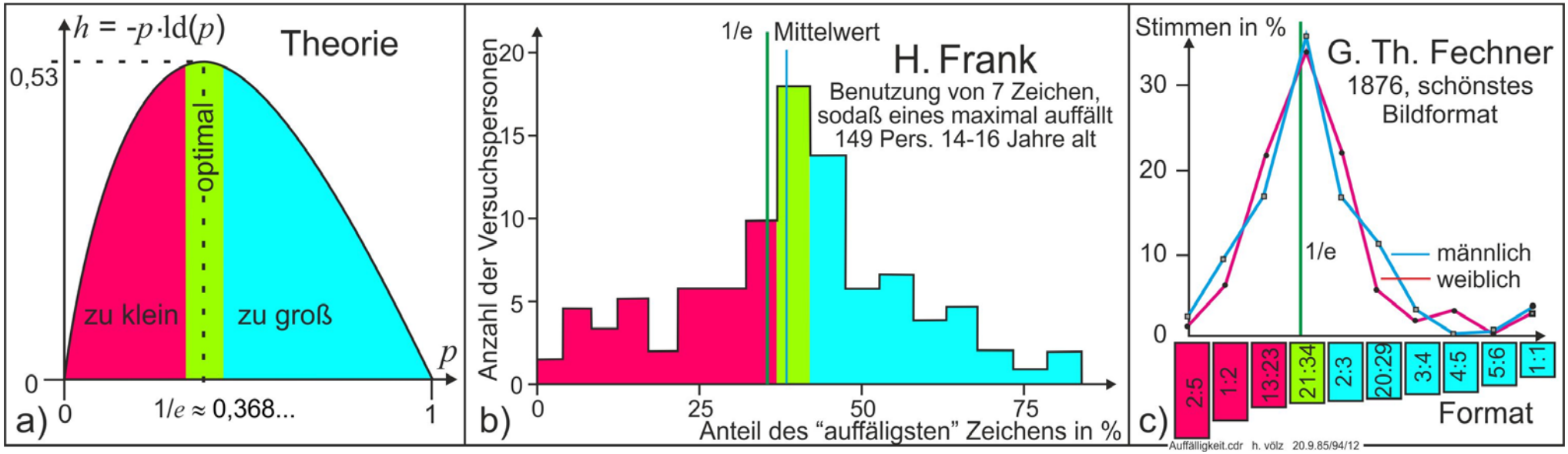
Anwendung Fotografie

Bei der Fotografie, im Film und Fernsehen (z. B. für die Position des Moderators) hat das näherungsweise gleiche 2:3-Verhältnis für die Bildgestaltung generelle Bedeutung



Weiteres Frank-Experiment

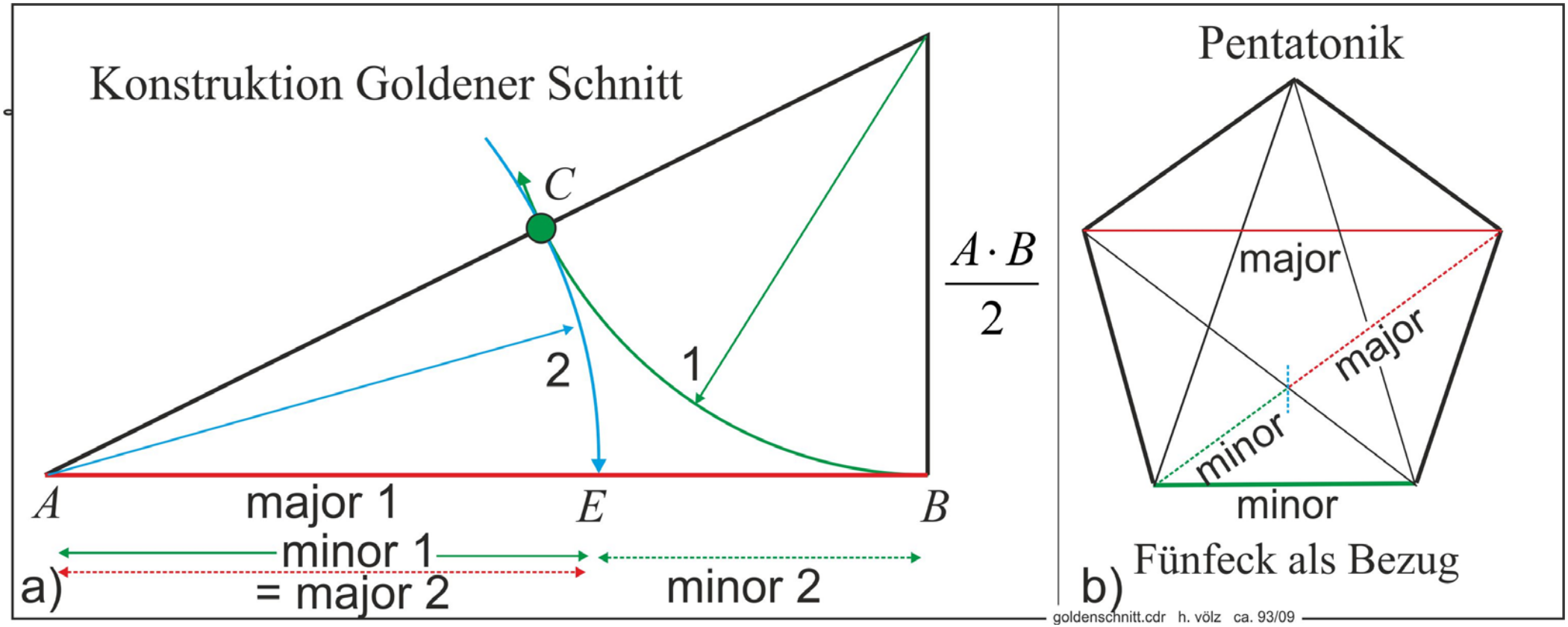
149 Personen soolten 7 Zeichen so verwenden, das eines besonders auffiel.
Dabei lag der Mittelwert mit 37 % besonders deutlichr bei der Auffälligkeit.
Das Bild zeigt dazu den Vergleich vom Verlauf der Kurve $h = -p \cdot \mathrm{ld}(p)$ (a) und alter Fechner-Untersuchung



U. a. war dies der Anlass zu einer einsetzenden erheblichen Kritik zu den Aussagen von Frank.
Der Goldenen Schnitt, ist spätestens im Mittelalter und als Pentatonik bereits den alten Griechen bekannt
Er besitzt einen ähnlichen Zahlenwert und war schon lange für die Schönheit von Bildern bekannt
die geometrische Auffälligkeit sollte also primär sein.
Jedoch die mathematische Informationstheorie ist weitaus allgemeiner

Konstruktion des Goldenen Schnittes

Die Abfolge der Kreisbögen ist durch 1 und 2 gekennzeichnet.
Auch im alten Pentagramm tritt das Verhältnis auf (b).

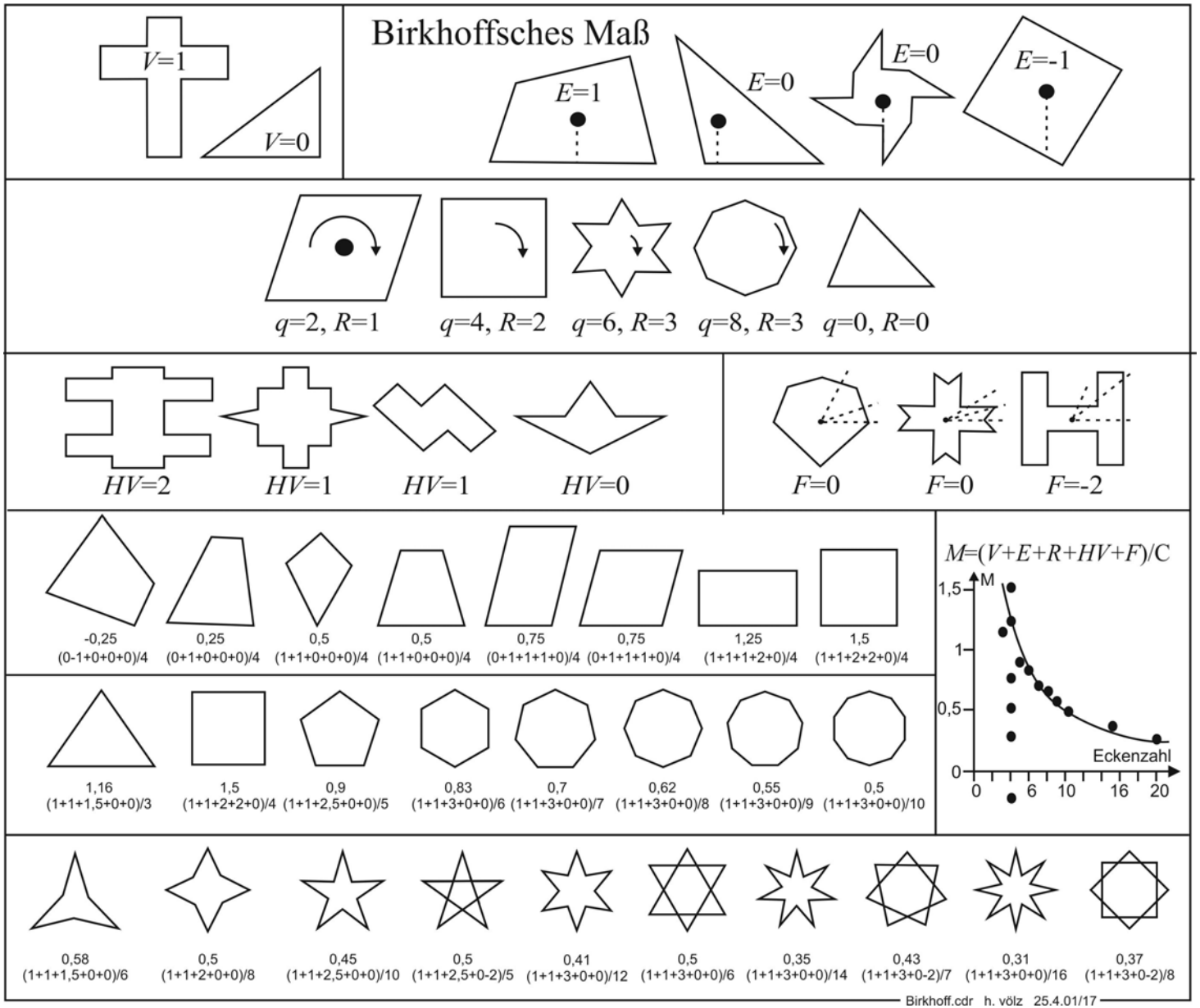


Andere mathematischen Betrachtungen zur Ästhetik gehen auf **Birkhoff** um 1930 zurück

[Bir33], auch [Mas70], [Völ82], S. 284ff.

