Fehlerbehandlung

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

Bei ***kontinuierlichen*** Signalen werden hauptsächlich Spektrum und Kurvenform verändert.
Das ist schwierig zu korrigieren

Bei ***diskreten, digitalen*** Signalen verändern die Störwirkungen die Signalwerte.
Siewerden in andere umgewandelt oder mit anderen vertauscht
Es können auch signalwerte ausfallen oder hinzukommen
Sind von *n* übertragenen Werten beim Empfang *m* falsch, so beträgt die ***Fehlerrate*** $f_R = m/\mathrm{n}$.

Digitale Fehler müssen zunächst durch **Fehlererkennung** (EDC error detection code) erkannt werden.
Oft genügt es deren Dateiorte zu bestimmen und dann $0 \rightarrow 1$ bzw. $1 \rightarrow 0$ zu tauschen.
Durch Redundanz (eingefügte Zusatzwerte) werden ungültige Signale erkannt
Das realisiert die **Fehlerkorrektur** (ECC error correction code).

Dabei wird vorausgesetzt, dass keine ***Taktfehler*** auftreten.

Für alle Fehlerbehandlungen ist vorteilhaft, wenn eine Datei nur wenige Fehler auftreten.
Deshalb werden sie in passende Abschnitte (***Blöcke)*** zerlegt.
Eine dafür vorteilhafte Maßnahme ist das Spreizen.

Die optimale **Fehlerrate** liegt meist bei 10^{-4} bis 10^{-5}. Gilt auch bei der Genetik.

Fehlertypen

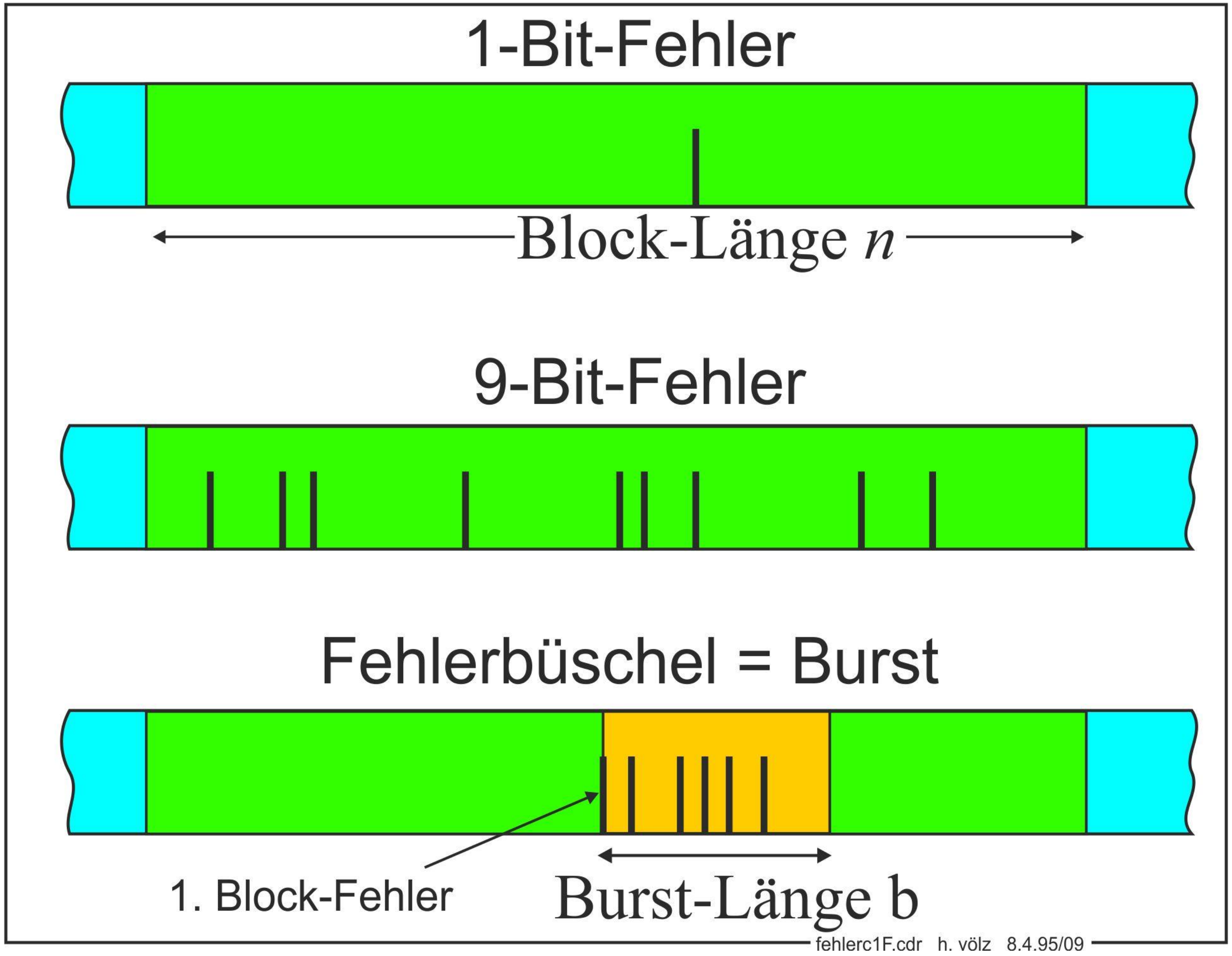
Die meisten *Fehler treten zufällig* auf. Daher besteht ein statistischer Zusammenhang.

Es kann ***kaum eine absolut fehlerfreie*** *Übertragung* geben. Es sind aber sehr kleine Fehler erreichbar. Jedoch ihre Feststellung 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 Einzelfehlern treten auch Bündel- bzw. Büschelfehler = ***Burst*** auf.
Bei ihnen liegen mehrere Fehler-Bit dicht gehäuft beieinander. (magnetischen Speicherung!)
Auch Funken, Blitze, Ein- und Ausschalten von Geräten usw. können Fehler erzeugen

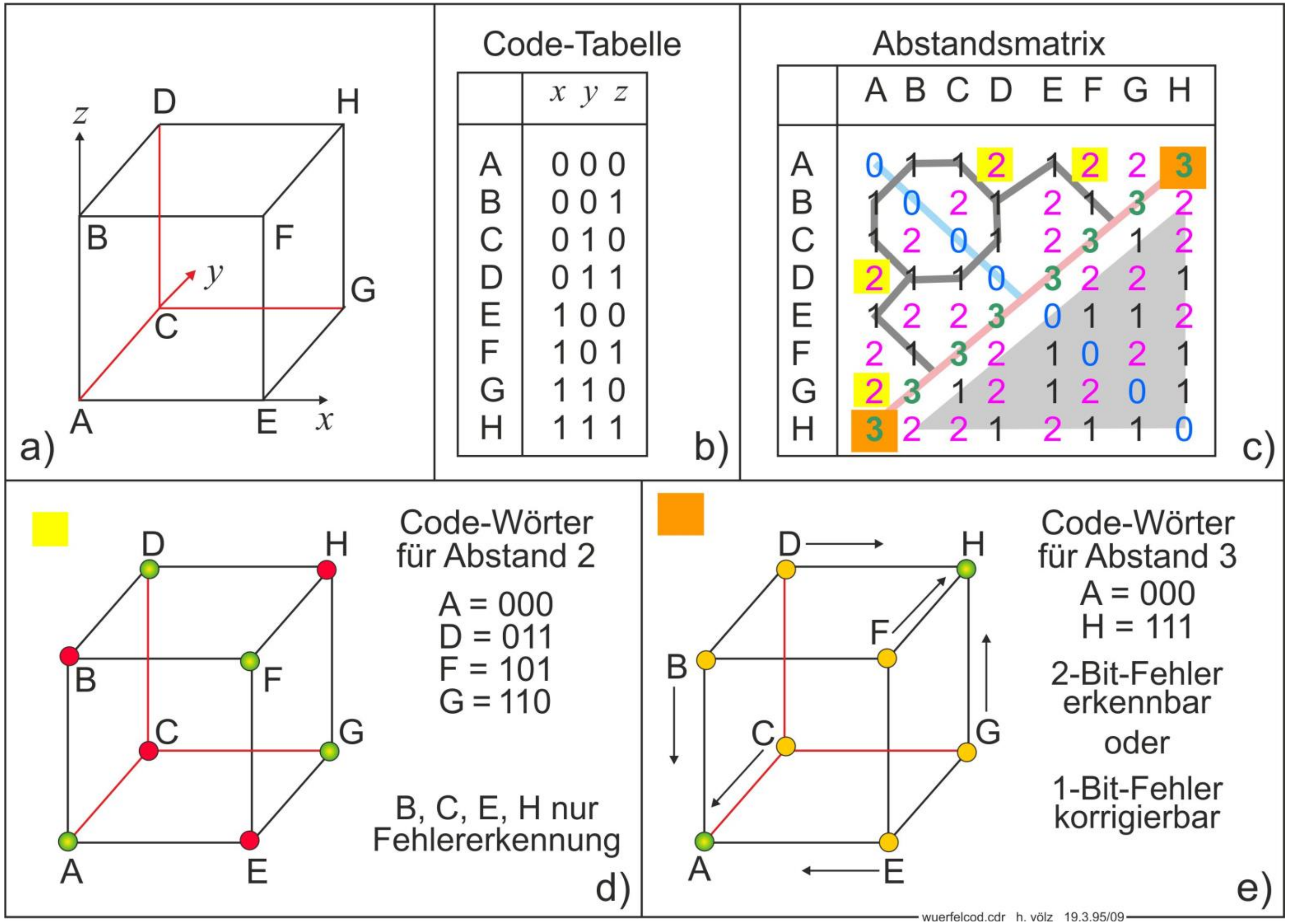
Fehler-Arten



Anschauliche Einführung der Korrektur

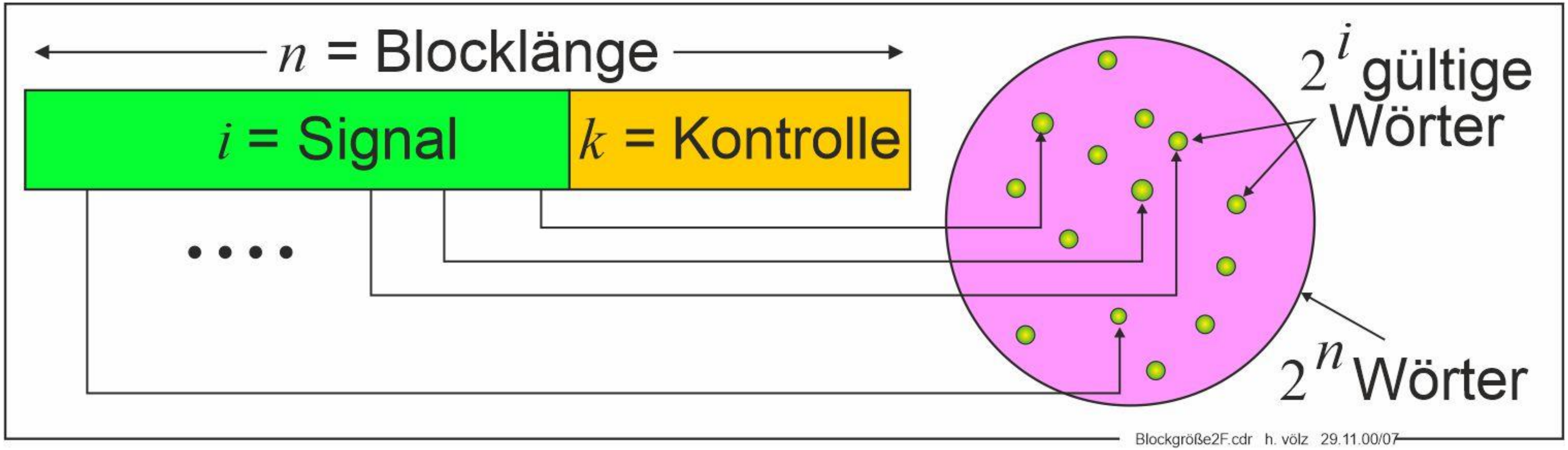
3 mögliche Bit in den drei *xyz*-Koordinaten lassen sich als **Würfel** darstellen (a).
Sie ermöglichen die 3-Bit-Wörter von A bis H der Codetabelle (b).

