Verfahren der Kompression

verlustbehaftet	**verlustfrei**
immer notwendig, auf den Empfänger bezogen, beim Menschen z.B.: Grenzen der Wahrnehmung, Möglichkeit des Wiedererkennens **Grenzen** durch Kenntnisse bzgl. Empfänger	gilt nur für digitale Daten, nur sie lassen sich exakt wiederherstellen, es lassen sich alle Methoden der Datenverarbeitung nutzen; also Algorithmen und gespeicherte Daten. da immer endliche Datenmenge, gilt nicht Satz von Chaitin **Grenzen** vor allem Analysezeit für die Struktur der Datei

beide Verfahren lassen sich **verbessern** mittels:

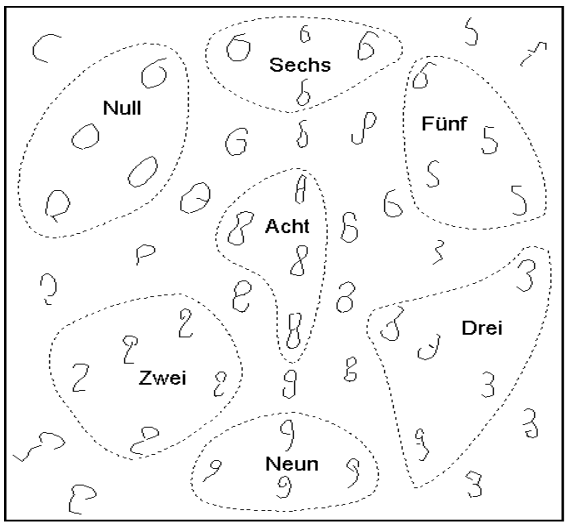
größerer Speicherkapaität für Zusatzdaten und
kompliziertere Verarbeitungsalgorithmen

kompression2.cdr h. völz 30.11.98

(vorübergehende) Senkung der Informationsmenge

Rendundanz	Relevanz
speziell: in Bezug auf Shannon: Verhältnis von Codeaufwand und Entropie **allgemein**: was aus einer Datei vorübergehend so entfernt werden kann, daß sie später wieder in ihrem Originalzustand vesetzt werden kann **angewendet** darf dabei alles werden, inbesondere Algorithmen die Strukturen erkennen Daten die bekannt bzw. gespeichert sind **förderliche** Redundanz schafft Schutz gegenüber Störungen und Angriffe Fehlererkennung, Fehlerkorrektur, Kryptographie, Steganographie **Entfernung** von Redundanz führt zur **verlustfreien** Kompression Faktor ist abhängig von Datei **und** Methode	**ergibt** sich: aus den **Eigenschaften des Empfängers** um sie zu nutzen sind oft vielfältige Signaltranformationen erforderlich Beim **Menschen** sind z. B. irrelevant: Schall > 20 kHz Licht mit λ > 800 nm und λ < 400 nm Bildwechsel > 100Hz *Hexcode* für x86 kann meist nicht für Motorola genutzt werden **Entfernung** von Irrelevanz führt zur **verlustbehafteten** Kompression wesentlich ist hierbei der **Empfänger**

redundanz.cdr h. völz 30.11.98

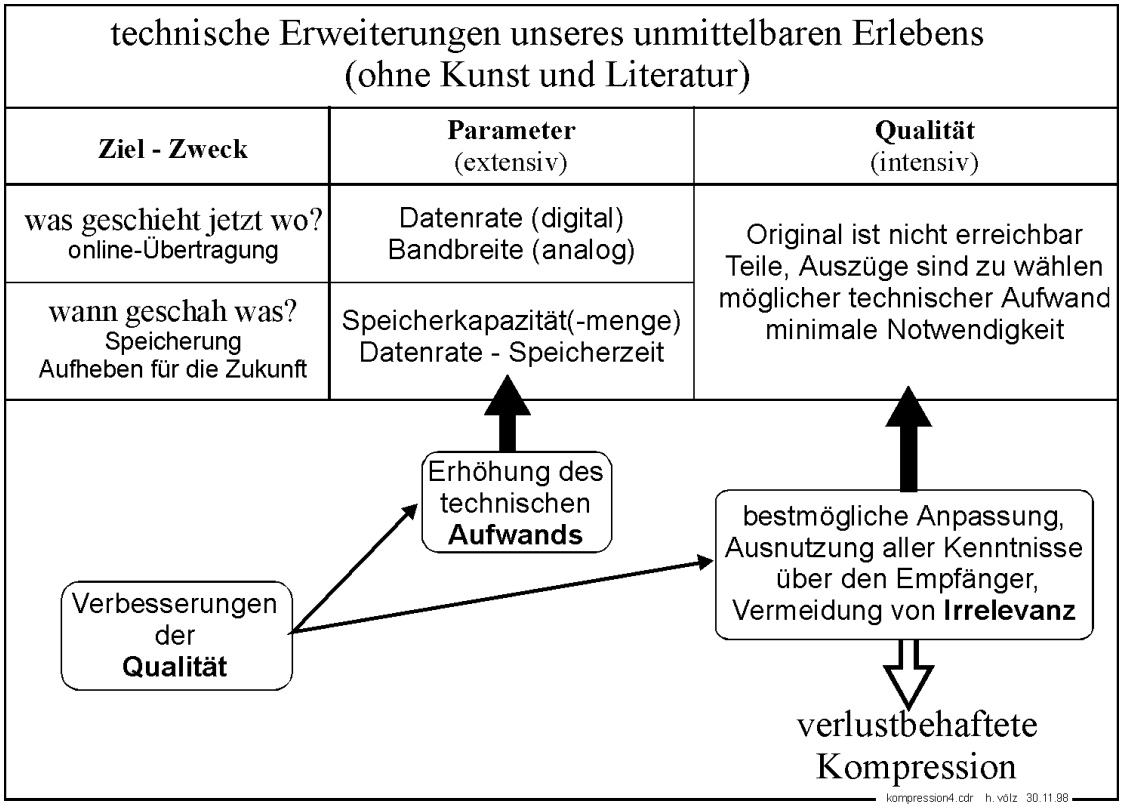


Irrelevanz

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

Redundanz

*D**s *st **n T*xt*
Dies ist ein Text
Dies ist ein Text



Lauflängen-Codierung

CCCCCCAABBBBAAAAEE _ 6C2A4B4A2E Pixel- und Zähl-Byte unterscheiden

Pointer-Verfahren

ABRABRIKADABRA Ab 4. Buchstaben ABR _ Verweis <1,3>. total _ **ABR<1,3>IKAD<1,3>A**