Birkhoff

ben dem Goldenen Schnitt gibt es zu Bildern nur noch Die wichtigsten Beispiele und Zusammenhänge zeigt Bild 99. Sein Maß *M* berücksichtigt Symmetrien und Komplexität. In der Praxis hat es sich kaum bewährt. Eigentlich wurde es nur von Birkhoff benutzt und auf Sterne, Vasen usw. angewendet



Birkhoff-Formel

$$M = \frac{V + E + R + HV + F}{C}.$$

Die *Komplexität* C ist die minimale Anzahl der geraden Linien, auf denen wenigstens eine Polygonseite liegt.

Die Vertikalsymmetrie V = 1, falls ein Polygon zu einer vertikalen Achse symmetrisch ist, sonst 0.

Da *Gleichgewicht* ist E = 1, falls V = 1 ist, ebenfalls ist E = 1, falls das Zentrum K des vom Polygon eingeschlossenen Gebietes senkrecht über einem Punkt innerhalb einer Horizontalstrecke des Polygons liegt, welche das Polygon von unten her stützt, jedoch derart, dass die Längen beide grösser als 1/6 der gesamten Horizontalbreite des Polygons sind.

Die *Rotationssymmetrie* R ist gleich dem Minimum der beiden Zahlen q/2 und 3, falls Rotationssymmetrie vorliegt mit Winkel $\alpha = 360°/q$, wobei α der kleinste Drehwinkel für Deckungsgleichheit ist. In allen übrigen ist R = 0.

Die Beziehung *Horizontal-Vertikal* beträgt HV = 2, falls alle Seiten des Polygons auf den Linien eines gleichförmigen Horizontal-Vertikal-Netzes liegen. HV = 1 gilt, falls wenige Ausnahmen vorhanden sind oder sich das Polygon in ein anderes Netz von zwei parallelen Geradescharen vollständig einfügt. In den übrigen Fällen gilt HV = 0.

Die *Erfreuliche Form F* = 0 liegt vor, wenn jeder vom Zentrum K (Schwerpunkt) ausgehende Halbstrahl das Polygon in genau einem Punkt schneidet, sonst gilt F = -2.

Schönheit des Hauses

in einem längeren Fortbildungsseminar am Institut für Städtebau und Architektur der Bauakademie der DDR zur Informationstheorie versuchte ich 1985 u. a. die Auffälligkeit auf das übliche Wohnhaus zu übertragen. Dabei ging ich davon aus, wie wahrscheinlich ein Kind ein Haus beschreiben würde:

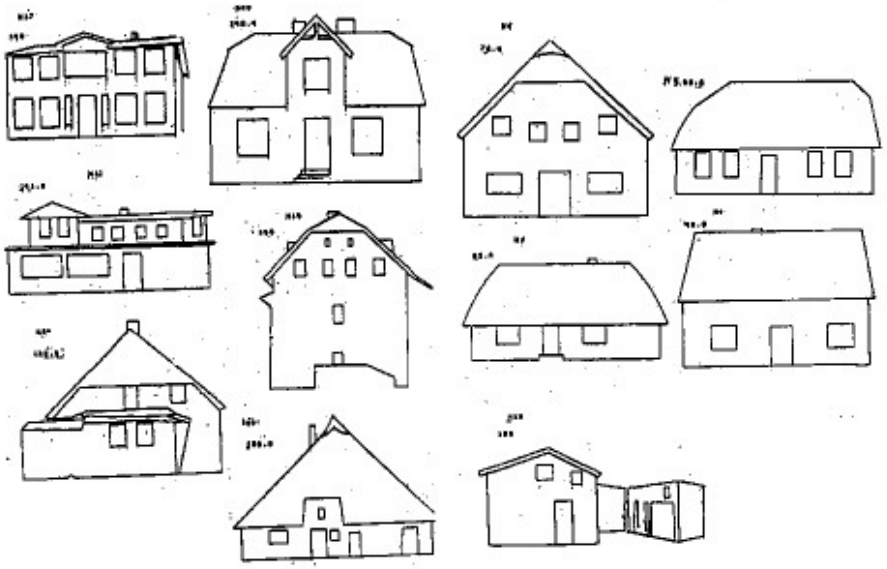
1. Es ist ein Dach über dem Kopf und 2. hat es Türen und Fenster.

Das führte zu umfangreichen Diskussionen mit den teilnehmenden Dozenten und Professoren.
Schließlich wurde ein Dozent beauftragt, während seines gerade bevorstehenden Kurzurlaubs an der Ostsee Häuserfotografien anzufertigen. Er kam mit den 24 Fotografien zurück.

Die Bestimmung der Flächenanteile von den Dächern bzw. Türen + Fenster zur Gesamtfläche ergaben zu meiner Freude ziemlich genau je ≈37 %.
So definierte ich als Summe von beiden ein Maß für die „Schönheit“ des Hauses.
Doch als ich damit dann auch noch die Rangfolge der „Schönheit“ festlegte und die Häuser danach anordnete, gab es deutlichen Protest.

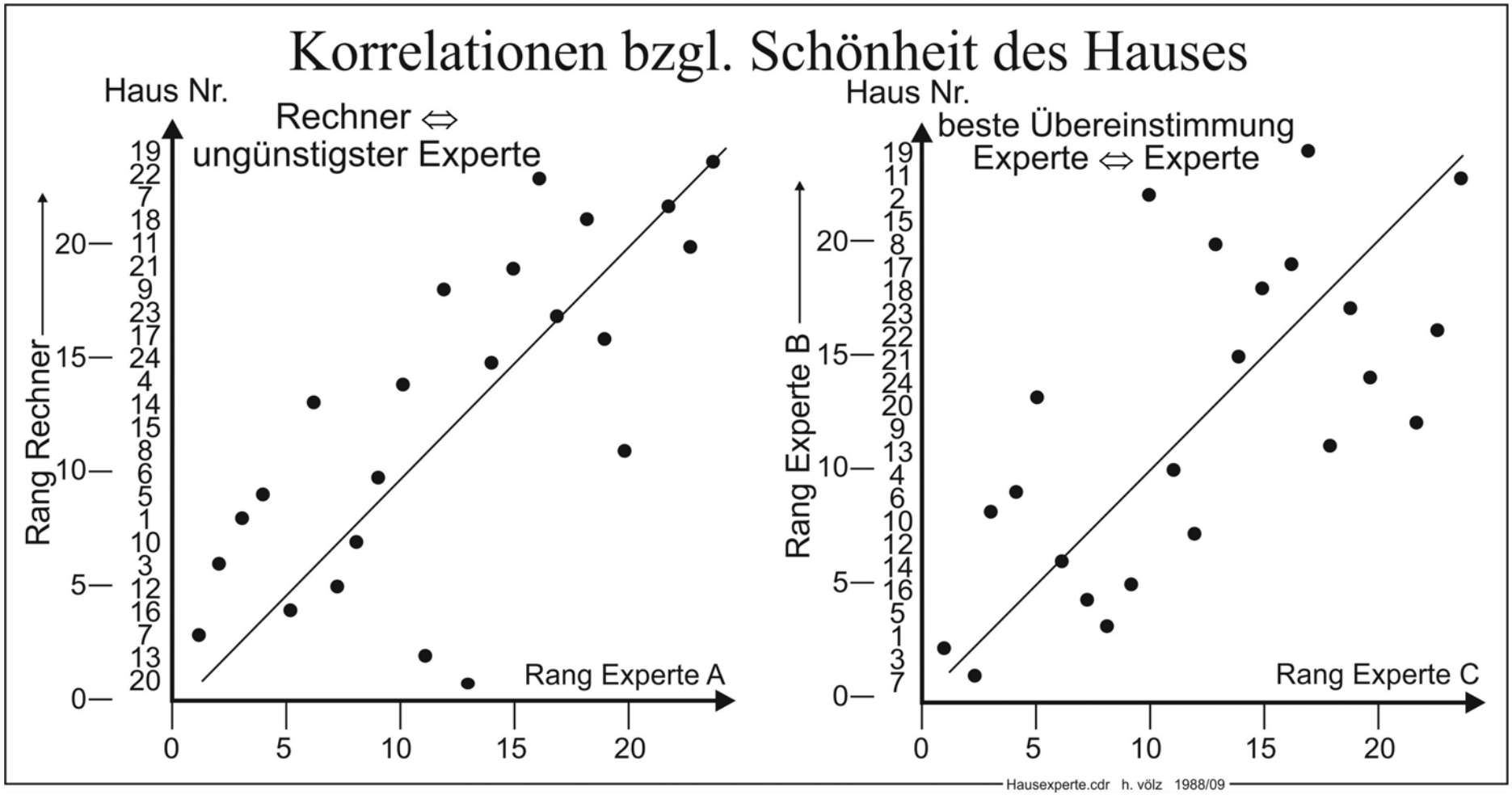
So wurde beschlossen, die Fotografien in Strichzeichnungen umzusetzen, um insbesondere den ästhetischen Vorteil des Strohdachs zu unterdrücken

Dann gaben 10 Professoren ihre Reihenfolge an [Völ88].