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*** des Signals die beiden, aus $n > i$ Bit (Kanalwortlänge *n*) bestehenden ***gültigen*** bzw. ***ungültigen*** Wörter.



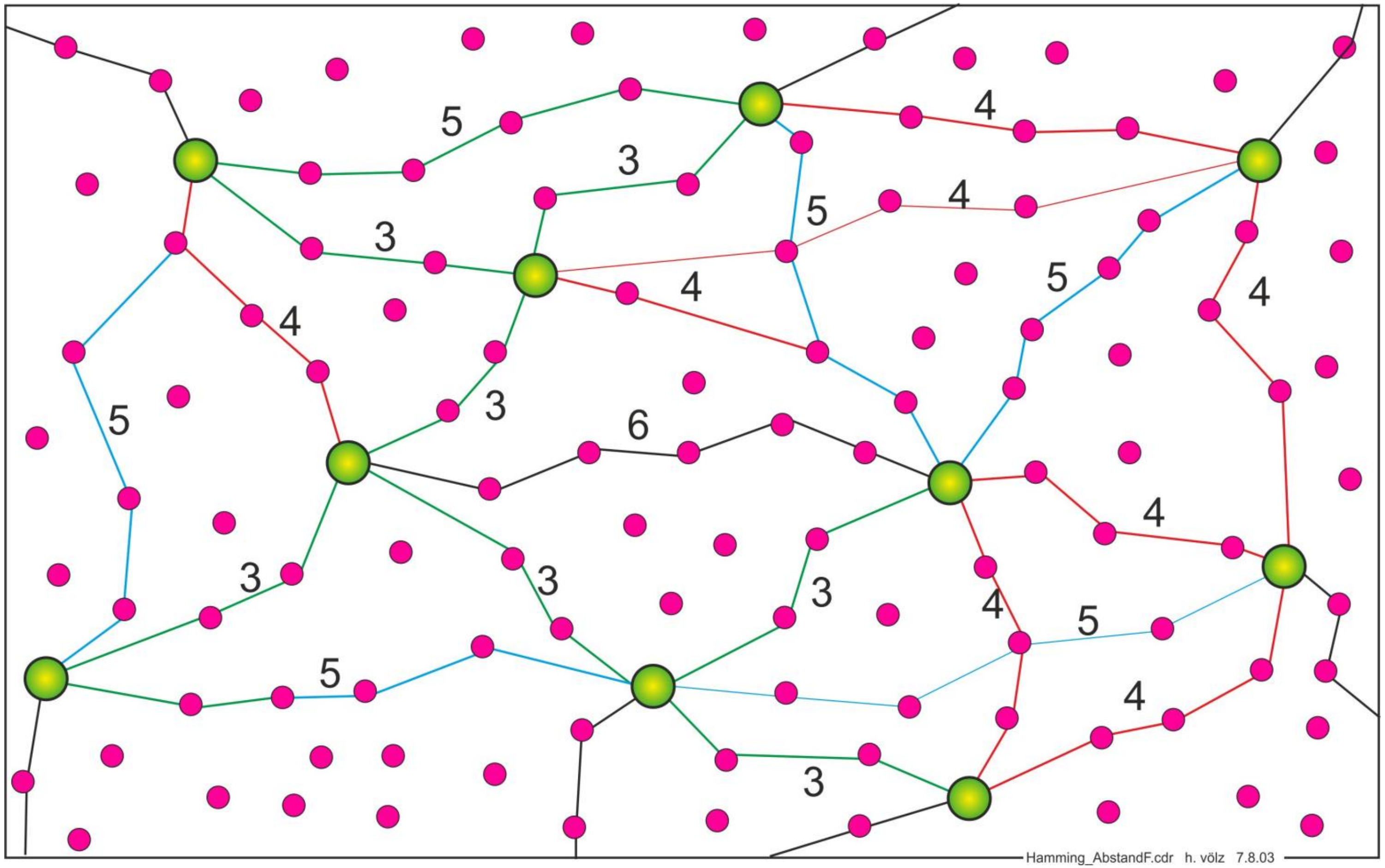
Die **Zuordnung** zwischen **übertragenden** (Block) und **gültigen** Wörtern (Signal) kann Vielfältig sein
Für die Erkennung müssen aus den 2^n-langen Wörtern nur *m* gültige Wörter festgelegt werden,
Der Bit-Abstand zwischen allen ihnen soll möglichst groß sein.
Bei der vorigen 3-Bit-Anordnung mit Würfel war das noch recht übersichtlich.
Beim Versuch mit 4 oder gar mehr Bit das Finden recht kompliziert.
Daher ist die Auswahl der gültigen Wörter die erste schwierige Aufgabe für die Fehlerbehandlung.

Hamming-Abstand

Für ***n*-Bit-Wörter** ist ein kaum vorstellbarer *n*-dimensionaler Raum erforderlich.
Das Bild zeigt dazu einen eingeebneten Ausschnitt seiner Oberfläche.
Gültigen Zeichen sind wieder grün hervorgehoben.
0↔1-Vertauschungen führen über mehre 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
Und zwar so, dass der **größtmögliche Hamming-Abstand** auftritt

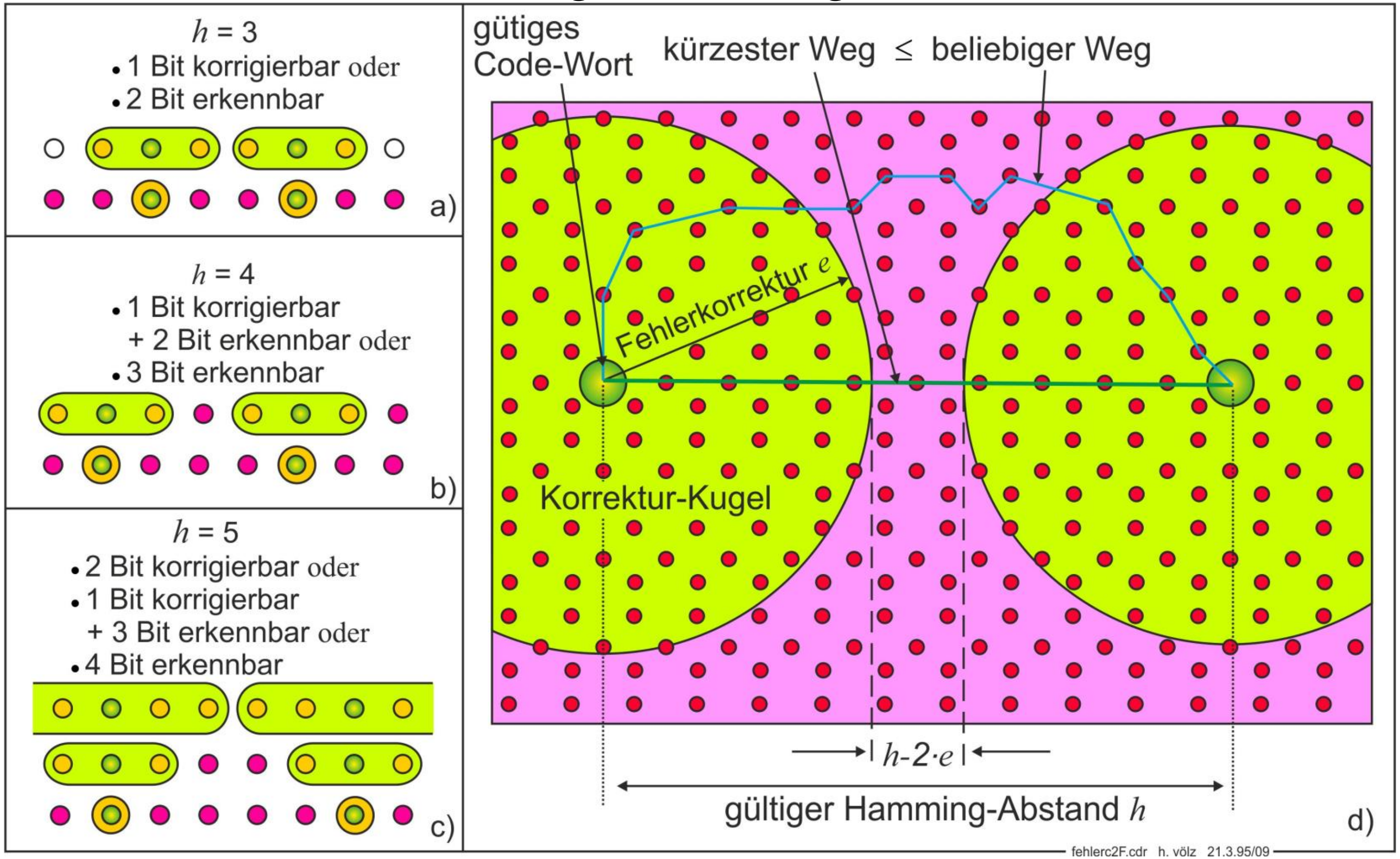


Hamming-Abstand und Grenzen der Fehlerbehandlung

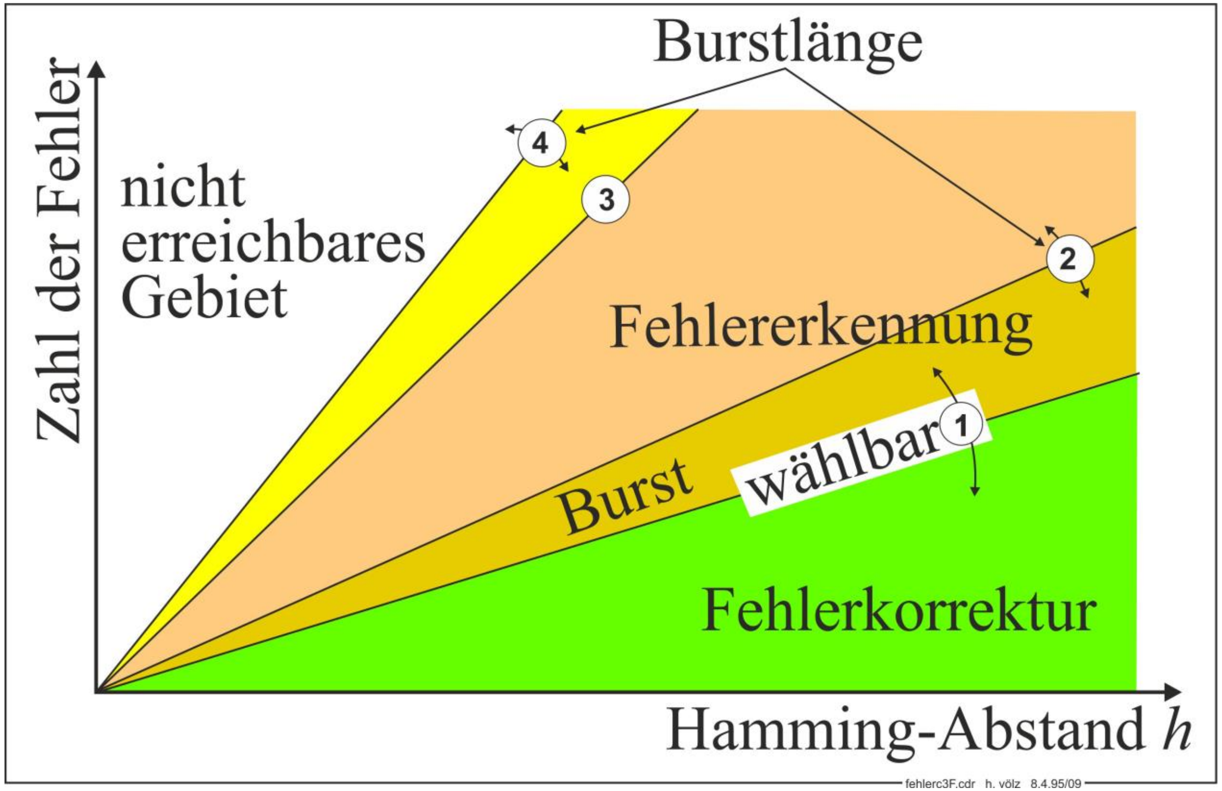
für einen bekannten Hamming-Abstand *h* lassen sich leicht Aussagen zur Leistungsfähigkeit angeben.
Beträgt *h* = 3 so sind ersichtlich 1-Bit-Fehler korrigierbar
oder wenn das nicht genutzt wird (ohne Korrektur) 2-Bit-Fehler erkennbar.
Falls ein 2-Bit- oder Mehr-Bit-Fehler auftrat, wird leider sogar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