Problem Verweislänge, daher meist gleitend

Code-Erweiterung

J. Ziv u. A. Lempel 1977, Ab 1984 Welch (LZW, PKZIP) nutzbar: 256 ASCII-Zeichen + zusätzliche für häufige Zeichenketten:

Komprimieren **drei Vorgänge**:

1) Die Ausgabe in die komprimierte Datei
2) Die Erweiterung des Zeichensatzes
3) Die Verwendung des erweiterten Zeichensatzes

wieder diese Kinder

"**w**" wird ausgegeben, "**wi**" erhält die Nr. 256
"**i**" wird ausgegeben, "**ie**" erhält die Nr. 257
"**e**" wird ausgegeben, "**ed**" erhält die Nr. 258 usw.

Bei "**diese**" existiert bereits "**ie**" wird ausgegeben und "**ies**" erhält 264 257:ie 259:de 261:r ... 270:nd 271: der **wieder d"257"se Kin"259"261"**

Entropie-Codierung

Morse, wissenschaftliche Begründung Shannon (1940/48), optimale Lösung 1952 durch Huffman.

Testlauf bestimmt Häufigkeit, 2. Lauf codiert: statisch (fest gegeben), dynamisch (je Datei) adaptiv z.B. je 128 Zeichen neu.

Vergleich

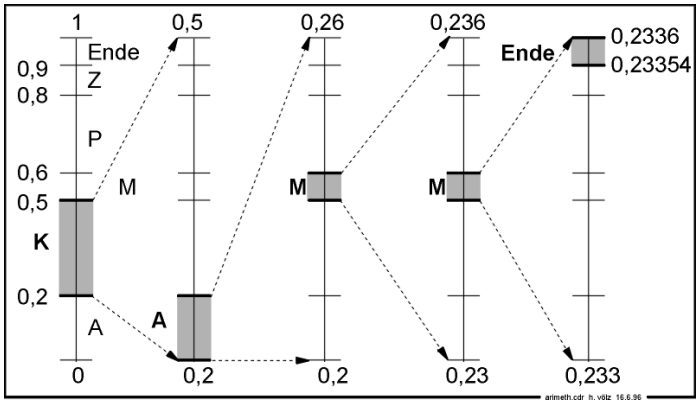
Übergänge der Codierung	
Run-Length	Byte -> Zähler und Byte
Pointer	Mehrbyte -> Pointer
Zip-Lempel	Mehrbyte -> 9- bis 12-Bit-Wort
Huffman	Byte -> rekurrenter Bit-Code
Arithmetisch	Ganze Datei -> lange Zahl
Hilberg	Worte -> Pointer
Redundanzen betreffen:	
Run-Length	Unmittelbare Wiederholung einzelner Zeichen
Pointer und Zip-Lempel	Wiederholung von Zeichenfolgen in einer begrenzten Umgebung
Huffman	Statistik der Zeichen
Arithmetisch	Statistik der Zeichen
Hilberg	Vektorstatistik bei Wörtern

Arithmetische Codierung P. Elias um 1985

Zeichen	Häufigkeit	Intervall
a	0,2	0-0,2
e	0,3	0,2-0,5
i	0,1	0,5-0,6
o	0,2	0,6-0,8
u	0,1	0,8-0,9
!	0,1	0,9-1,0

"**eaii**" fortlaufende Einschränkung des Intervalls:

Beginn	[0, 1)
e	[0.2, 0.5)
a	[0.2, 0.26)
i	[0.23, 0.236)
i	[0.233, 0.2336)
!	[0.23354,0.2336) Ausgabe der Zahl 0.23358



Zipf und Entropie

Eine oft brauchbarere Näherung für das Zipfsche Gesetz lautet

$$p(r) \approx \frac{1}{r \cdot \ln(N)} \text{ für } r > 1$$

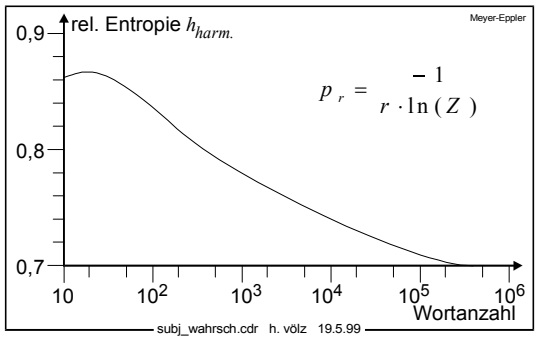
Darin bedeuten r den Rang; N Anzahl der Klassen und p die Wahrscheinlichkeit. Hieraus leitet u.a. Meyer-Eppler (S.112) die Entropie

$$H \approx \frac{\mathrm{ld}(N)}{2} + \mathrm{ld}(\ln(N))$$

für die relative Entropie folgt dann

$$h = \frac{H}{\mathrm{ld}(N)} \approx \frac{ld(\ln(N))}{\mathrm{ld}(N)} + \frac{1}{2}$$

Diesen Verlauf zeigt das Bild



Mit der ursprünglichen Formel von

$$p(r) \approx \frac{1}{r \cdot \ln(1{,}78 \cdot R)}$$

berechnet dagegen Schroeder (Fraktale usw. Spektrum, 41) folgende ähnliche Entropie:

$$H \approx {}^1/_2 \cdot \mathrm{ld}\,(2 \cdot N \cdot \ln(2 \cdot \mathrm{N})) \quad \text{in Bit/Wort}$$

Bei 12 000 verschiedenen Wörtern folgt $H \approx 9$, bei 30 0000 $H \approx 11{,}5$ Bit/Wort.