Subjektive und objektive Werte

Das dann ermittelte Ergebnis als Vergleich von 21 Urteilen ist nicht ganz einfach darzustellen.
Hier seien nur die zwei wichtigsten Extreme ausgewählt.
Die schlechteste Korrelation zwischen dem Rechnerergebnis und einem Experten.
sie war besser, als jene zwischen den am besten übereinstimmenden Experten.



Publikationsverbot und Nicolai-Viertel

Als ich nun auf die „Schönheit“ der Hochhäuser hinwies, entstand weiterer Protest.
Ich erhielt für zwei Jahre Publikationsverbot.
Offensichtlich wurde es dazu genutzt, um an gerade im Bau befindlicher Architektur des Nicolai-Viertels?!, künstliche bzw. zusätzliche Dächer einzubauen.

Außerdem entstanden neue Proteste gegen die Auffälligkeit. Schließlich wies mich ein Physiologe ausdrücklich auf das Weber-Fechner Gesetz hin. Dort besteht ja der Logarithmus.
Die Auffälligkeit kann so leicht als Bewertung in diesem Sinne angesehen werden.
So gilt: der Goldene Schnitt gilt nur für Bilder, die Auffälligkeit dagegen für alle Wahrnehmungen.
Die informationstheoretische Mathematik der Auffälligkeit ist also im Gegensatz zur Zirkel-Lineal-Konstruktion des Goldenen Schnittes allgemein [Beu89], [Hag58].

Zahlenwerte zum Vergleich

Der Goldener Schnitt und die Auffälligkeit werden zuweilen durch ganzzahlige Verhältnisse genähert. Für die wichtigsten Werte und deren Abweichungen gilt dabei:

Verhältnis	Goldener Schnitt	Auffälligkeit
2:3	+4,9 %	+3,5 %
3:5	-1,8 %	-----
5:8	+0,7 %	-0,7 %
7:11	------	+0,4 %

Die Abweichungen sind also relativ gering.
Deutlicher ist die Abweichung zum massenhaft benutzten DIN-Format [Tuc??].
Es ist dadurch festgelegt, dass bei jeder Halbierung eines Bogens das Seitenverhältnis exakt erhalten bleibt.
Für den Vergleich gilt dann:

Goldner Schnitt: Minor : Major = $(\sqrt{5} - 1)/2$ ... $\approx 0{,}618033988...$

Auffälligkeit: Wahrscheinlichkeit = $1-1/e$ ..... $\approx 0{,}632120558...$

DIN-Format Seitenverhältnis = $1/\sqrt{2}$ ..... $\approx 0{,}707106781...$

Der häufige Gebrauch von DIN-Formaten hat unser Gefühl für den Goldenen Schnitt beeinflusst und es daher näher an die Auffälligkeit bewegt.

Goldener Schnitt andere Zusammenhänge

Erstaunlich ist es, dass der Zahlenwert des Goldenen Schnittes auch in anderen Zusammenhängen auftritt. Einmal sind das Zahlen, die Fibonacci im Buch „Liber abadi“ bei der **Vermehrung von Kaninchen** fand.
Er beginnt mit einem Kaninchenpaar und fragt, wie viele Kaninchen in den folgenden Generationen vorhanden sein werden.
Seine These war, dass jedes Paar pro Monat ein neues Paar gebiert. Dabei setzte er Unsterblichkeit. Dabei ergibt sich die Reihe [Wor77]

$$F(1) = 1,\ F(2) = 1 \text{ und } F(n)=F(n\text{-}1)+F(n\text{-}2).$$

Es entsteht so die Zahlenfolge 1, 1, 2, 3, 5, 8, 13, 21, 34 .... , die gegen $(\sqrt{5} - 1)/2$ konvergiert. F(n) ist auch direkt berechenbar:

$$F(n) = \frac{\left(1 + \sqrt{5}\right)^n - \left(1 - \sqrt{5}\right)^n}{2^2 \cdot \sqrt{5}}.$$

In der Natur, z. B. bei **Blütenblätter**n sind die Fibonacci- Zahlen deutlich bevorzugt. Z. B. ist ein vierblättriges Kleeblatt sehr selten.
Auch **Kettenbrüche** können zum Goldenen Schnitt führen

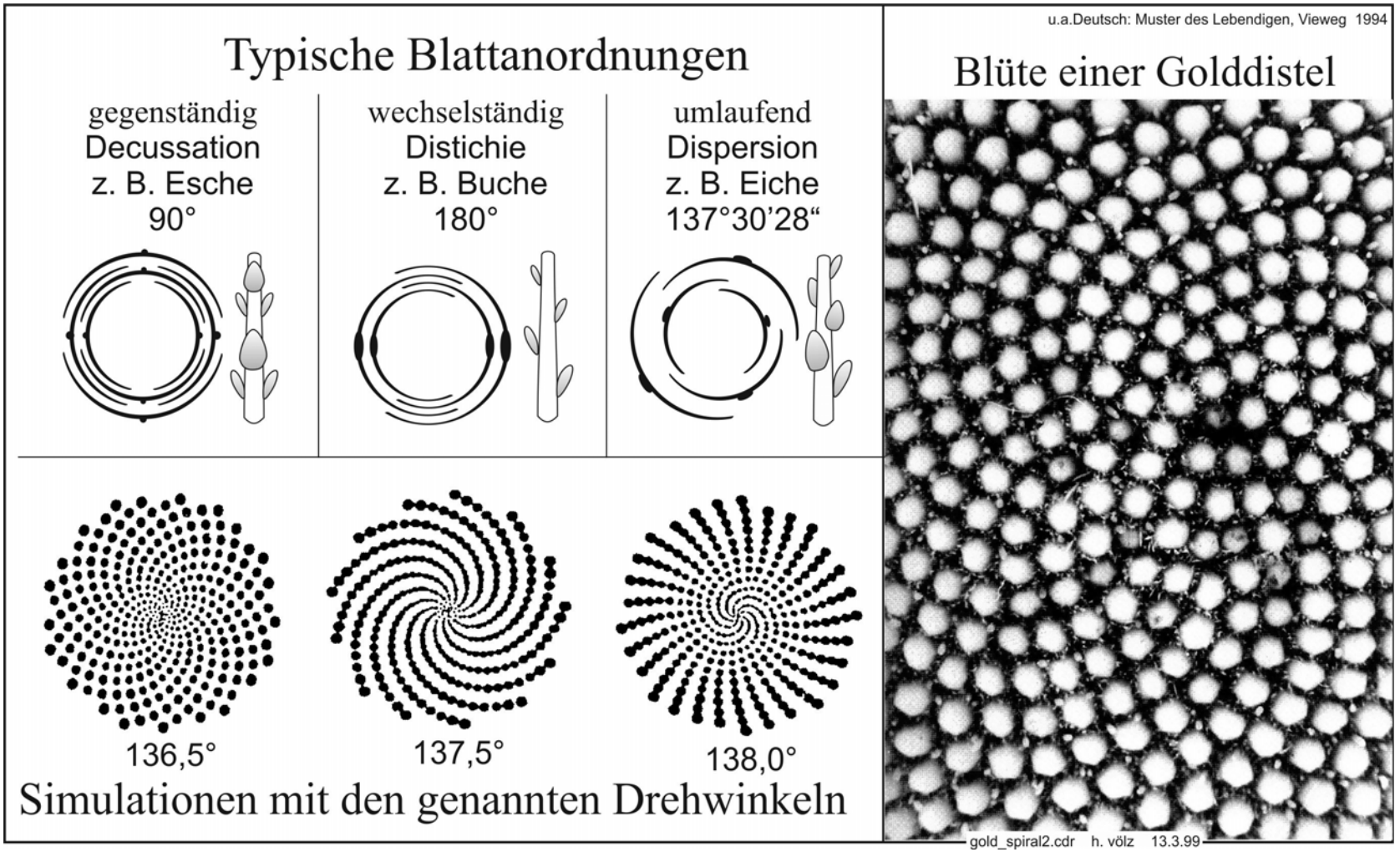
$$x = \cfrac{1}{a + \cfrac{1}{b + \cfrac{1}{c + \cfrac{1}{d + \ldots.}}}}.$$

Für a = b = c = d = e = ... folgt wiederum $x = (1+ \sqrt{5})/2$.