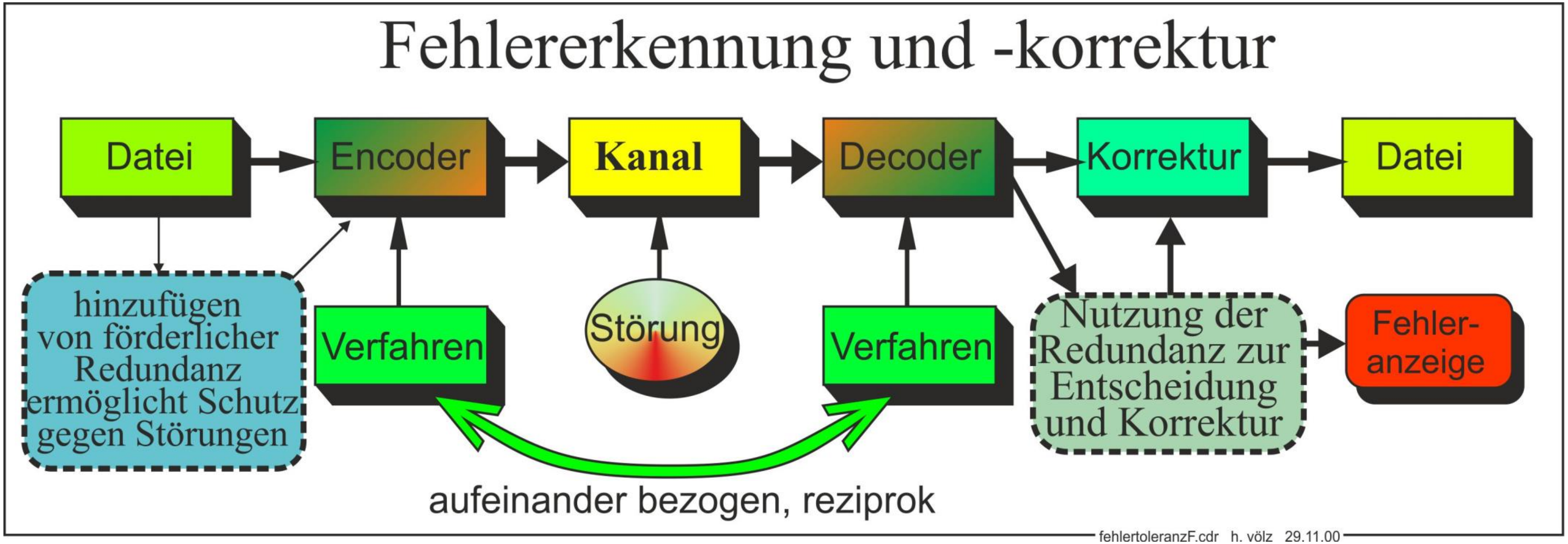


Abschätzungen



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 später behandelten Komprimierung.
Dies 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 gleich falsch).



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		

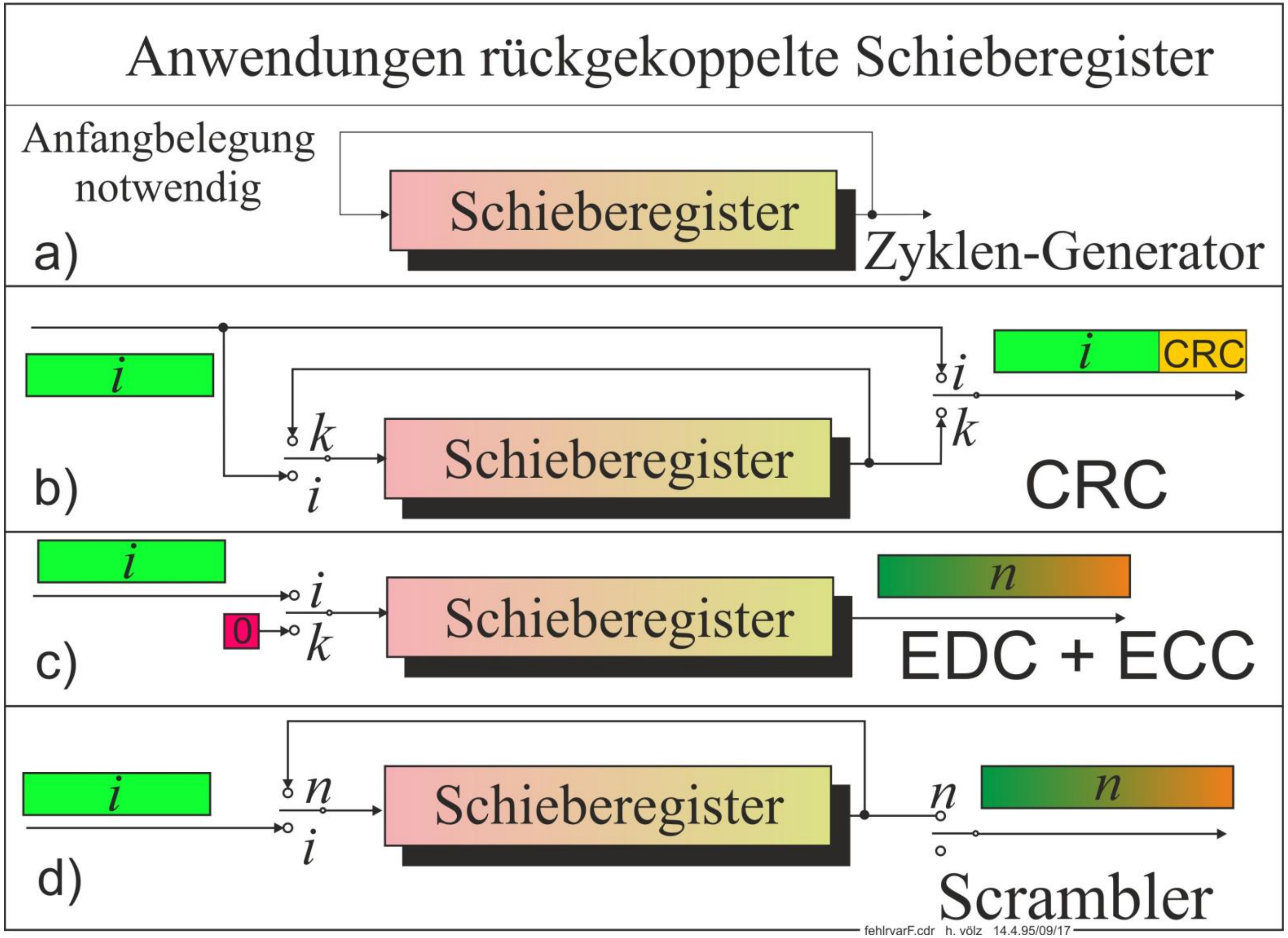
Kurzer Überblick

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

Namen oft nach den **Erfindern** benannt. Detail s. Buch „Das ist Information“ auch Literatur dort

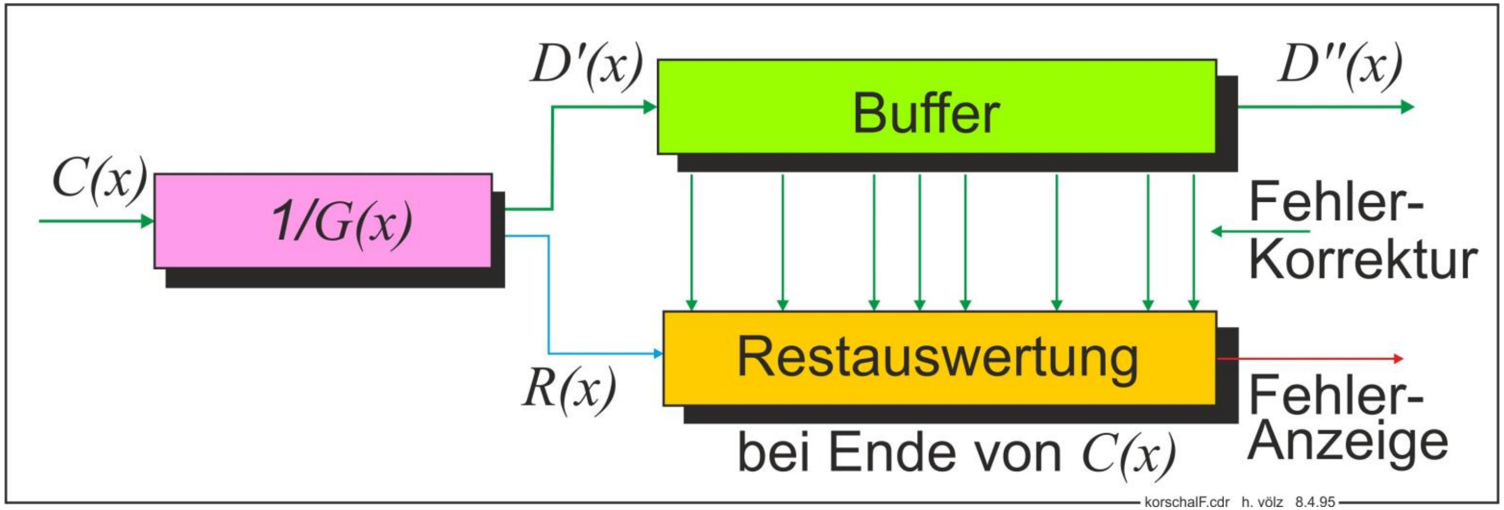
Häufig erfolgen die Untersuchungen zur Fehlerkorrektur mit den **Galois-Feldern** in der Matrizen-Methode. Denn sie ermöglicht auch, dass nicht-binäre, also höherwertige Variablen nutzbar sind.

Eine andere Variante ist die übersichtlichere ***Polynom-Methode***.



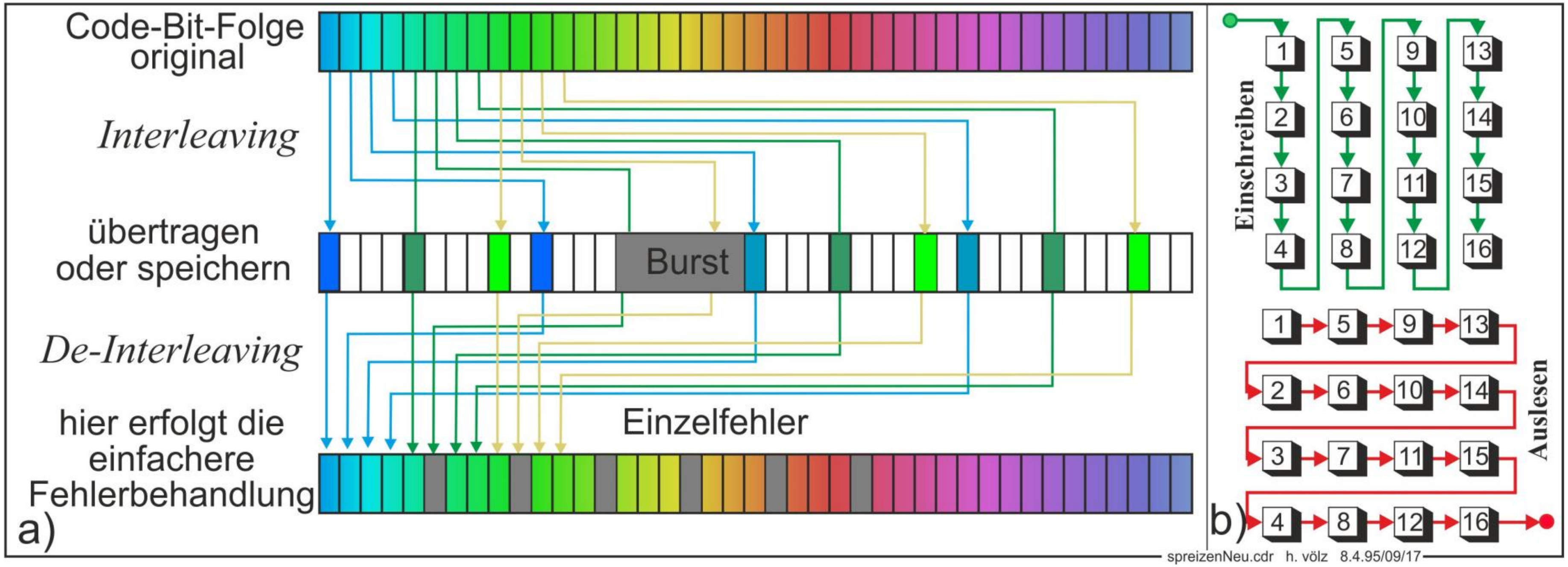
Genormte Polynome

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	$2 \cdot 10^{-10}$	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$



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.