Auch **Voss** liefert hierzu einen Beitrag. Er geht von N willkürlich festgelegten Klassen aus und gewinnt aus seinem allgemeinem Gesetz für die relative Entropie aus praktischen Beispielen für typische Fälle bei Sprachen Werte zwischen 0,83 und 0,88. Für soziologische Systeme erhält er Werte zwischen 0,8 und 0,9.

Vergleich

Übergänge der Codierung

Run-Length	Byte -> Zähler und Byte
Pointer	Mehrbyte -> Pointer
Zip-Lempel	Mehrbyte -> 9- bis 12-Bit-Wort
Huffman	Byte -> rekurrenter Bit-Code
Arithmetisch	Ganze Datei -> lange Zahl
Hilberg	Worte -> Pointer

Redundanzen betreffen:

Run-Length	Unmittelbare Wiederholung einzelner Zeichen
Pointer und Zip-Lempel	Wiederholung von Zeichenfolgen in einer begrenzten Umgebung
Huffman	Statistik der Zeichen
Arithmetisch	Statistik der Zeichen
Hilberg	Vektorstatistik bei Wörtern

Hilberg-Codierung

Jedes Wort zwei Code-Symbole x_i, y_i und Pointer z_i

y_1x_1 Wort 1 z_1	y_2x_2 Wort 2 z_2	y_3 x_3 Wort 3
0 0 Rotkäppchen 1	0 0 Rotkäppchen 1	0 0 den Jäger
0 1 der Wolf 1	0 1 der Wolf 1	0 1 die Großmutter
1 0 trifft 0	1 0 trifft 0	1 0 den Hänsel
1 1 erkennt 0	1 1 erkennt 0	1 1 die Gretel

Start-Code aus y_1x_1 begonnen. Ausgabe Pointer z_1. Zu x_2 ergänzt mit neuen Code y_2 = neues Wort

Gemäß Tabelle folgende 16 Sätze möglich: genügen also 4 Bit.

$y_1x_1x_2x_3$	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

es gibt:

- eine universelle Turingmaschine M
- ein Programm p
- ein auf dem Speicheband stehende Inschrift i

sie erzeugen die Folge

$$s = M(p, i)$$

Es seien alle Programme p_i bekannt,
die auf leerem Speicherband s erzeugen
Hierunter gibt es ein kürzestes p_k
seine Länge

$$L_M(s)$$

ist die Kolmogorow-Komplexität bezüglich M

Es gibt eine zweite universelle Turingmaschinen U
Dafür gilt

$$L_M(s) \le L_U(s) + K_{M,U}$$

$K_{M,\,U}$ ist die Länge des Simulationsprogramms von M auf U

Beispiele

für **unendliche** Folgen

`333333333...`
wiederholen und anhängen von 3

`01010101010101`
wiederholen und anhängen von `01`

`01001000100001000001`
Startkette $s_0 = 01$
wiederholen $s_0 = 0$ & s_0 : $s_{n+1} = s_n$ & s_0

`2 3 5 7 11 13 17`
Primzahlfolge

`1 4 7 2 5 0 3 6 1 4 7 2`
Start 1: Addition von 3: Mod 8

Satz von Chaitin

Es lassen sich endlich viele Ketten verkürzen, aber unendlich viele nicht

Dies gilt aber nicht für **endlichlange** Ketten, also für alle praktischen Anwendungen

kolmo komplodr h.völz 30.11.98

Berechnung des Codierungsgewinns beim Zusammenfassen

H	alte Entropie
m	alte Zeichenzahl
s	alte Dateigröße H·m
a	Anzahl des Symbols a
b	Anzahl des Symbols b
ab	Anzahl der möglichen Zusammenfassungen von a und b
n	= m - ab Neue Zeichenzahl

Entropie

$$H' = \frac{m}{n}\left(H - \mathrm{ld}(m) + \mathrm{ld}(n)\right) \quad - \frac{ab}{n}\cdot \mathrm{ld}\left(\frac{ab}{n}\right)$$
$$+ \frac{a}{n}\cdot \mathrm{ld}\left(\frac{a}{n}\right) \qquad - \frac{a-ab}{n}\cdot \mathrm{ld}\left(\frac{a-ab}{n}\right)$$
$$+ \frac{b}{n}\cdot \mathrm{ld}\left(\frac{b}{n}\right) \qquad - \frac{b-ab}{n}\cdot \mathrm{ld}\left(\frac{b-ab}{n}\right)$$