Sonnenblume unerklärbar

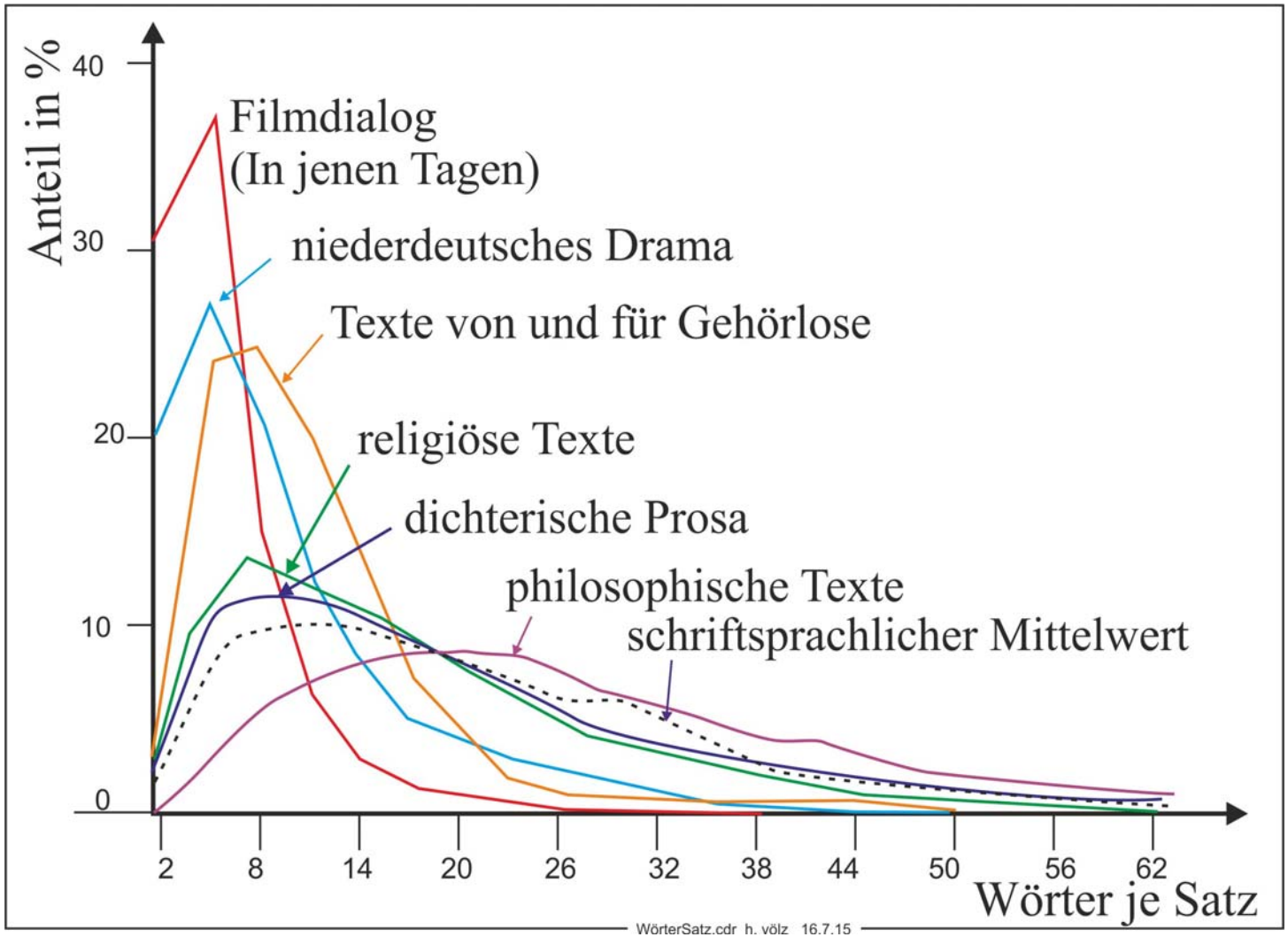
Bei Pflanzen git es 3 typischen Blattanordnungen: gegen-, wechselständig sowie umlaufend.
Für den letzten Fall gilt der harmonische Winkel 137,5° ≈ (1 - g)·360°.
Für g folgt wieder erstaunlich genau der Wert des Goldenen Schnittes

Wie gering die Anweichung für das spiralförmige Muster nur sein darf, zeigen die Simulationen, U. a. die Dolde der Golddistel, die Samen der Sonnenblume und Brokoli.
Für diese Fakten gibt es aber keine einleuchtende Erklärung. Sie treten einfach auf [Völ07], S.62 ff.



Eigenschaften von Texten

Jeder Einzelne und jede Sprache hat bevorzugte Wörter mit weitgehend feststehenden Häufigkeiten.
Auf dieser Grundlage lassen sich individuelle Textstatistiken ermitteln.
Das wurde systematisch durch Fucks untersucht [Fuc64]. Ein erstes Beispiel zeigt das Bild



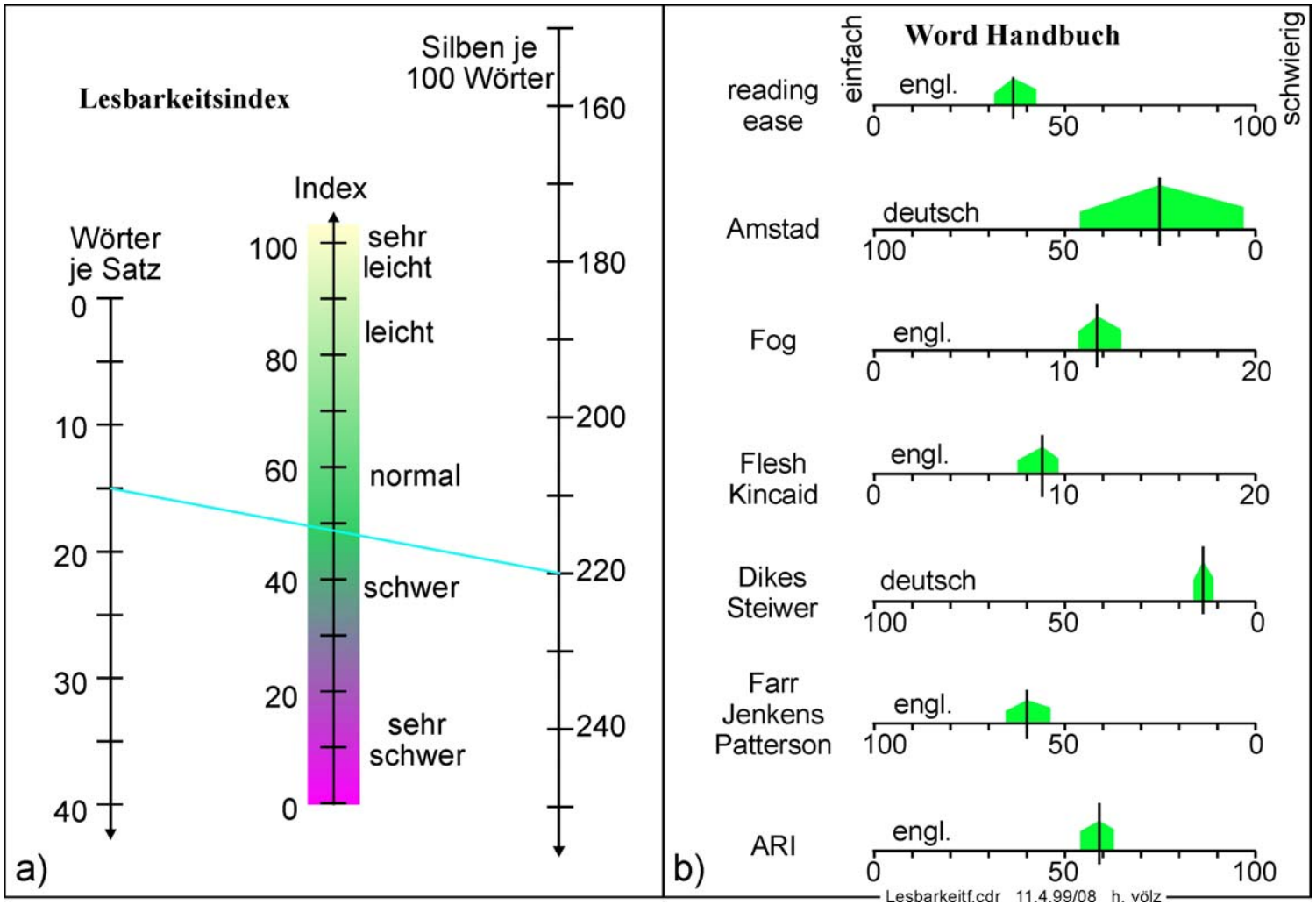
Lesbarkeitsindex

Je nach Alter usw. sind Texte unterschiedlich leicht zu lesen.
Lange Sätze und vielsilbige Wörter erschweren das Verstehen, die Lesbarkeit eines Textes.
So entstand ein Lesbarkeitindex (L in %) , der für Kinder unterschiedlichen Alters Hinweise gibt.
Auch für Journalisten ist so ein Wert nützlich.
Einige Mathematikzeitschriften lassen danach sogar Artikel überarbeiten.
Mit den Wörter je Satz (W) und den Silben je 100 Wörter (S) als Mittelwerte gilt

$$L \text{ (in \%)} \approx 230 - 0.96 \cdot (W + 0.78 \cdot S) \approx 240 - W + 0{,}8 \cdot S.$$