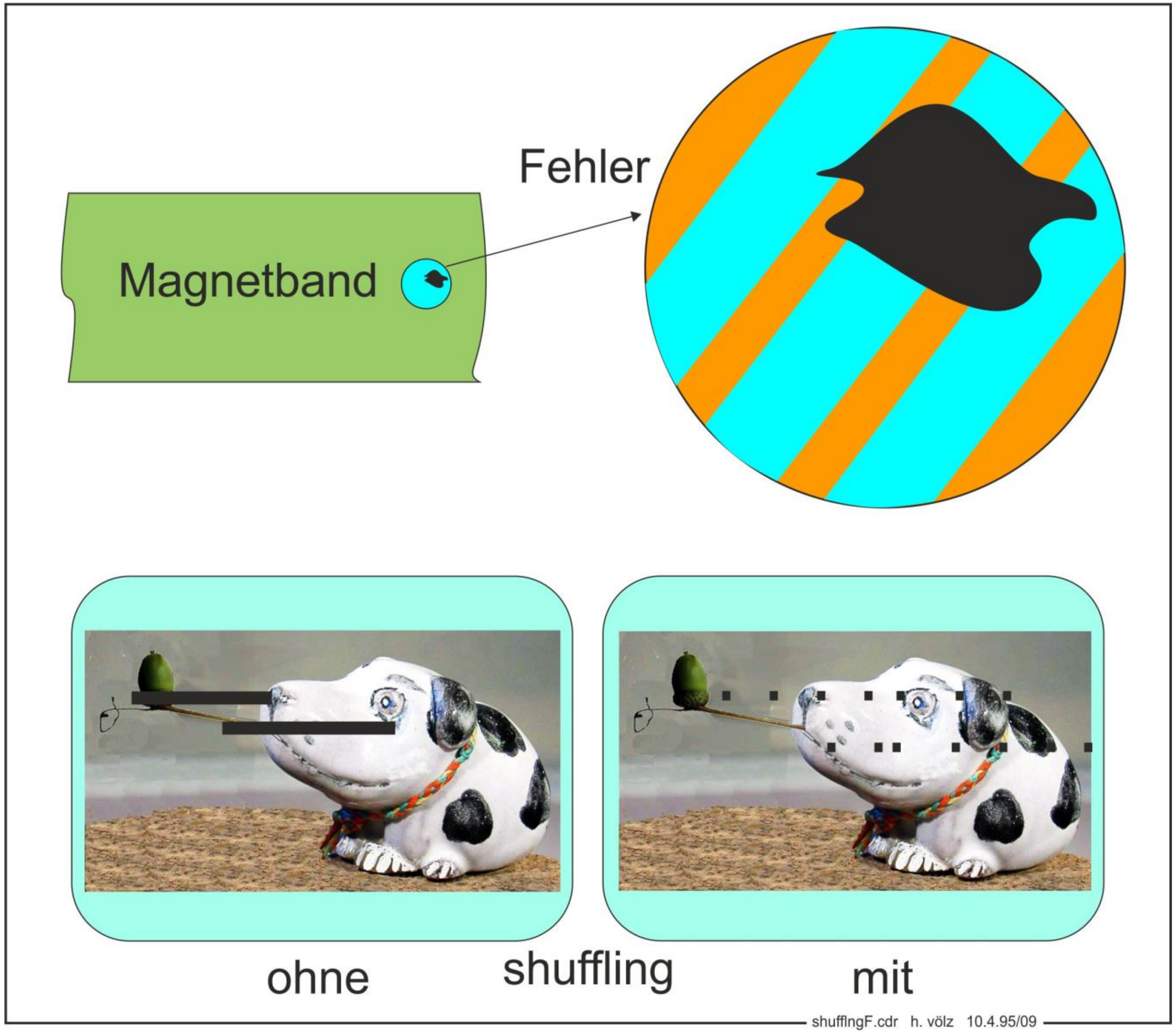
Spreizung



Einen Überblick zum Geschehen bei Codespreizung zeigt das Bild
Sie soll möglichst groß sein. R nimmt mit der Länge der zu übertragenden Wörter zu.
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.
Benachbarte Fehler lassen sich so zu Einzelfehlern in verschiedenen Blocks umwandeln.
Nach Korrektur erfolgt die Rücksortierung
Oft erfolgt dies mittels Matrix-Zusatzspeicher spaltenweise Einschreiben
und nach Korrektur durch zeilenweise lesen.

Erfolg beim Fernsehen

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

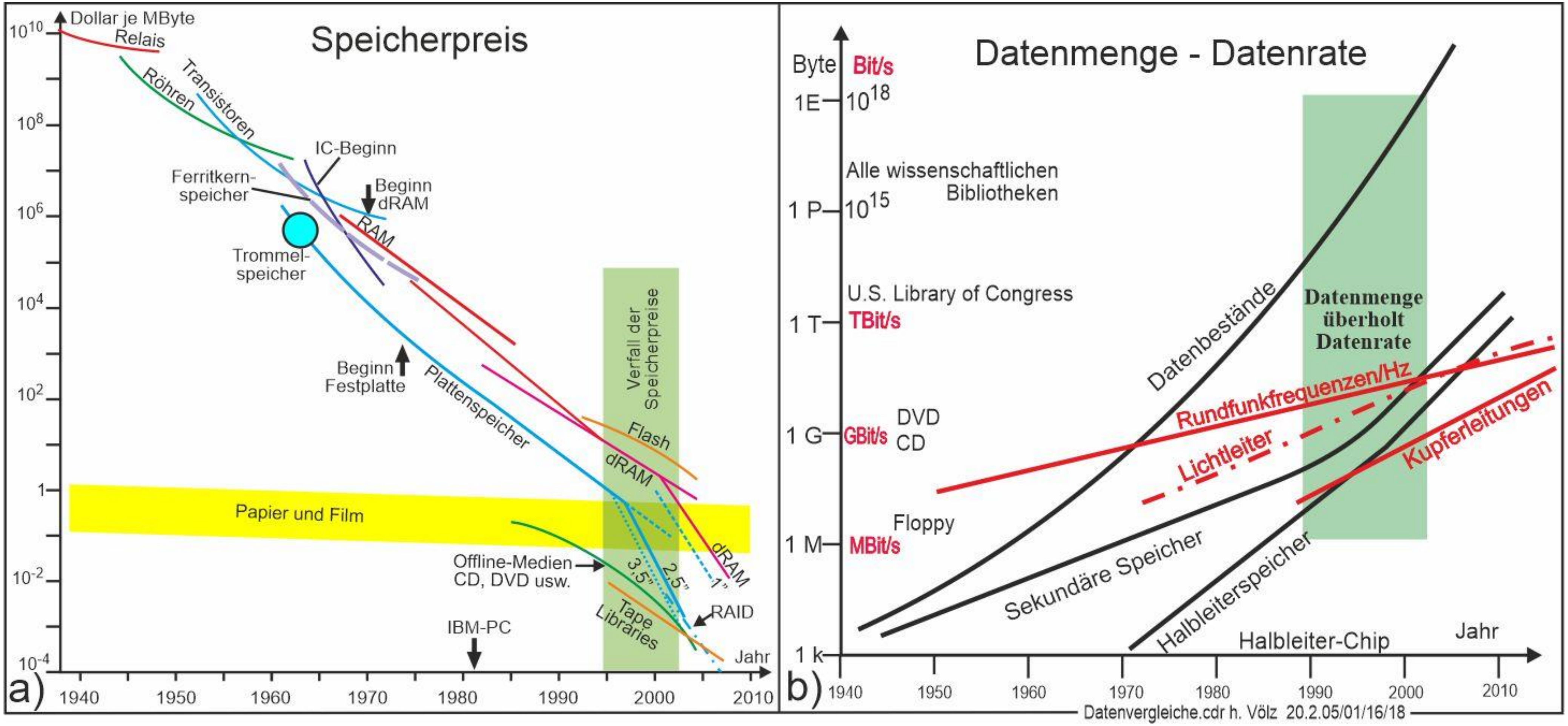


Komprimierungen

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 Auswege können die Künstliche Intelligenz und Datenkomprimierung bringen
Was big data zu leisten wird ist noch unklar.

Speicherpreis und Datenrate



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

Dies ist ein Text

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

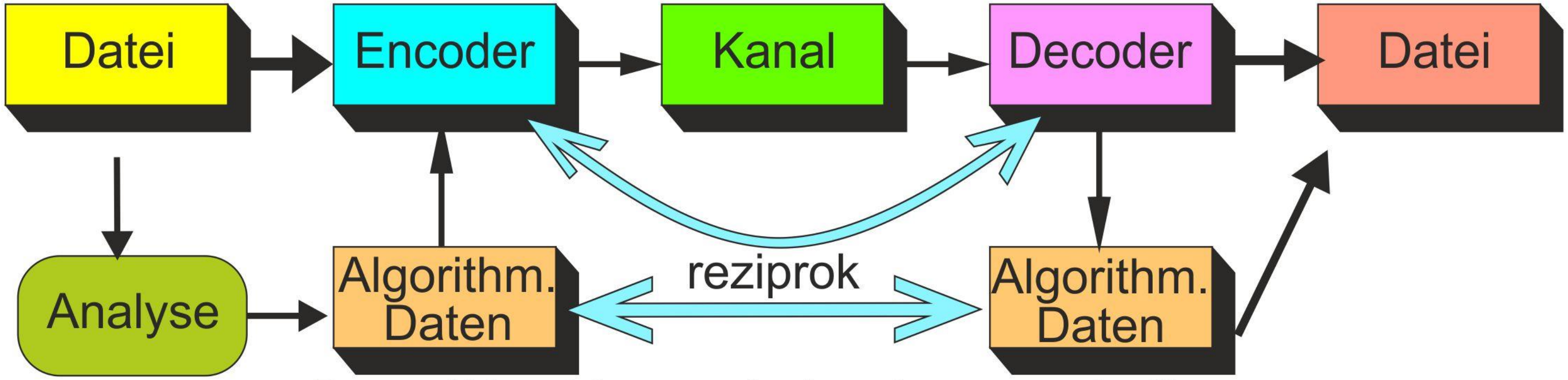
Irrelevanz, Redundanz haben beachtlichen Bezug zu Ockhams Rasiermesser, nicht Notweniges abschneiden dort auch Abstraktion und Reduktion.
Bei Fehlerbehandlung wird Redundanz in genau bestimmter Weise hinzugefügt wird
Bei der Komprimierung dagegen soviel wie möglich oder zulässig entfernt werden.
Dafür 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)