Neue Dateigröße s‘ = H‘·n

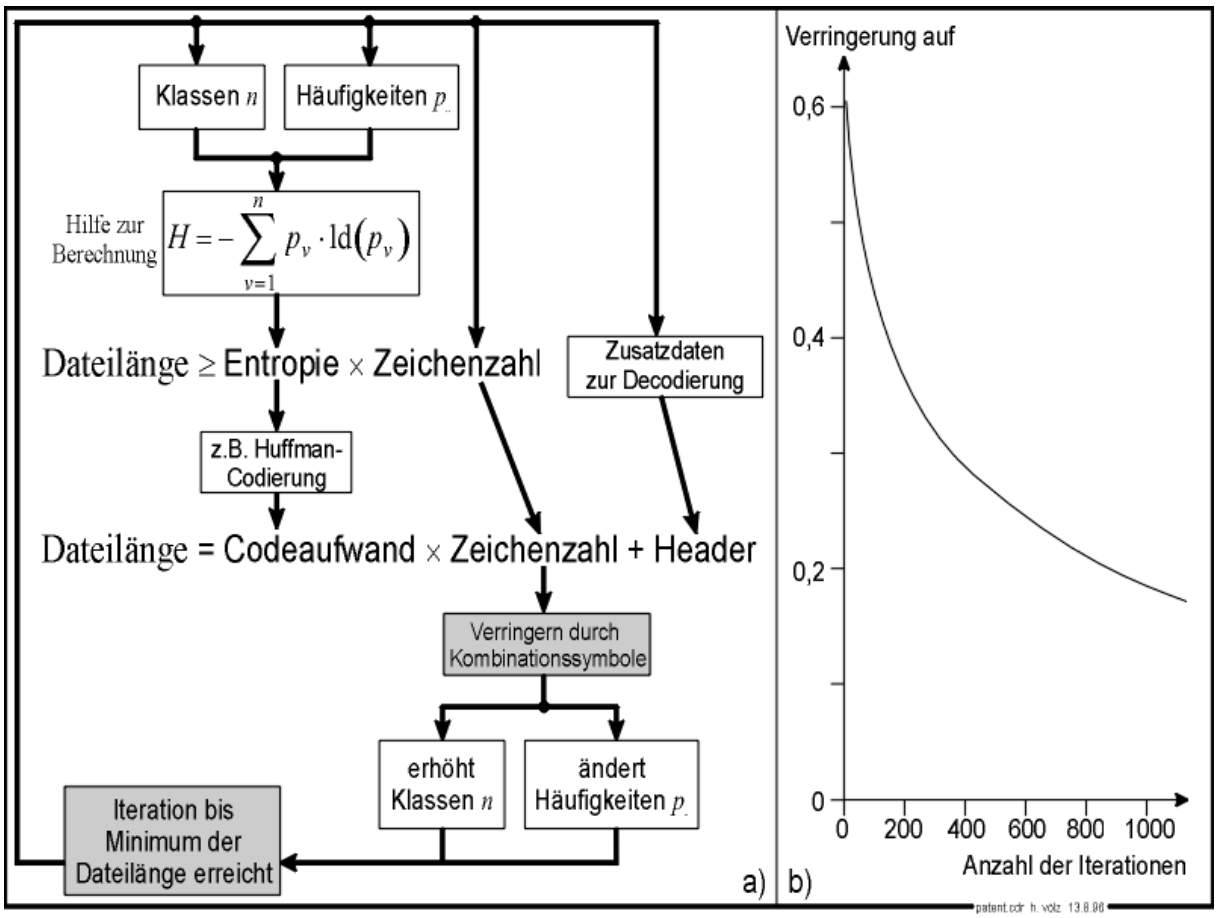
Berechnung des Codierungsgewinns beim Auflösen

H	alte Entropie
m	alte Zeichenzahl
s	alte Dateigröße H·m
ab	Anzahl des Symbols, hervorgegangen aus Kombination von a und b
a	Anzahl des Symbols a
b	Anzahl des Symbols b
n	= m + ab Neue Zeichenzahl

Entropie

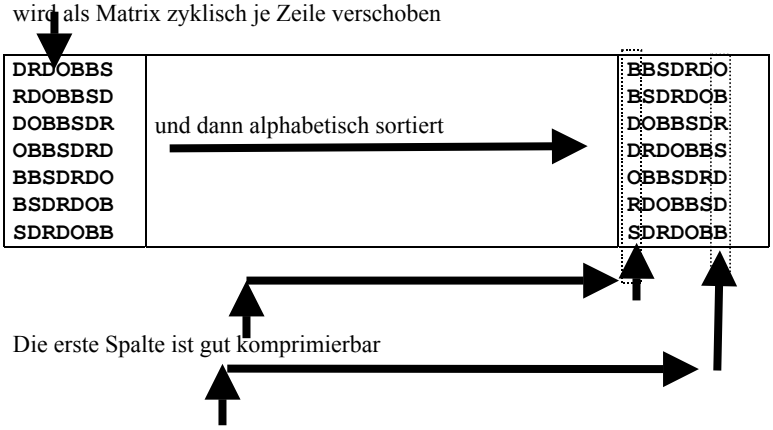
$$H'' = \frac{m}{n}\left(H - \mathrm{ld}(m) + \mathrm{ld}(n)\right) \quad + \frac{ab}{n}\cdot \mathrm{ld}\left(\frac{ab}{n}\right)$$
$$- \frac{a}{n}\cdot \mathrm{ld}\left(\frac{a+ab}{a}\right) \quad - \frac{ab}{n}\cdot \mathrm{ld}\left(\frac{a+ab}{n}\right)$$
$$- \frac{b}{n}\cdot \mathrm{ld}\left(\frac{b+ab}{b}\right) \quad - \frac{ab}{n}\cdot \mathrm{ld}\left(\frac{b+ab}{n}\right)$$

Neue Dateigröße s“ = H“·n



Kompression mit der BWT = Burrows-Wheeler-Transformation

1983 wird von Wheeler diese Transformation intern publiziert
1994 erfolgt der erste Report in Bezug auf Kompression
Ein Text z. B. **DRDOBBS**
wird als Matrix zyklisch je Zeile verschoben



Die letzte Spalte ist das Präfix zur ersten Spalte
Hierüber lässt sich mit erster Spalte (komprimiert) und Reihenfolge letzter Spalte (nur als Zahlen) die Datei wieder zurück-dekomprimieren.