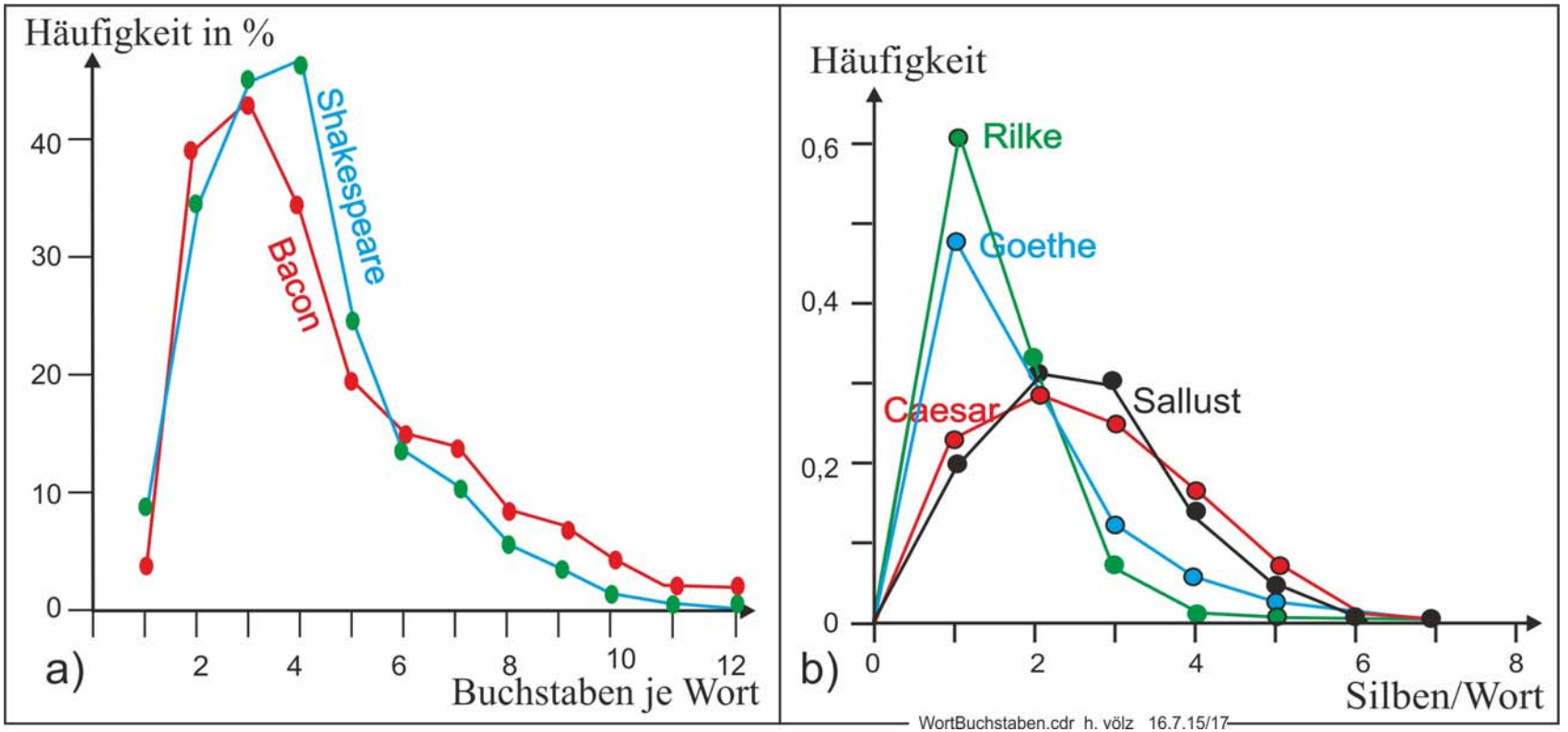
Daraus folgt das linke Nomogramm.

Es enstanden bald mehrere ähnliche Formeln. Eine Anwendung auf die Lesbarkeit von Handbüchern zeigen die rechts stehenden Skalen.

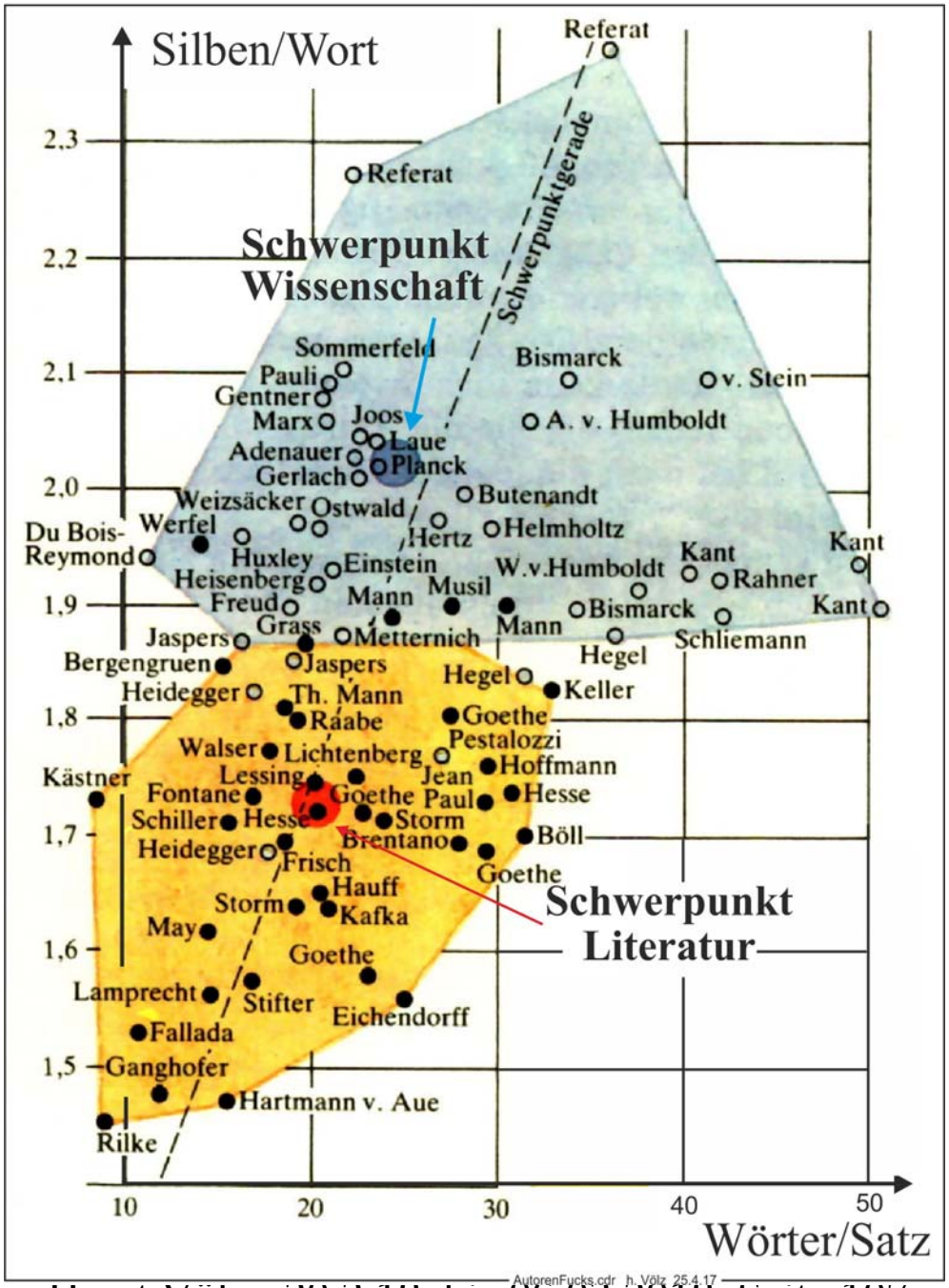


Kennzeichnen einzelner Autoren

Ab 1957 brachten statistische Analysen viele interessante Ergebnisse [Fuc68].
So wurden typische Werte für Autoren gefunden. Einen noch recht einfachen Zusammenhang zeigt das Bild.

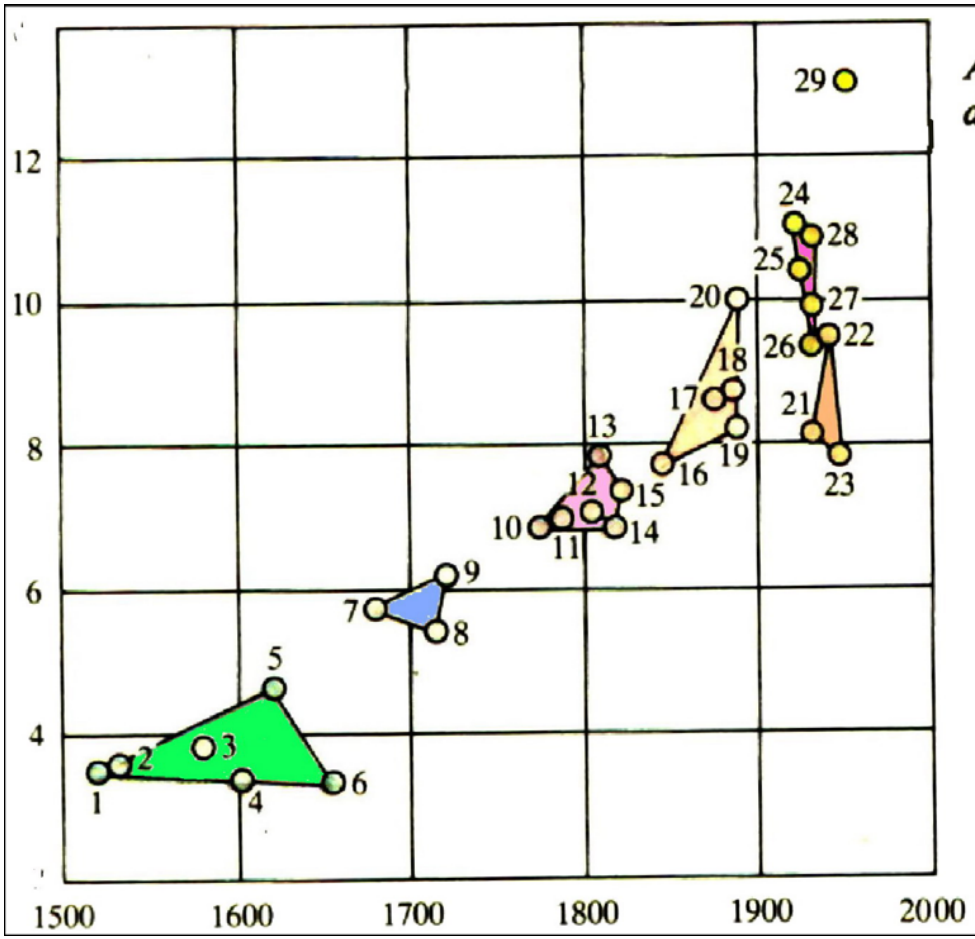


Das folgende Bild zeigt das viel Umfnagreicher. Fucks unterscheidet Wissenschaftler ° und Literaten •.
Ihre Flächen besitzen typische Schwerpunkte.
Die meisten Autoren befinden sich an einen festen Punkt im Diagramm.
Nur wenige wie z. B: Goethe sind „wortgewaltig“. Sie verlagern ihn im Laufe ihrer Lebenszeit.
Das ermöglicht sogar nachträgliche Änderungen in ihren Werken zu erkennen und zeitlich einzuordnen.