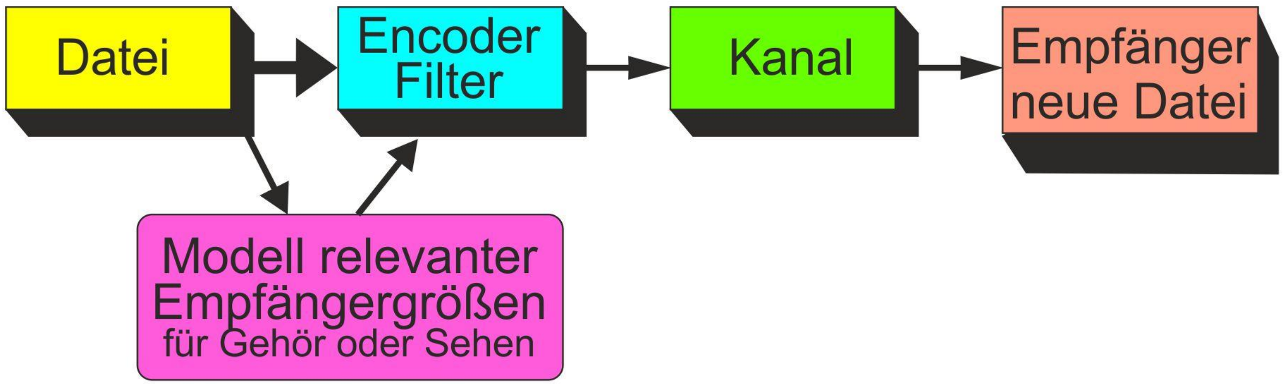


Daten/Algorithmen sind vorher zu schaffen,
evtl. bei der Übertragung zu ergänzen

a)

verlustbehaftete Komprimierung

nur empfänger-relevante Daten müssen berücksichtigt werden



beruht auf Untersuchungen

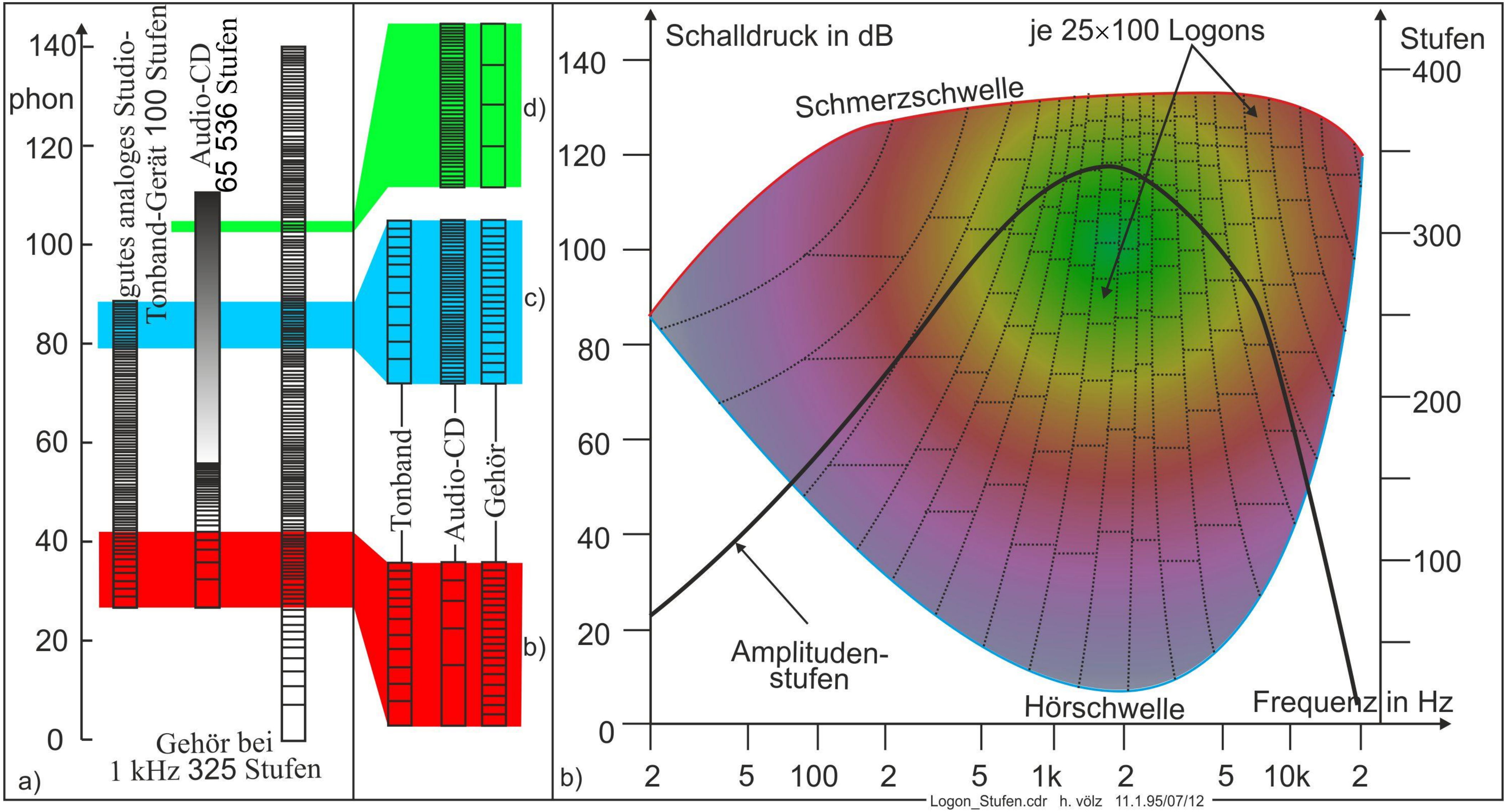
b)

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.

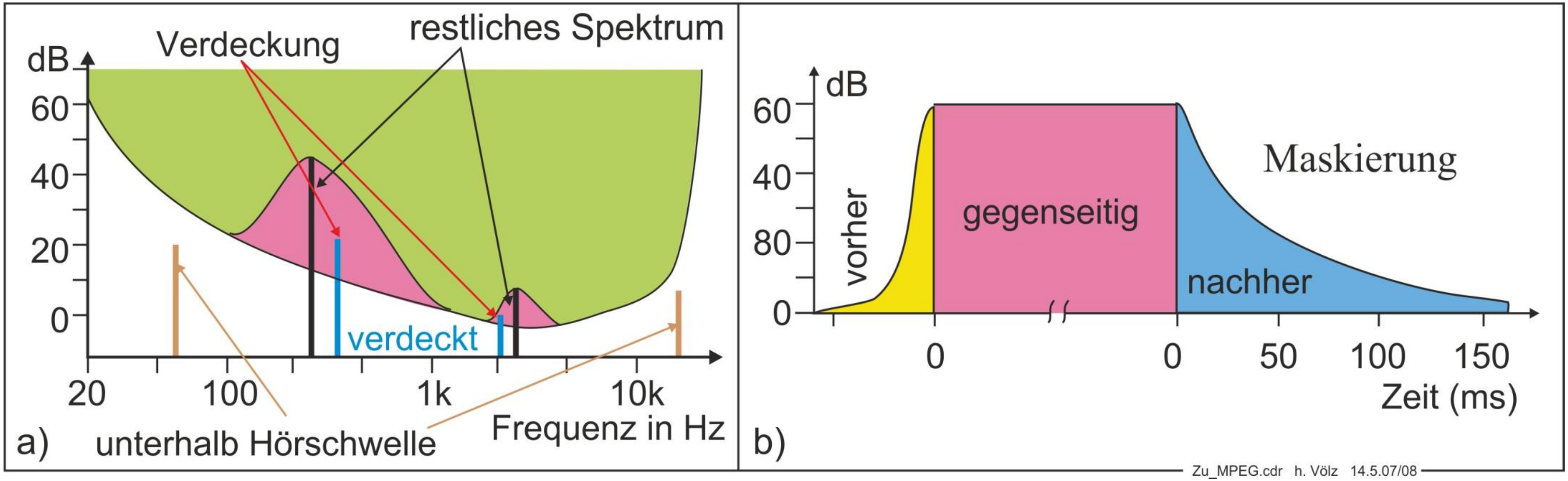
Bei großen Lautstärken können aber unnötig viele kleine Stufen zu einer zusammengefasst werden.
Anwendung u.a. bei MP3 (einstellbar s.u.) , bei Mobiltelefonen sogar nur **5** Lautstärkestufen.



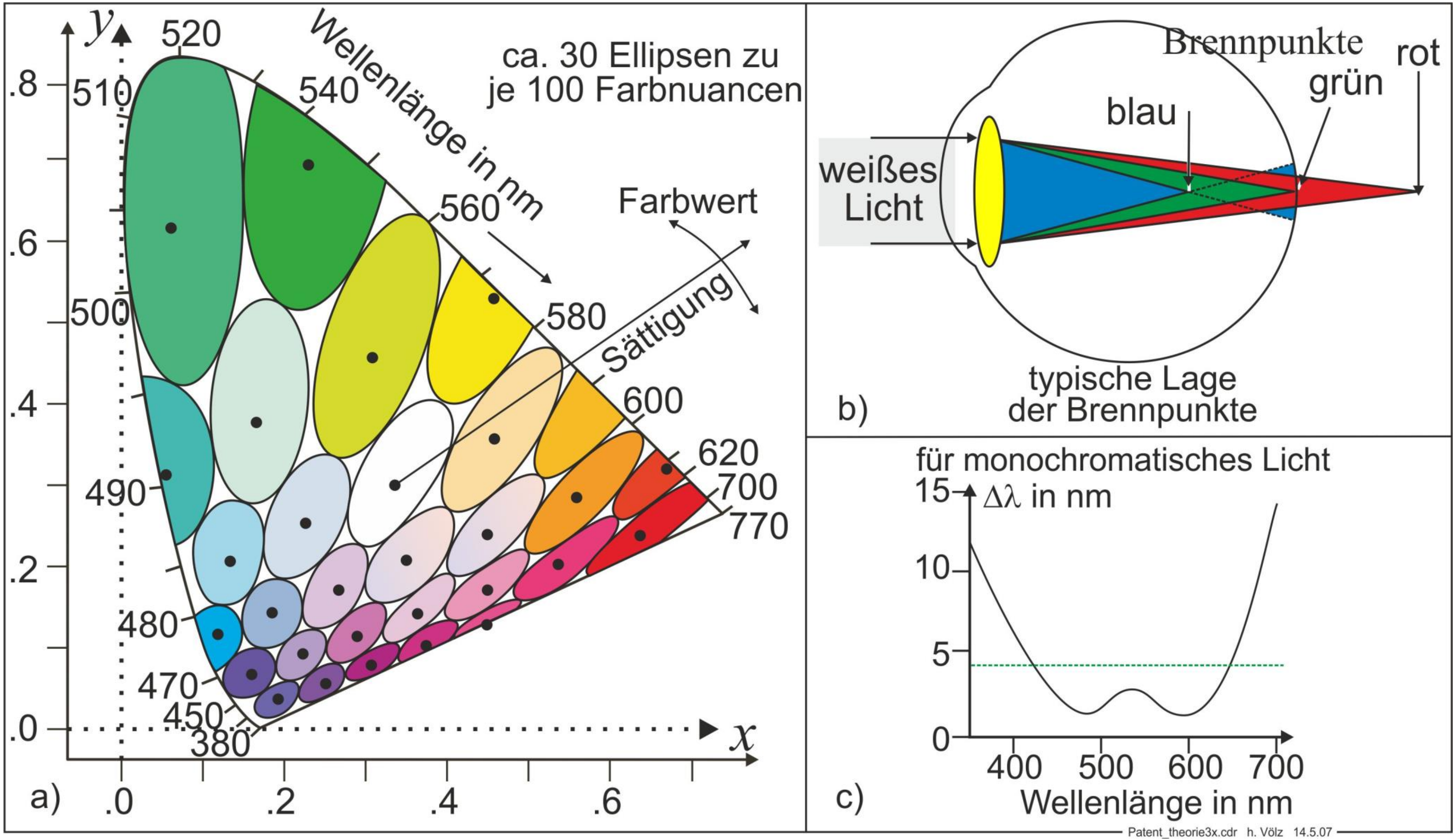
140
phon
120
100
80
60
40
20
0
gutes analoges Studio-Tonband-Gerät 100 Stufen
Audio-CD 65 536 Stufen
Gehör bei 1 kHz 325 Stufen
a)
d)
c)
b)
Tonband
Audio-CD
Gehör
Schalldruck in dB
je 25×100 Logons
Stufen
400
300
200
100
Schmerzschwelle
Hörschwelle
Amplituden-stufen
Frequenz in Hz
2
5
100
2
5
1k
2
5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