Verlustbehaftete Kompression

Ziel sei der Mensch, dann zwei unterschiedliche Wege:

1. Was wir nur wollen | brauchen:

Sprache ⇒ ASCII (Schrift) 1:1000 und mehr
Musik ⇒ MIDI (Noten) 1:1000 und mehr
Sprachverstehen ⇒ Vocoder ca. 1:300
Bild | Film fehlt so etwas (Formeln, CAD, Landkarten, Fraktale?)
Qualitätstufen, Fernsprechen, Bandbreite, Stufenzahl

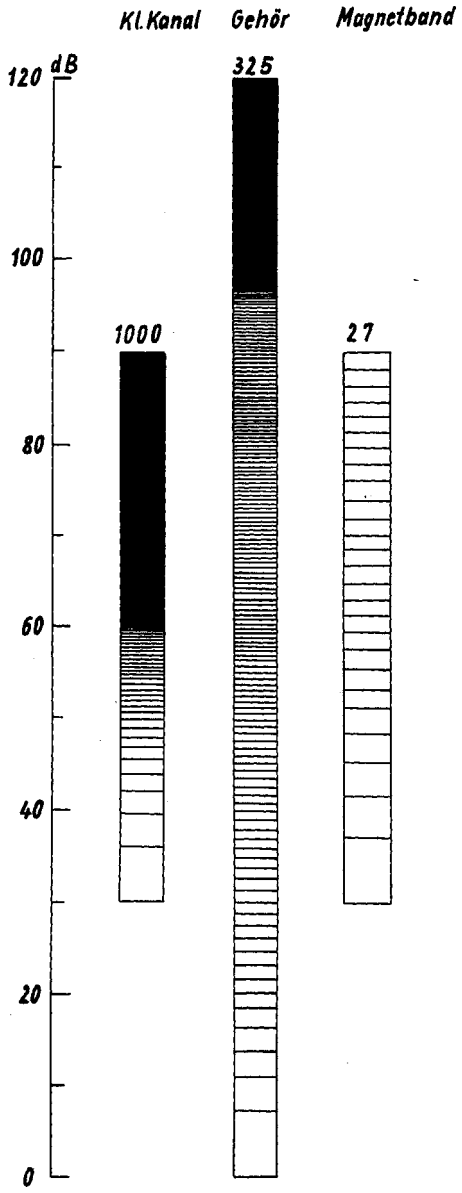
2. Das wir von der Komprimierung nichts merken:

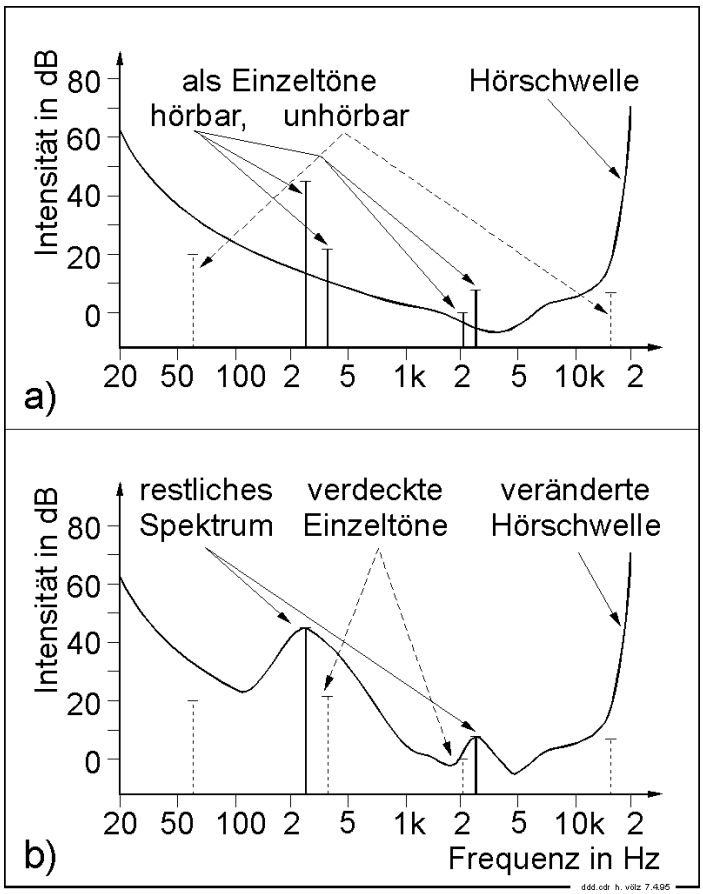
Hören:
Hörschwelle, Verdeckung, Frequenz- u. Zeitauflösung
u.a. ATTRAC, JPEG, MPEG usw.
Weber-Fechnersches Gesetz (nichtlineares Sampling)
Probleme. Hören von Richtung und Raum
Sehen:
chromatischer Fehler der Linse: YUV; HSI, JPEG usw.
MacAdams-Ellipsen: PAL ↔ NTSC
Zeitliche Auflösung: 25 bis 100 Hz; MPEG
Winkel- (+ Farb-) Auflösung des Auges (??)
Weber-Fechnersches Gesetz (??)
Probleme: Stereobild = 3D,
Holographie um das virtuelle Objekt herumgehen