Silben/Wort
Wörter/Satz
2,3
2,2
2,1
2,0
1,9
1,8
1,7
1,6
1,5
10
20
30
40
50
Schwerpunktgerade
Schwerpunkt Wissenschaft
Schwerpunkt Literatur
Referat
Sommerfeld
Pauli
Gentner
Marx
Joos
Laue
Planck
Adenauer
Gerlach
Bismarck
v. Stein
A. v. Humboldt
Butenandt
Weizsäcker
Ostwald
Hertz
Helmholtz
Du Bois-Reymond
Werfel
Huxley
Einstein
Heisenberg
Freud
Mann
Musil
W.v.Humboldt
Kant
Rahner
Bismarck
Schliemann
Grass
Metternich
Jaspers
Bergengruen
Hegel
Keller
Heidegger
Th. Mann
Raabe
Goethe
Pestalozzi
Walser
Lichtenberg
Jean Paul
Hoffmann
Kästner
Lessing
Fontane
Hesse
Schiller
Storm
Brentano
Böll
Frisch
Hauff
Kafka
May
Lamprecht
Stifter
Eichendorff
Fallada
Ganghofer
Hartmann v. Aue
Rilke
AutorenFucks.cdr h. Völz 25.4.17

Anwendung Musik

Für Musik konnte Fucks leider keine ähnlich guten Ergebnisse finden
Erst mit der Krümmung als zweite Ableitung der Statistik fand er einen zeitlichen Zusammenhang.
Seine Vermutung, dass um 1900 eine Zweiteilung beginne, hat sich aber nicht bestätigt



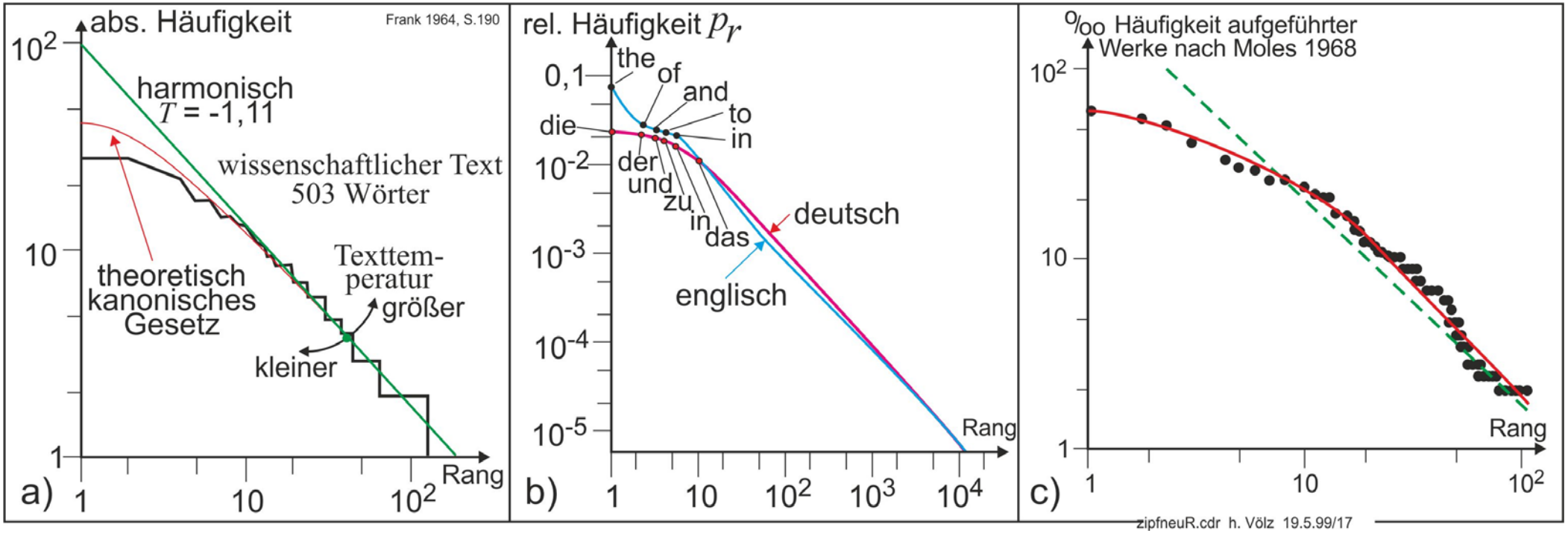
Auswertung von 29 Musikwerken hinsichtlich der Tonhöhenstreuung der 1. Violine

1 – Willaert: Fantasien;
2 – de Modena: Fantasien;
3 – Palestrina: Ricercari;
4 – Haßler: Intraden;
5 – Schein: Suiten;
6 – Rosenmüller: Stud.-M.;
7 – Corelli: Concerto gr. 8;
8 – Vivaldi: Concerto gr. 3,2;
9 – Bach: Konzert für 2 Viol.;
10 – Mozart: Violin-K., KV 219;
11 – Mozart: Symph. g-Moll;
12 – Beethoven: 5. Symphonie;
13 – Beethoven: Streich-Qu. 74;
14 – Spohr; Violin-Konzert;
15 – Schubert: 8. Symph.;
16 – Schumann: 2. Symph.;
17 – Brahms: Violin-Konz.;
18 – Tschaikowski: 5. Symph.;
19 – Strauss: Till Eulenspiegel;
20 – Tschaikowski: 6. Symph.;
21 – Hindemith: Mathis d. M.;
22 – Bartók: Suite II;
23 – Egk: Orchester-Suite;
24 – Berg: Streich-Qu. 3;
25 – Webern: Streich-Trio 20;
26 – Berg: Violin-Konzert;
27 – Webern: Streich-Qu. 28;
28 – Schönberg: Violin-K.;
29 – Nono: Varianti

Zipfsches Gesetz

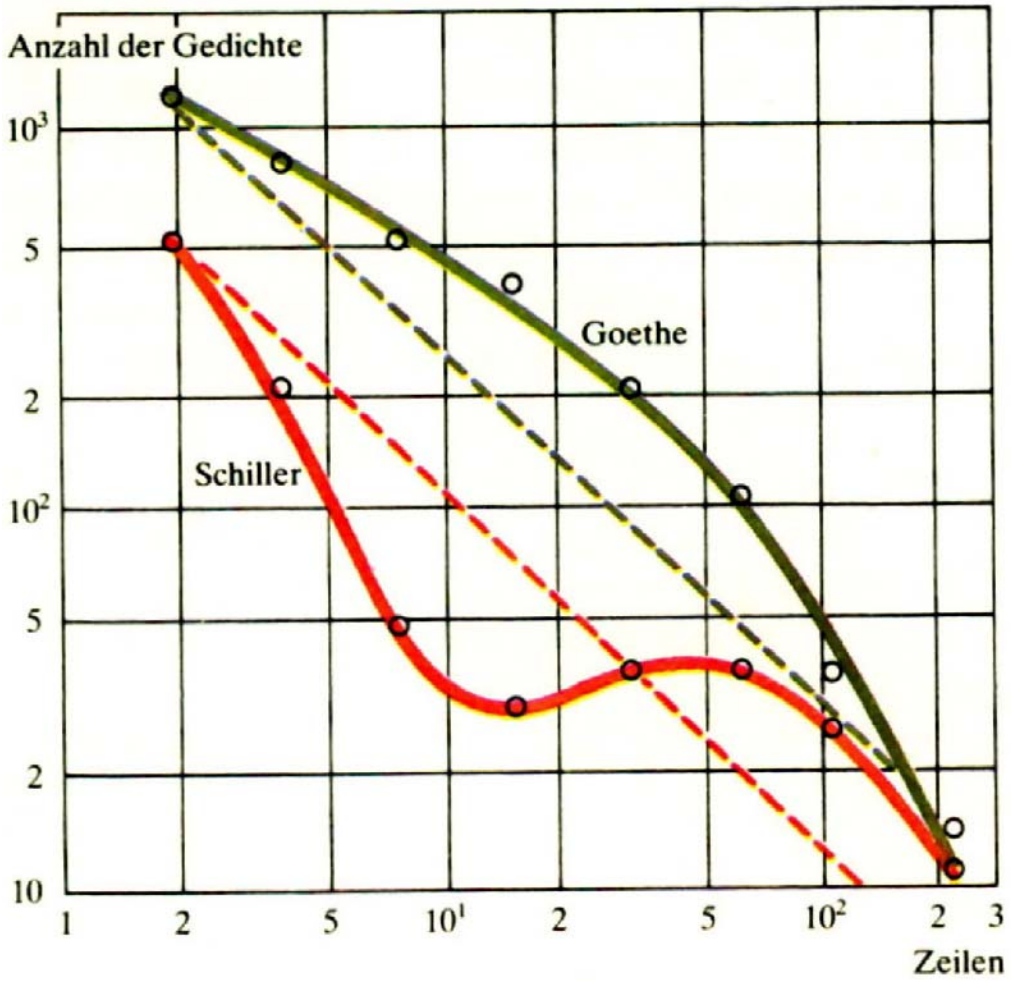
Es gilt $p(R) \approx \frac{K}{R}$. Die Wörter werden in Reihenfolge ihrer Häufigkeit $p(R)$ als Rang R geordnet.

Es gibt vom Gesetz mehrere Abwandlungen des ursprünglichen harmonischen Gesetzes von Estop und Zipf
Für die Anwendungen in Literatur und Musik ist der exponentielle Zusammenhang bedeutsam.
Die Steigung der Geraden heißt Texttemperatur T. Dr Zusammenhang tritt generell auf, selbst für Haushaltsmittel usw.
Die Beethoven-Sonate A-Dur op. 3 enthält im 2. Satz 1630 Noten mit 71 Tonhöhen. Mit einem Korrelationskoeffizienten $\rho^2 = 0.855$ ergeben sich $K = 0{,}625$, $T = 0{,}708$ sowie $H = 5{,}124$ Bit/Note.