Weitere Varianten

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

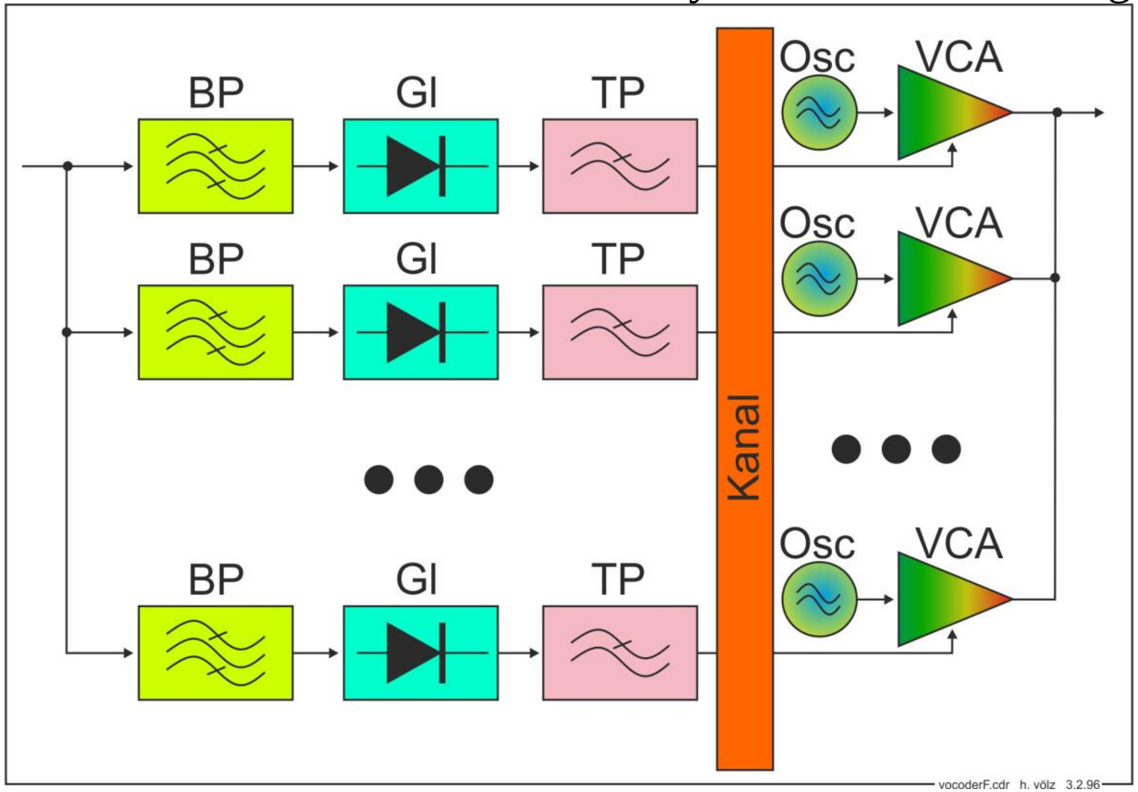


Sehen


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

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.

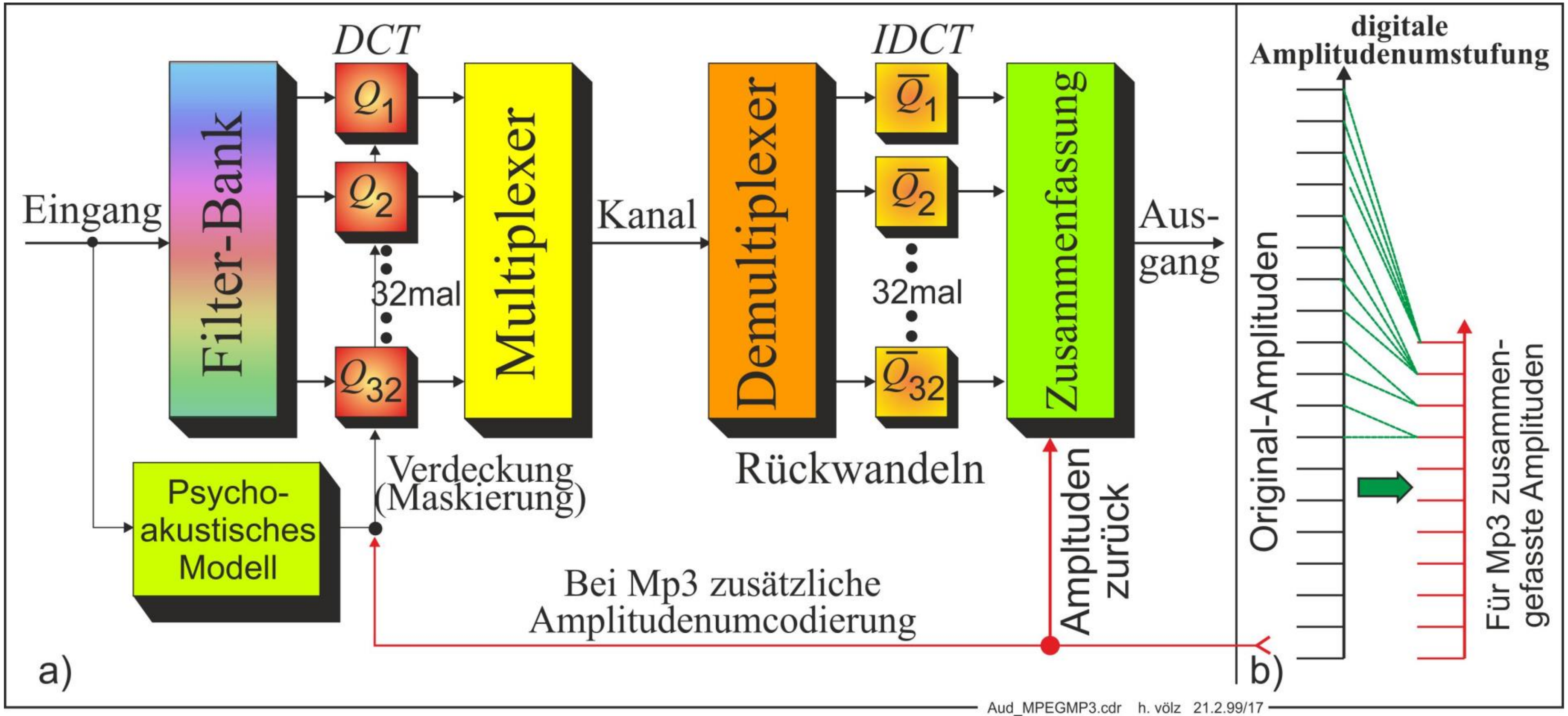


So wird vor dem Lautstärkanstieg den Pegel unhörbar langsam abgesenkt u. nachher wieder erhöht.
Für Speicher (Schallplatte, Heimtonbandgerät) wird das z. T. mit automatischen Dynamikregelungen ereicht
Dazu entstanden ab den 70er Jahren mehrere Dolby-Verfahren mit Com- und Expandern.
Prinzip ≈ reziproke Dynamikregelung

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

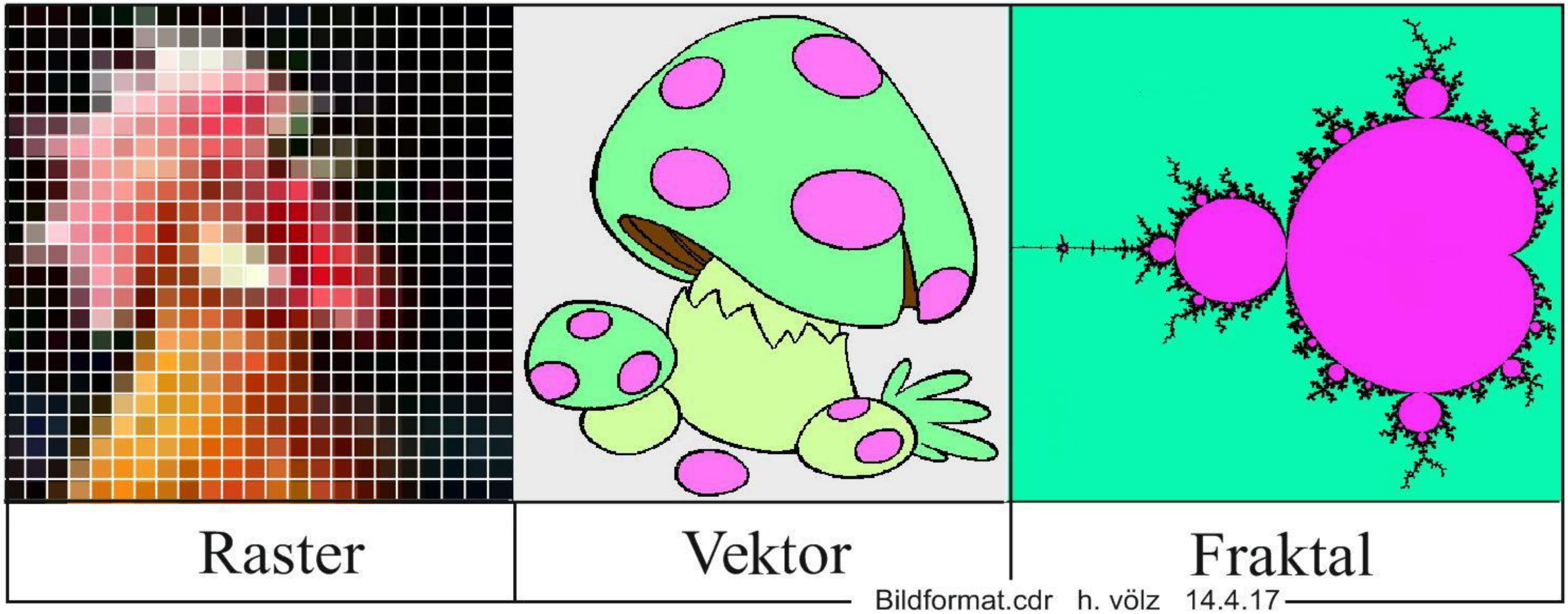
(moving picture expert group) entstand primär für Video ab Ende der 1980er Jahre.
Dabei waren auch Audio-Komprimierungen notwendig, die getrennt von Video als mp2 bezeichnet werden.
Sie nutzen vor allem die Verdeckungen durch laute Frequenzen und z. T. auch Maskierungen(s.o.)
So ergibt sich das Prinzip des Bildes, mit dem zunächst nur Verdichtungen um 1:4 möglich waren.
Mp3 ist eine Weiterentwicklung durch Zusammenfassung 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.