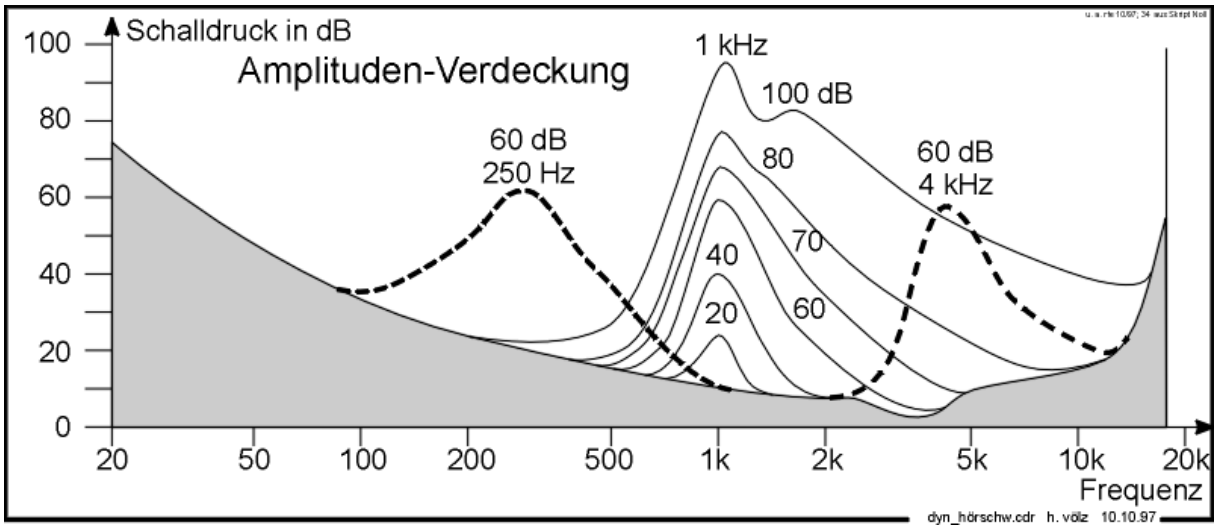
Was Lösungen erfordern

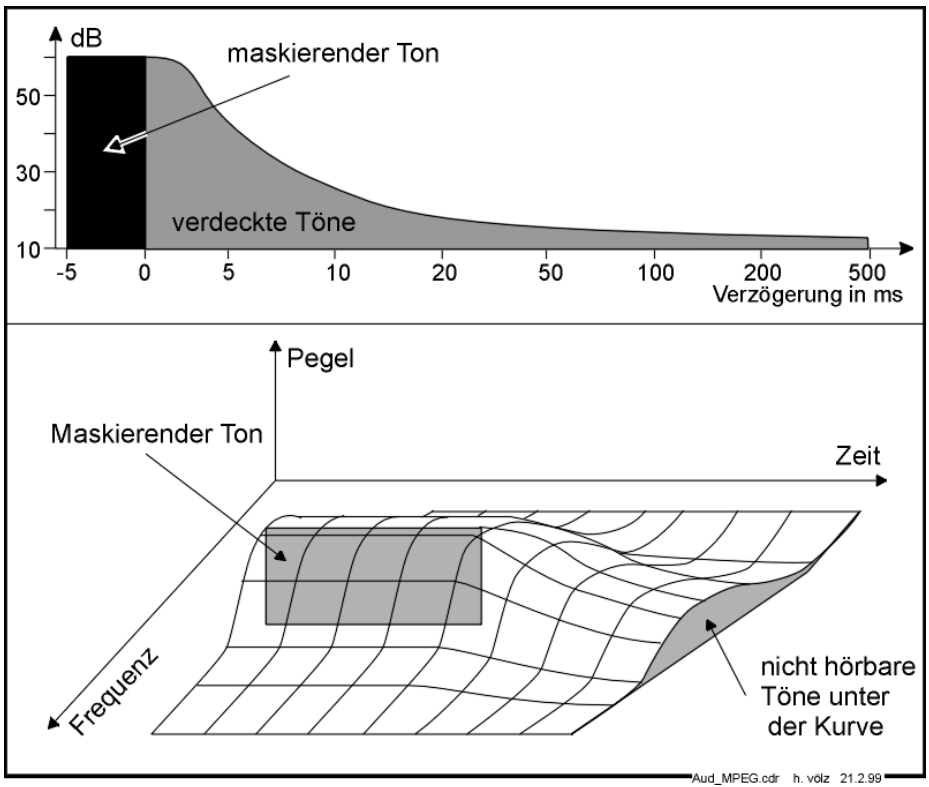
Die entsprechenden Eigenschaften sind oft nicht direkt nutzbar
Transformationen in angepaßte „Räume“ sind erforderlich
Komplexe Algorithmen heute nicht mehr das Problem
Daher ist in Zukunft viel zu erwarten

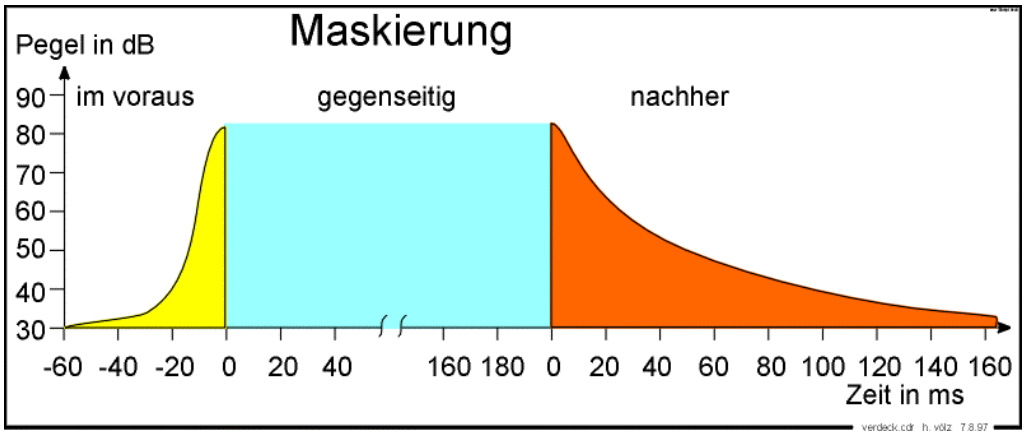
	Ton	Bild	Film	virtuelle Realität
Wahrnehmen	hören	sehen	sehen, hören	erleben, fühlen
aktive Tätigkeit	sprechen, singen, musizieren	malen, zeichnen	aufführen	bewegen, ertasten
direkte analoge Signale	Mikrophon	Scanner Photoapparat	Videokamera	Bewegungs-sensoren, Datenhandschuh (Maus, Tablett, Tastatur)
gequantelte, diskretisierte, z.T .digitalisierte Signale	CD, *.WAV; Sprach-erkennung	Pixel-Dateien, *.PCX, *.GIF; OCR; Tracing, Bilderkennung	digitales Fernsehen	
verlustbehaftete Kompression, beseitigte Irrelevanz	MUSICAM, MD, (MPEG)	JPEG, Wavelets fraktale Kompr.	MPEG, Video-CD DVD	?
Codierung zu komplexen, inhaltlichen Objekten	ASCII, MIDI	nur z.T. Vektor-Dateien, *.DXF, *.CDR	?	?