Gedichtlängen bei Goethe und Schiller

Bereits 1954 analysierte Ernst Lau die Zeilenzahl aller Gedichte von Goethe und Schiller
Er fand das Ergebnis vom Bild [Lau54].
Ohne sich auf das Zipfsche Gesetz zu beziehen, fiel ihm der „ungewöhnliche“ Verlauf bei Schiller auf.
Er folgerte, dass es für die „Senke“ Ursachen geben müsse.
Schiller hatte oft Geldsorgen. so könnte er wegen des Zeilenhonorars einige Gedichte verlängert haben.
Das hat damals in der DDR sehr harte Kritik der Germanisten ausgelöst
und Prof. Lau hätte es fast den Institutsdirektor (Optik u. Spektroskopie) gekostet. In anderen Fällen,
Später wurden deutlich ähnliche Fakten auch bei andere bekannt.



Johannesevangeliums

Sehr lange war die Autorschaft des Johannesevangeliums unsicher.
Zuweilen wurde der Autor von der Apokalypse vermutet.
1960 brachte die eine Anwendung der Textstatistik die Entscheidung.
Da die Apokalypse 10 412 Wörter besitzt, musste zum Vergleich das Johannesevangelium in zwei Teile zu je 8 500 Wörtern zerlegt werden. es ergaben sich die folgenden Übergangsmatrizen.

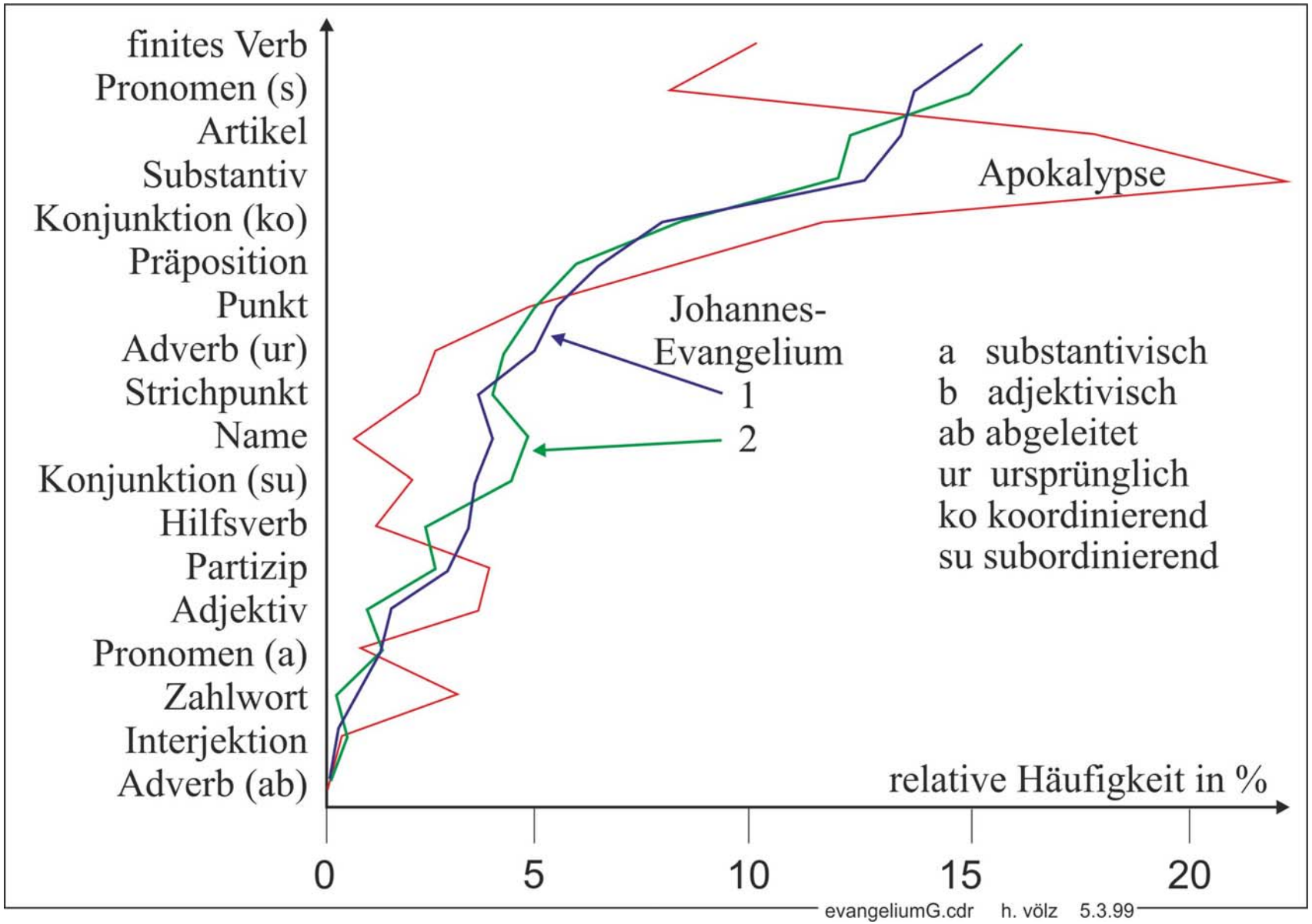
Johannes-Evangelium 1			**Johannes-Evangelium 2**		**Apokalypse**	
	Artikel	Substantiv	Artikel	Substantiv	Artikel	Substantiv
Artikel	8	674	6	689	6	1106
Substantiv	148	32	121	21	379	92

Das folgende Bild zeigt relativen Häufigkeiten der verschiedenen Wortarten
Mit dieser Untersuchung war die häufige Annahme mit sehr hoher Wahrscheinlichkeit widerlegt.

Das war der Beginn für viele statistische Untersuchungen, von denen nur wenige zuvor beschrieben sind.

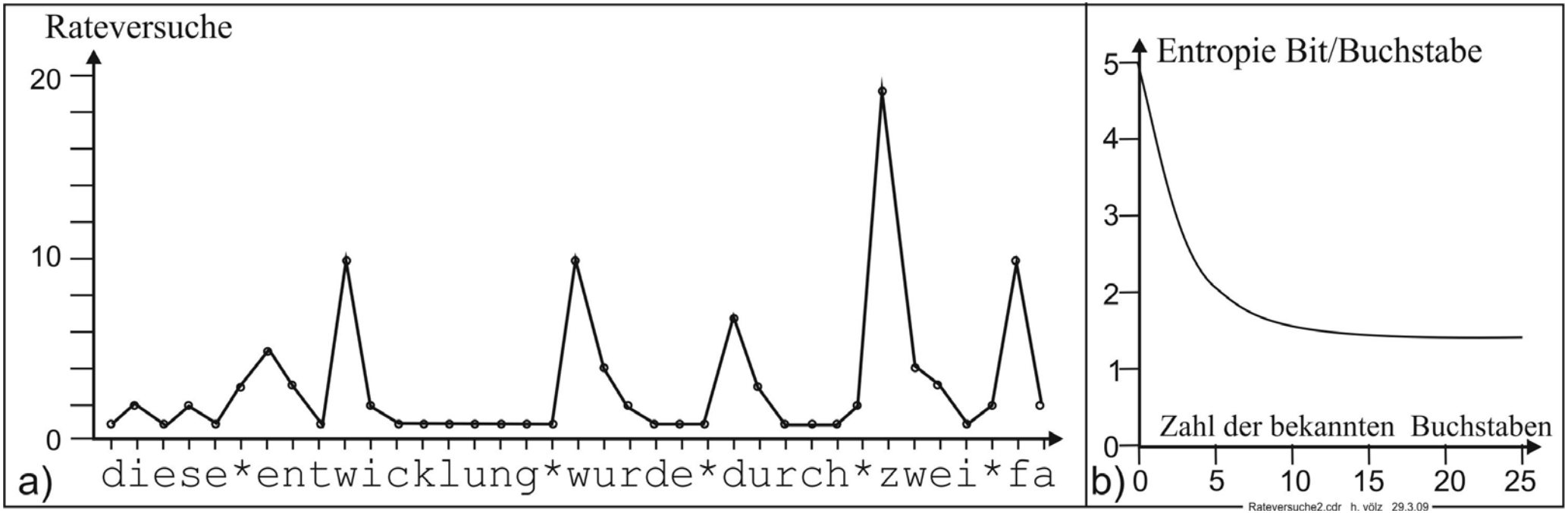
Es ist jedoch nicht erklärbar, warum diese Methoden ab etwa 1980 aufhörten, obwohl dann die Rechentechnik effektiv helfend zur Verfügung stand.

Apokalypse und Johannes-Evangelium 2.