Aud_MPEGMP3.cdr h. völz 21.2.99/17

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 drei Formate des Bildes unterschieden



Videoformate

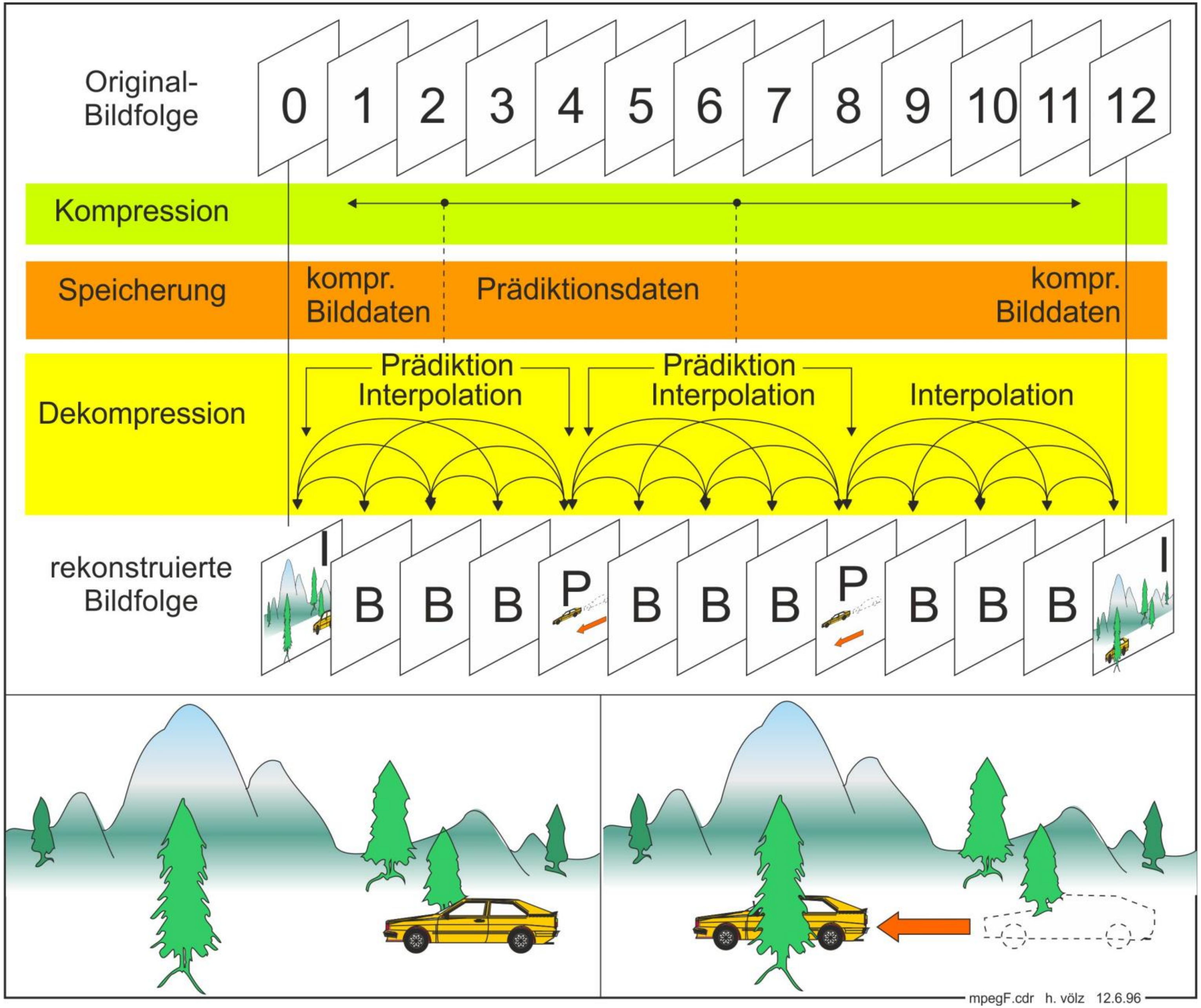
Für Videoformate ist jpg usw. ungeeignet: von Bild zu Bild ergeben such stark verändernde Artefarkte
Sie führen zu erheblich störenden Pixelrauschen. So entstand das mpeg-Format (moving peg).
Im folgenden Bild ist es vereinfacht dargestellt.
Eine aufeinander folgende Gruppe von meist 12 Bildern wird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

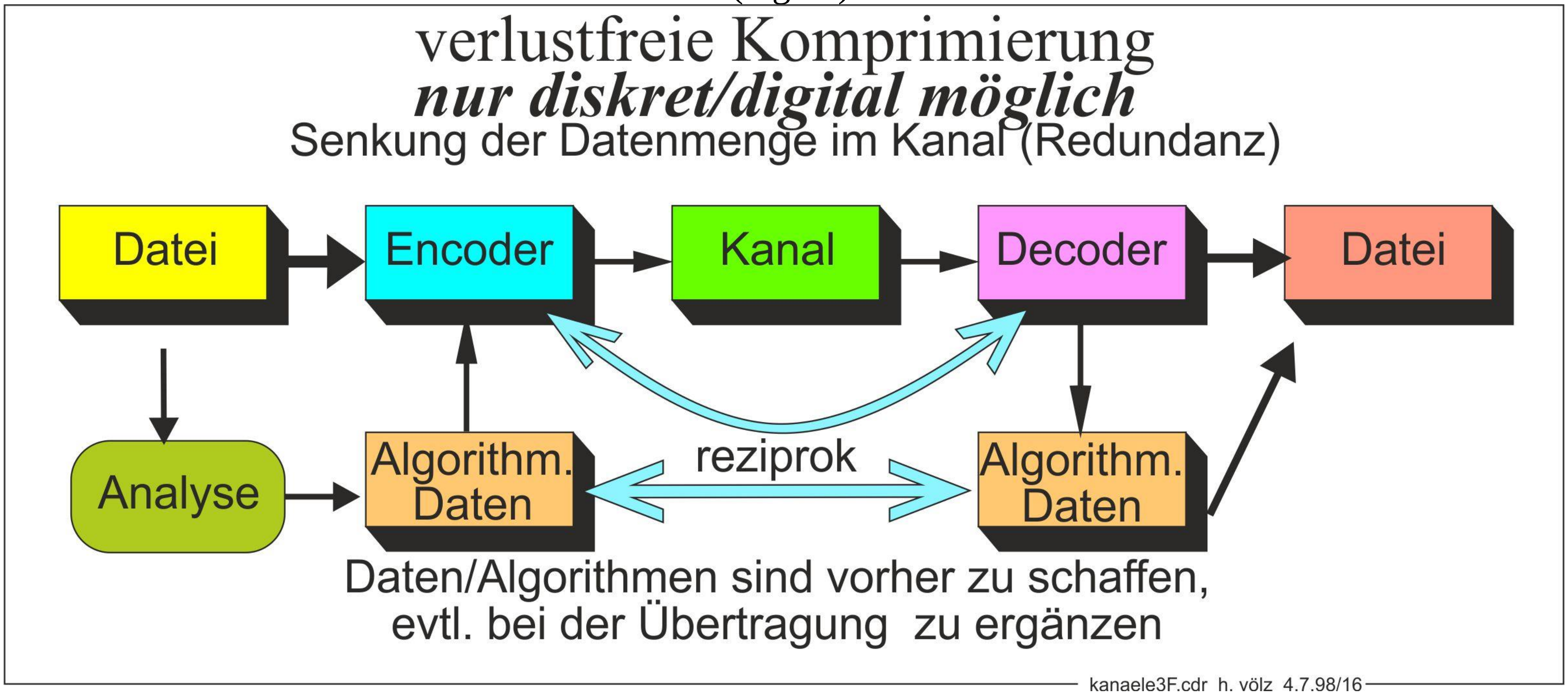
viele Details zur Videocodierung enthält [Str02] ab S. 188. im Buch sind mehrere Quellcodes abgedruckt.


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

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.