Kl.Kanal
Gehör
Magnetband
dB
120
100
80
60
40
20
0
1000
325
27


Intensität in dB
80
60
40
20
0
als Einzeltöne
hörbar,
unhörbar
Hörschwelle
20 50 100 2 5 1k 2 5 10k 2
a)
restliches
Spektrum
verdeckte
Einzeltöne
veränderte
Hörschwelle
b)
Frequenz in Hz


Schalldruck in dB
Amplituden-Verdeckung
100
80
60
40
20
0
1 kHz
100 dB
80
60 dB
250 Hz
60 dB
4 kHz
70
40
20
60
20 50 100 200 500 1k 2k 5k 10k 20k
Frequenz
dyn_hörschw.cdr h. völz 10.10.97





Fourieranalyse

Jede in T= 2·π/ω periodische Zeitfunktion = Summe harmonischer Schwingungen

Es gibt drei Schreibweisen

$$f(t)=\frac{a_0}{2}+\sum_{n=1}^{\infty}(a_n\cdot\cos(n\cdot\omega\cdot t)+b_n\cdot\sin(n\cdot\omega\cdot t))$$

$$f(t)=A_0+\sum_{n=1}^{\infty}A_n\cdot\cos(n\cdot\omega\cdot t+\varphi_n)$$

$$f(t)=\sum_{n=-\infty}^{\infty}c_n(j\cdot n\cdot\omega)\cdot e^{j\cdot n\cdot\omega\cdot t}$$

zu beachten ist:

$$e^{j\cdot\varphi}=\cos(\varphi)+j\cdot\sin(\varphi)$$

Für die Umrechnung der Koeffizienten außer für n=0 gilt

$$A_n=\sqrt{a_n^2+b_n^2}\quad\text{und}\quad\tan(\varphi)=\frac{b_n}{a_n}\qquad\text{bzw.}\qquad c_n=\frac{1}{2}\cdot A_n\cdot e^{j\cdot\omega\cdot t}$$

Zur Bestimmung der Koeffizienten gibt es mehrere Methoden.

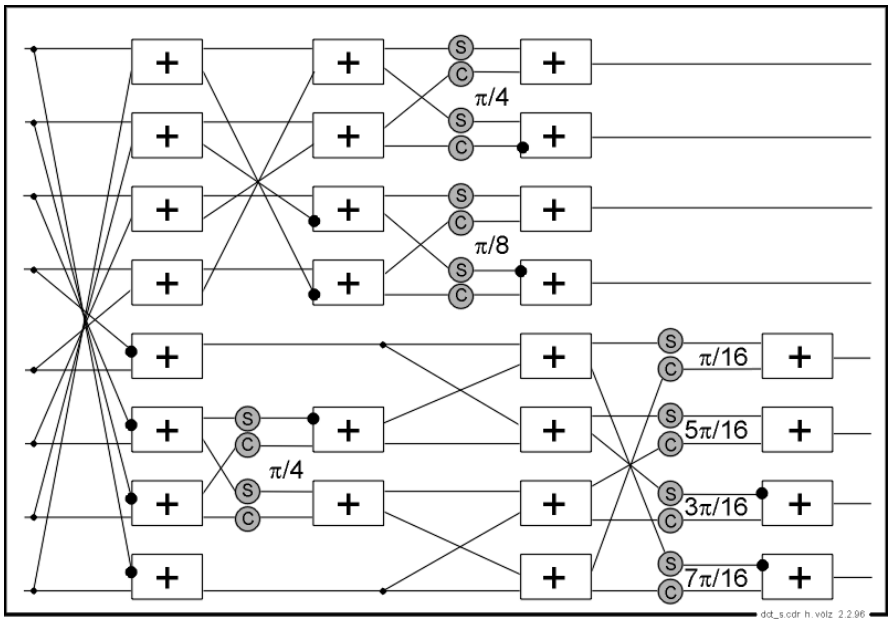
Ist $f(t)$ als mathematische Funktion, so gilt u.a.

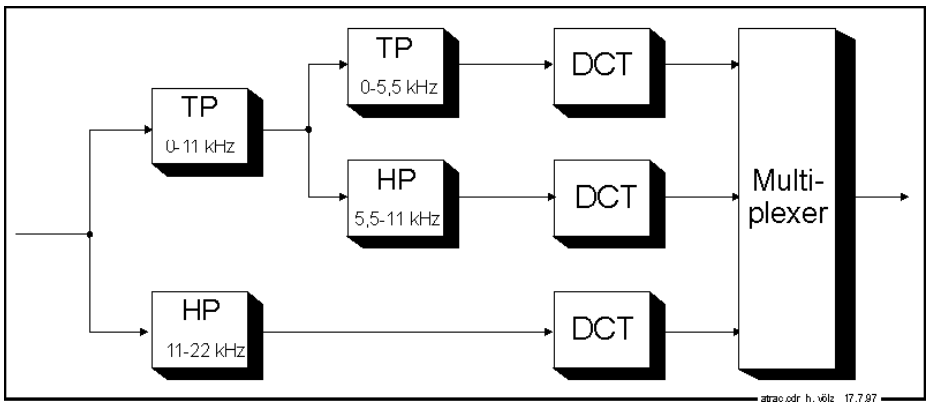
$$a_0 = \int_{t}^{t+2\cdot\pi} f(t)\cdot dt$$