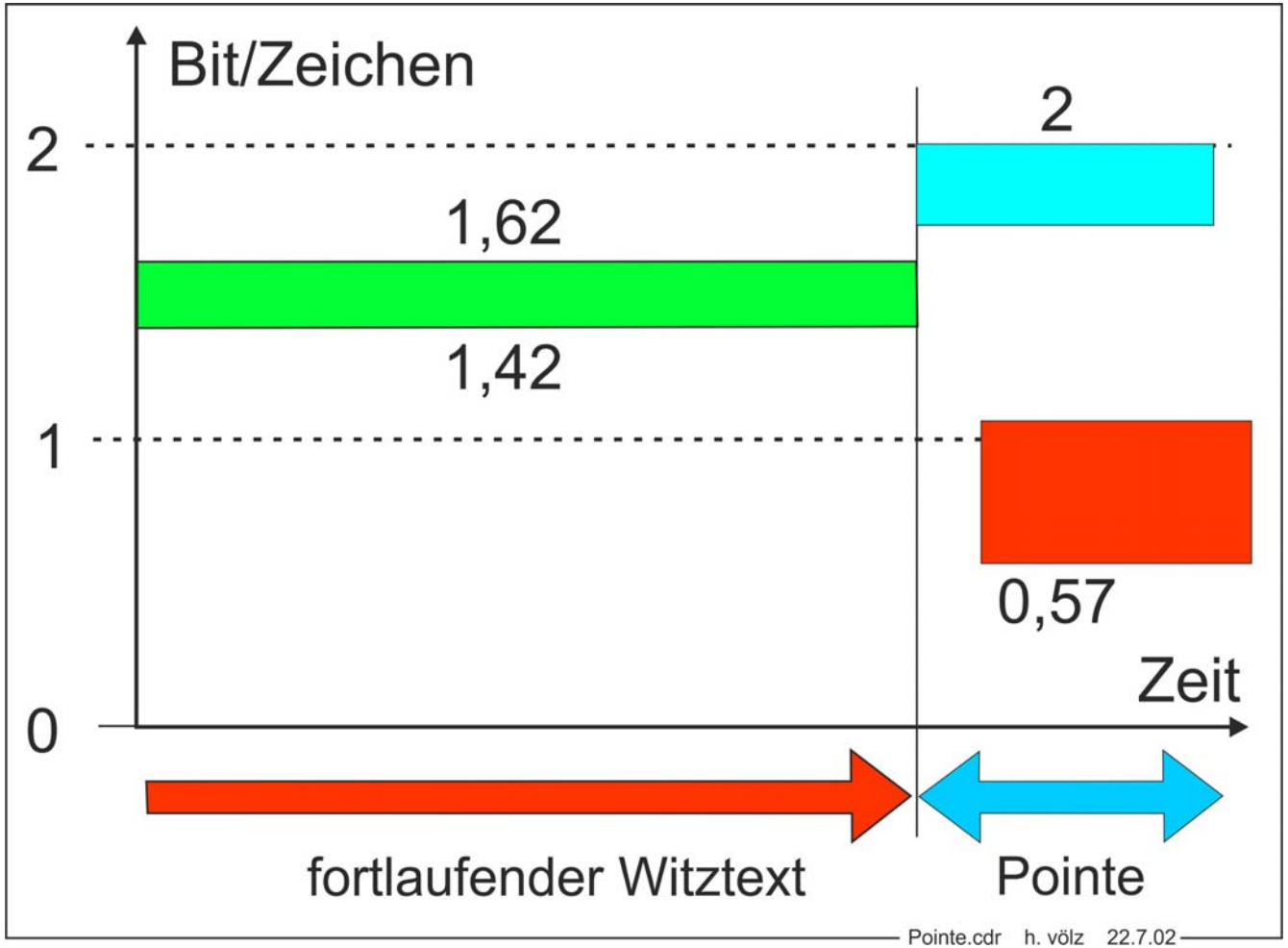
Rateversuche

Mittels Ja/Nein- Rateversuchen kann die Entropie je Buchstabe ermittelt werden.
Das Bild zeigt einen experimentellen Verlauf und b) statistisch ermittelte Ergebnisse.
Im Kontext sinkt die Entropie der Einzelbuchstaben sinkt von ≈ 4,7 Bit/Buchstabe auf den Grenzwert von rund 1,7



Das Witz-Erlebnis

Bei den Pointen von Witzen und dem Eintreten von Tragik ist es deutlich anders.
Intuitiv hat dafür bereits Freud die Begründung gegeben [Fre85].
Das Bild zeigt einen experimentell ermittelten Verlauf für Witze.
Die Entropie steigt hierbei von etwa 1,5 Bit/Zeichen auf gut 2 an.



Leistungen von Shannon

Cloude Elwood Shannon lebte vom 30.04.1916 bis 24.02.2001.
Er zählt zu den bedeutsamsten Wissenschaftlern.
Ohne seine Theorie würde die meiste heutige Informationstechnik bestenfalls technisch funktionieren, aber wohl kaum verstanden werden.
Dabei muss betont werden, dass es zur Zeit seiner Arbeiten noch keine Digitaltechnik gab.
Er erhielt keinen Nobel-Preis, weil es ihn für dieses Gebiet nicht gibt.
Für die fast äquivalente Fields-Medaille der Mathematiker wären andererseits neue mathematische Grundlagen notwendig gewesen.
Von Shannon gibt es nur wenige, aber dafür immer wissenschaftlich sehr fundamentale Arbeiten hoher Qualität. Seine wichtigsten **Lebensdaten** und seine Literatur sind:

1936 - 1938	Magister (PhD) „Boole'sche Algebra bei Relais“
1940	Dissertation „Algebra für theoretische Genetik“
1941	Fellow of the IAS, Princeton
1940 - 1945	Grundlagen zur Kryptographie (Geheimhaltung USA)
1941 - 1972	Mathematics Department der Bell-Laboratorien
1948	„Mathematical Theory of Communication“ Informationstheorie (Eingangsdatum aber 24.3.1940)
1958	Professor am MIT (Massachusetts Institute of Technology)
1960	Kybernetik, Shannon Maus, Labyrinth usw.

Publikationen von Shannon

A Mathematical Theory of Communication, Bell Systems Techn. J. 27 (Juli 1948), S. 379 - 423 und (Oktober 1948) S. 623 - 656. (eingereicht 24.3.1940). (Auch in: University Illonois Press 1949). Teil 2 auch: „Communication in the Presence of Noise“. Proc. IRE 37 (1949) pp. 10 - 20, (eingereicht am 24.03.1940), Übersetzt in: Mathematische Grundlagen der Informationstheorie. R. Oldenbourg, München - Wien, 1976

Prediction and Entropie of printed English Bell Sys. Techn. J. 30 (1951) 1, 50ff.

A symbolic analysis of relay and switching circuits (Eine symbolische Analyse von Relaisschaltkreisen), Transactions American Institute of Electrical Engineers 57; 1938, S. 713 - 723 (Eingang 1.3.38, revidiert 27.5.38; = seiner Master Thesis v. 10.8.37)

Die mathematische Kommunikationstheorie der Chiffriersysteme. Bell Systems Technical Journal 28 (1949) S. 656 - 715 (Ursprünglich 1.9.45)

Datenglättung und Vorhersage in Feuerleitsystemen Technical Report Bd. 1; Gunfire Control (1946) S. 71 - 159, 166f. (Nur Auszug vorhanden)

Die Philosophie der PCM Proc. IRE 36 (1948), S. 1324 - 1331 (Eingang 24.5.48)

Vorführung einer Maschine zur Lösung des Labyrinthproblems. Transactions 8th Cybernetics Conference 15. - 16.3.1951 in New York. Josiah Macy Jr. Foundation, (1952) S. 169 - 181

Eine Maschine, die beim Entwurf von Schaltkreisen behilflich ist (Proc. IRE 41 (1953) S. 1348 - 1351, (Eingang 28.5.53, revidiert 29.6.53)

Eine gedankenlesende Maschine. Bell Systems Memorandum 18.3.53 (Typoskript 4 Seiten)

Vorhersage und Entropie der gedruckten englischen Sprache. (Prediction and Entropie of printed English) Bell Systems Technical Journal 30 (1951) S. 50 - 64 (Eingang 15.9.1950)

Ein/Aus, Ausgewählte Schriften zur Kommunikations- und Nachrichtentheorie. Verlag Brinkmann + Bose, Berlin 2000.

Mehr Details und die vollständigen Arbeiten enthält [Roc09].

Wichtige Geschichtsdaten

1876 Fechner: Untersuchungen zu ästhetischen Bilderrahmen.
1924 Küpfmüller: Systemtheorie
1928 R. V. L. Hartley (*1888) formuliert den logarithmischen Zusammenhang zwischen Signalzahl und Information.
1930 Birkhoff: ästhetisches Maß
1933 Kotelnikow (1908*) formuliert als erster Grundlagen für ein Abtasttheorem.
1940 24.3. Eingangsdatum der Shannon-Arbeit im JIRE.
1946 Dennis Gabor (1900 - 1979) führt als kleinste Signaleinheit logon ein.
1948 Norbert Wiener (1894 - 1964) führt den Begriff der Information ein.
1948 J. W. Tukey benennt als kleinste Nachrichteneinheit „Bit“ (binary digit).
1949 Informationstheorie, Claude Shannon
1949 Shannon-Fano-Kodierung
1954 Carnap-Entropie
1956 Moles ästhetische Wahrnehmung
1960 H. Frank: Maß der Auffälligkeit
1962 Renyi: α-Entropie
1963 Bongard-Weiß-Entropie
1965 Marko: bidirektionale Information
1987 Hilberg deterministische Informationstheorie

Zusammenfassung zur S-Information

- Das primäre Ziel ist die **Nachrichten**- (teilweise auch **Speicher**-) Technik.
- Ihre Entropie und Kanalkapazität bestimmt die **theoretisch möglichen Grenzen**.
- Sie liefert Grundlagen zur **Fehlererkennung, -korrektor** sowie **Komprimierung** und ***Kryptografie***.
- Entscheidend für Berechnungen sind die ***Wahrscheinlichkeit bzw. Häufigkeit*** von Signalen.
- Es ist weder sinnvoll noch notwendig, die ***Inhalte der Zeichen*** zu berücksichtigen.
- Für die ***Quellen-Codierung*** sind Algorithmen, Codebäume oder Tabellen wichtig.
- Primär gilt die Theorie für ***diskrete Zeichen***. Mit zusätzlichen ***Störungen*** kann sie auf ***kontinuierliche*** Signale erweitert werden.
- Generell entstehen beim Übergang zwischen kontinuierlich ⇔ diskret ***Unschärfen und Fehler***.
- Bei der Anwendung bewirkt das ***Sampling-Theorem*** Grenzen der Genauigkeit.
- Einen Ausweg ermöglicht die ***kontinuierliche Digitaltechnik***.
- Mittels der Kanalkapazität kann die ***minimal je Bit notwendige Energie*** bestimmt werden.
- Für die vielen ***nicht nachrichtentechnischen*** (z. B. künstlerischen) ***Anwendungen*** ist die ***Auffälligkeit*** besonders nützlich.