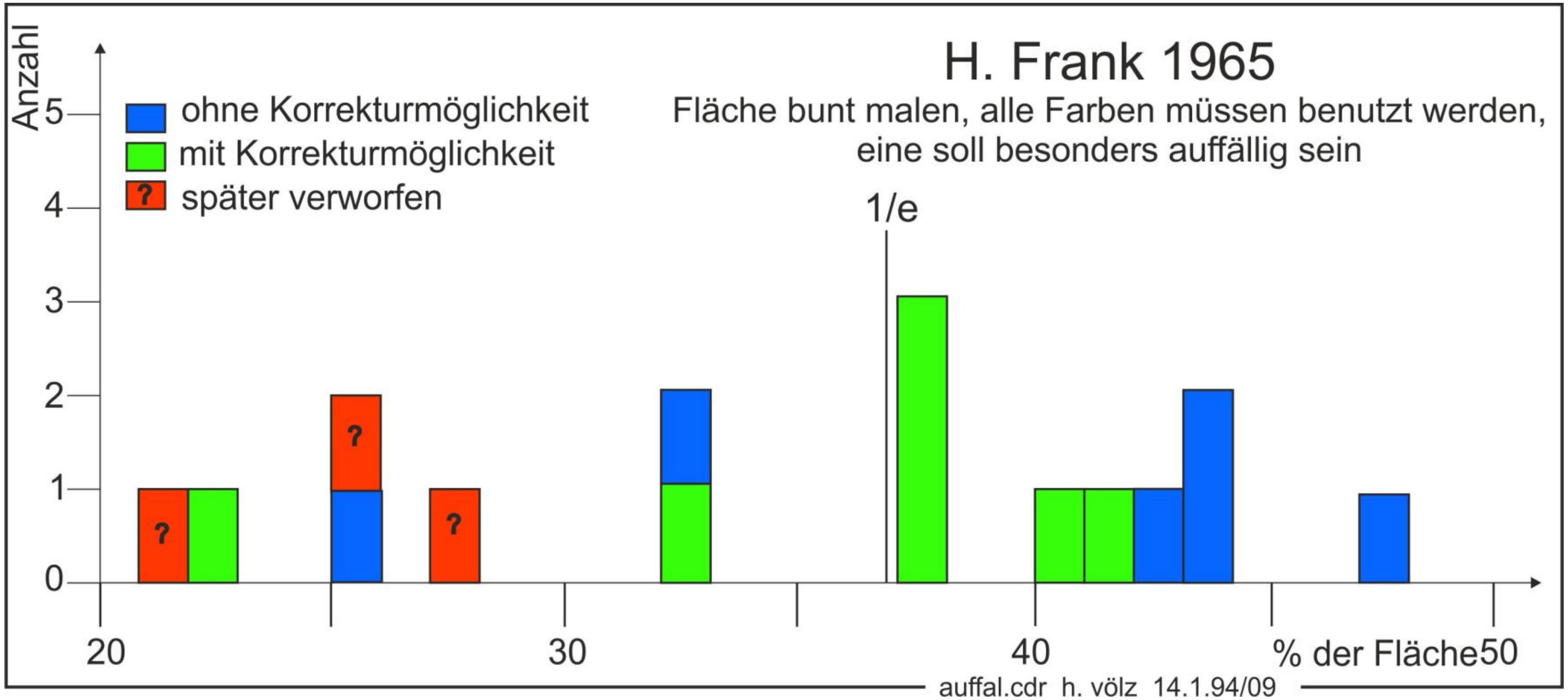


Kurze Zeittafel zur Komprimierung

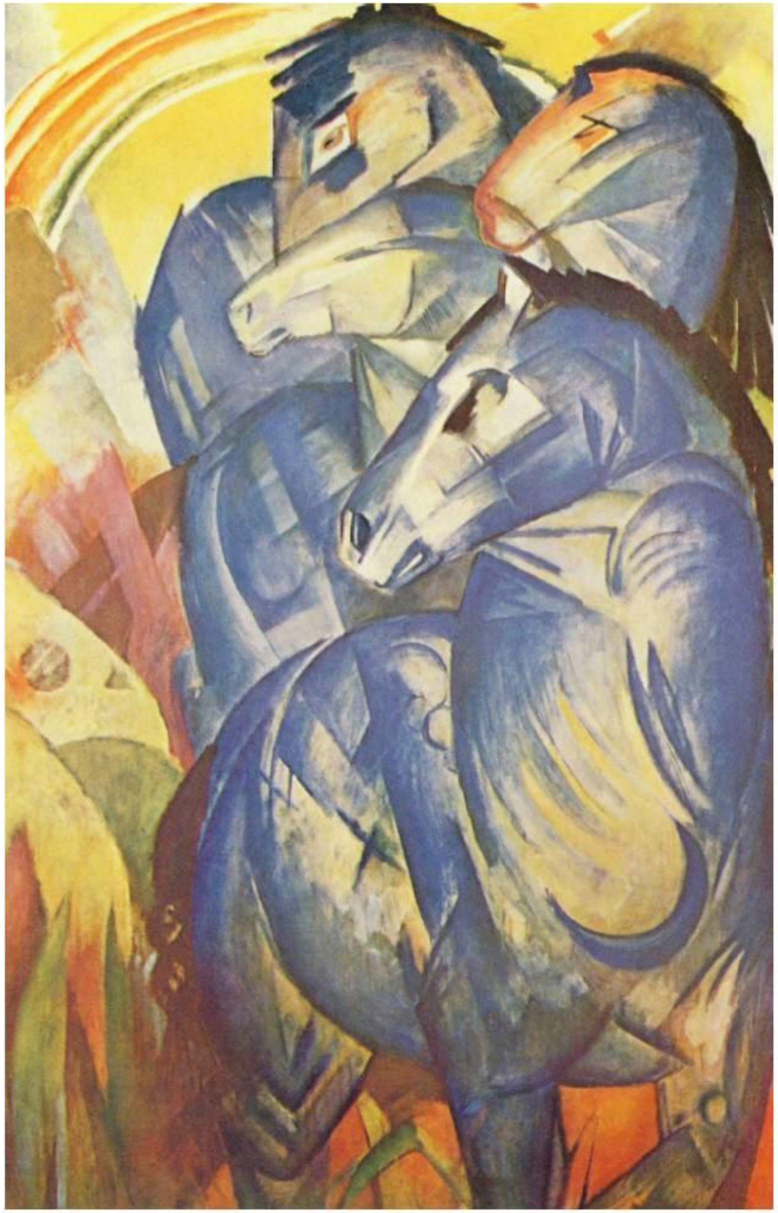
1949 Informationstheorie, Claude Shannon
1949 Shannon-Fano-Codierung
1952 Huffman-Codierung
1964 Konzept der Kolmogorow-Komplexität
1975 Integer Coding Scheme nach Elias
1975 Satz von Chatain
1977/78 Lempel-Ziv-Verfahren LZ77, LZ78
1979 Bereichs-Codierung als Implementierung der arithmetischen Codierung
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 -$p$ld(p) ein. Es liegt bei 1/e ≈ 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.

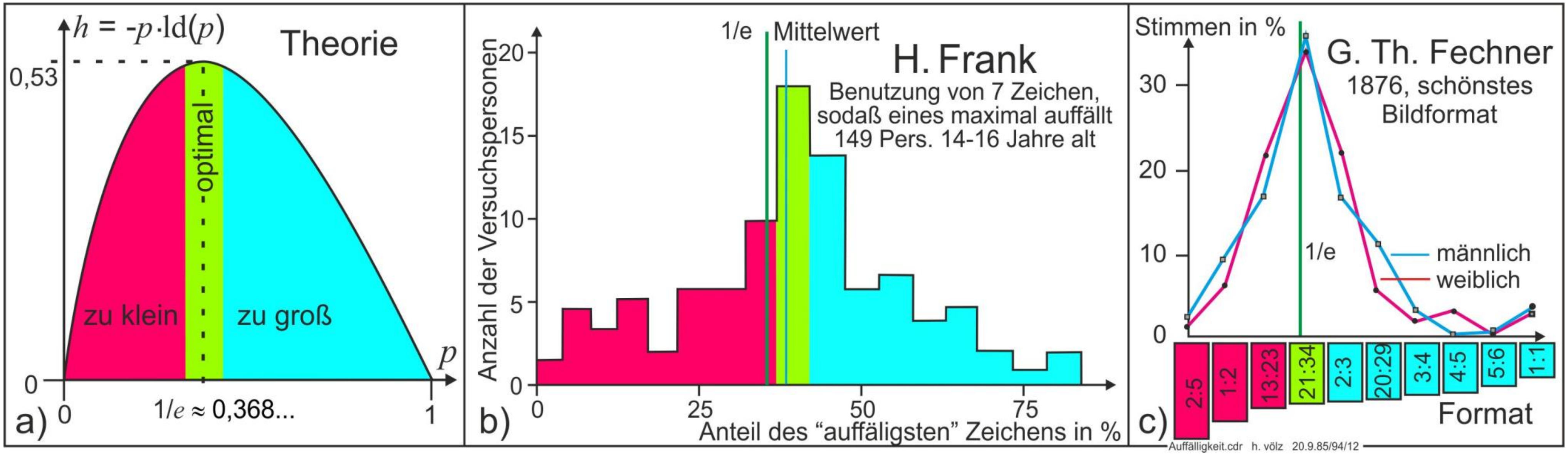
Hear the sl***e***dg***e***s with the b***e***lls, silver b***e***lls!
What a world of m***e***rrim***e***nt their m***e***lody foret***e***lls!

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.

Weiteres Frank-Experiment

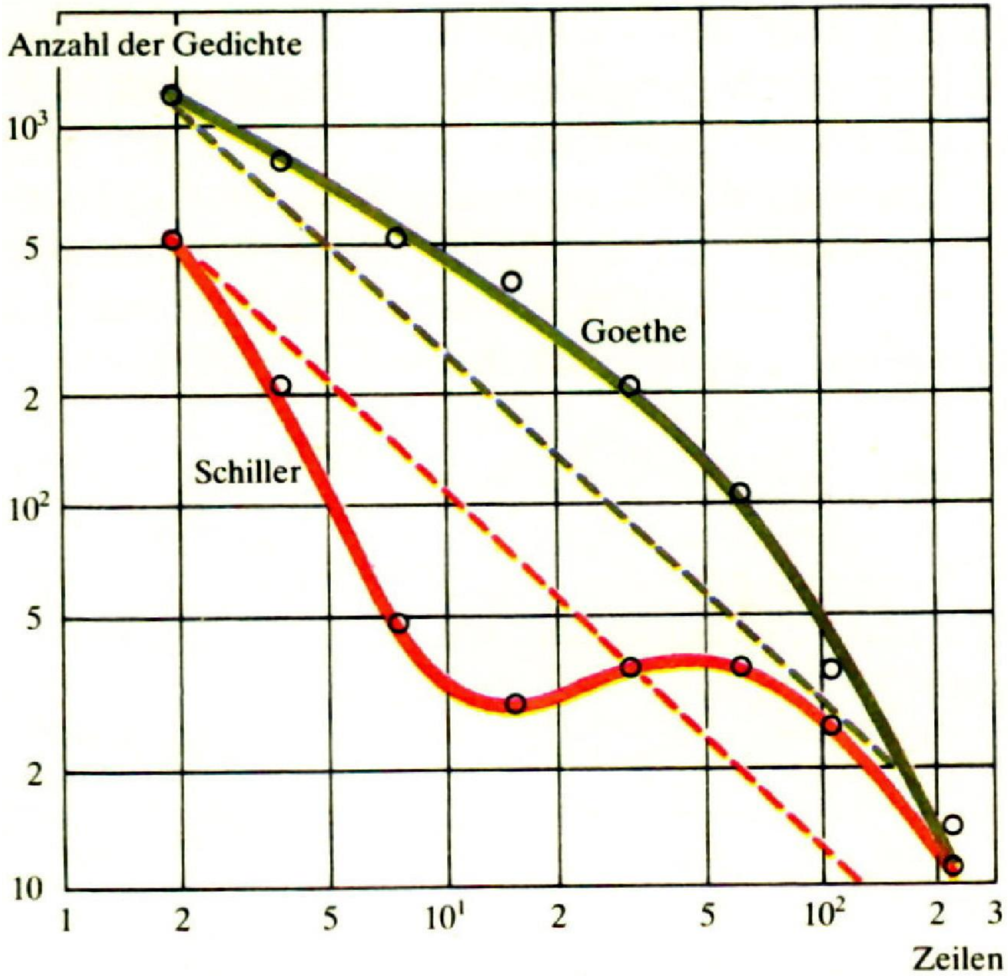
149 Personen soolten 7 Zeichen so verwenden, das eines besonders auffiel.
Dabei lag der Mittelwert mit 37 % besonders deutlicher bei der Auffälligkeit.
Das Bild zeigt dazu den Vergleich vom Verlauf der Kurve $h = -p \cdot \mathrm{ld}(p)$ (a) und alter Fechner-Untersuchung



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

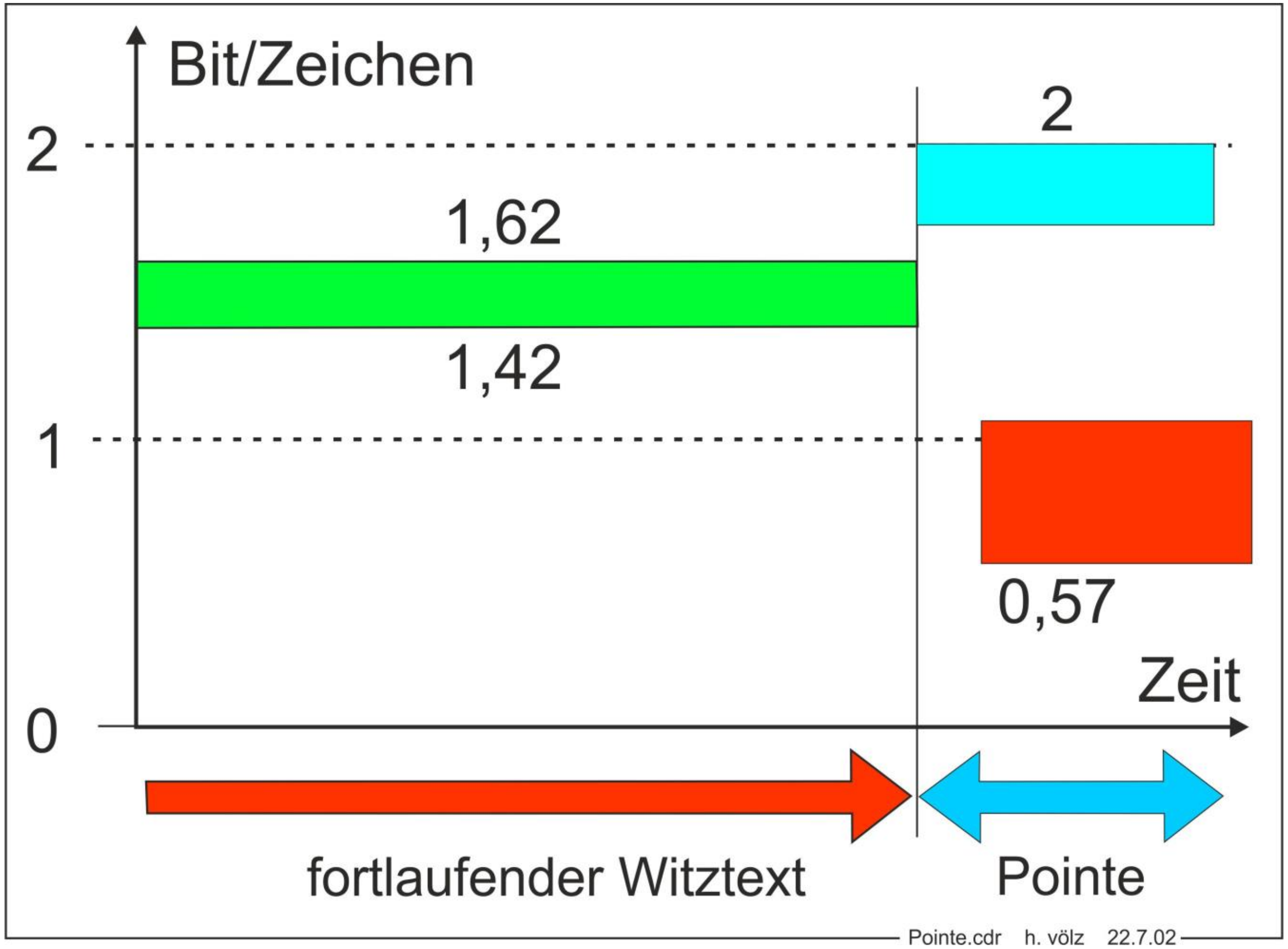
Gedichtlängen bei Goethe und Schiller

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.



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.

Zusammenfassung zur S-Information

- Das primäre Ziel ist die **Nachrichten-** (teilweise auch **Speicher-**) Technik.
- Ihre Entropie und Kanalkapazität bestimmen 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.