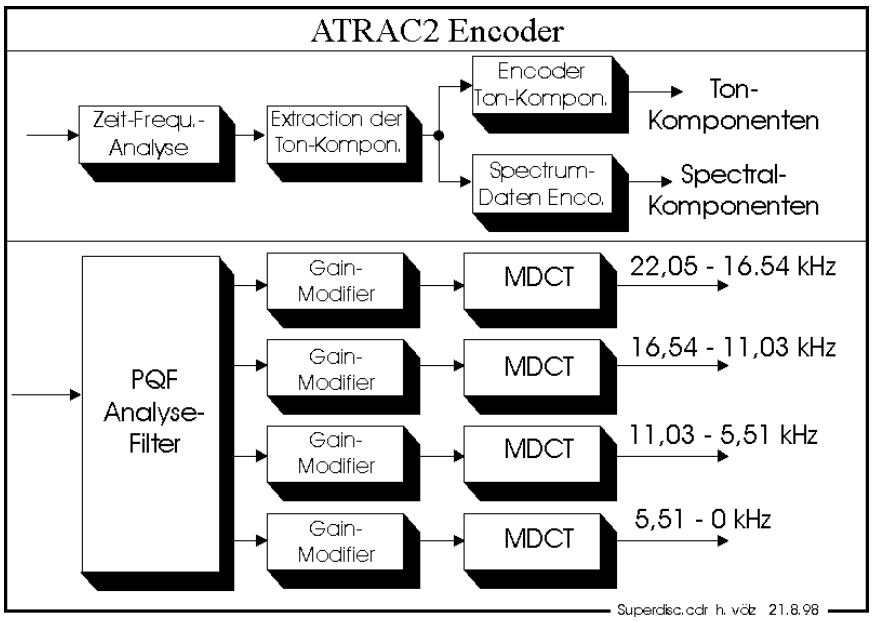
$$a_n = \int_{t}^{t+2\cdot\pi} f(t)\cdot\cos(n\cdot\omega\cdot t)\cdot dt$$

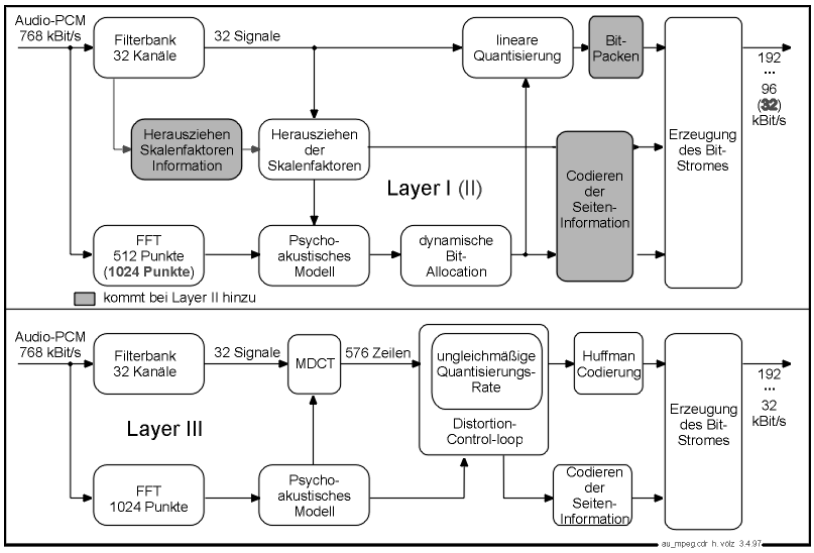
$$b_n = \int_{t}^{t+2\cdot\pi} f(t)\cdot\sin(n\cdot\omega\cdot t)\cdot dt$$

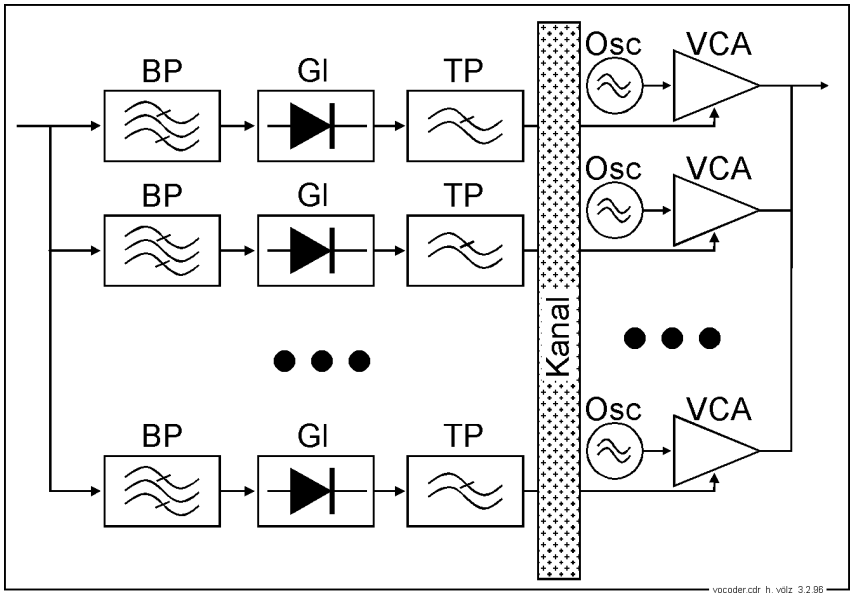
Die Koeffizienten hängen erheblich vom Startzeitpunkt *t* ab.
Die Umrechnung auf die A_n und φ_n bzw. c_n ist im Prinzip einfach
aber mit Tücken behaftet.
fourier.doc h. völz 19.3.97 / 1

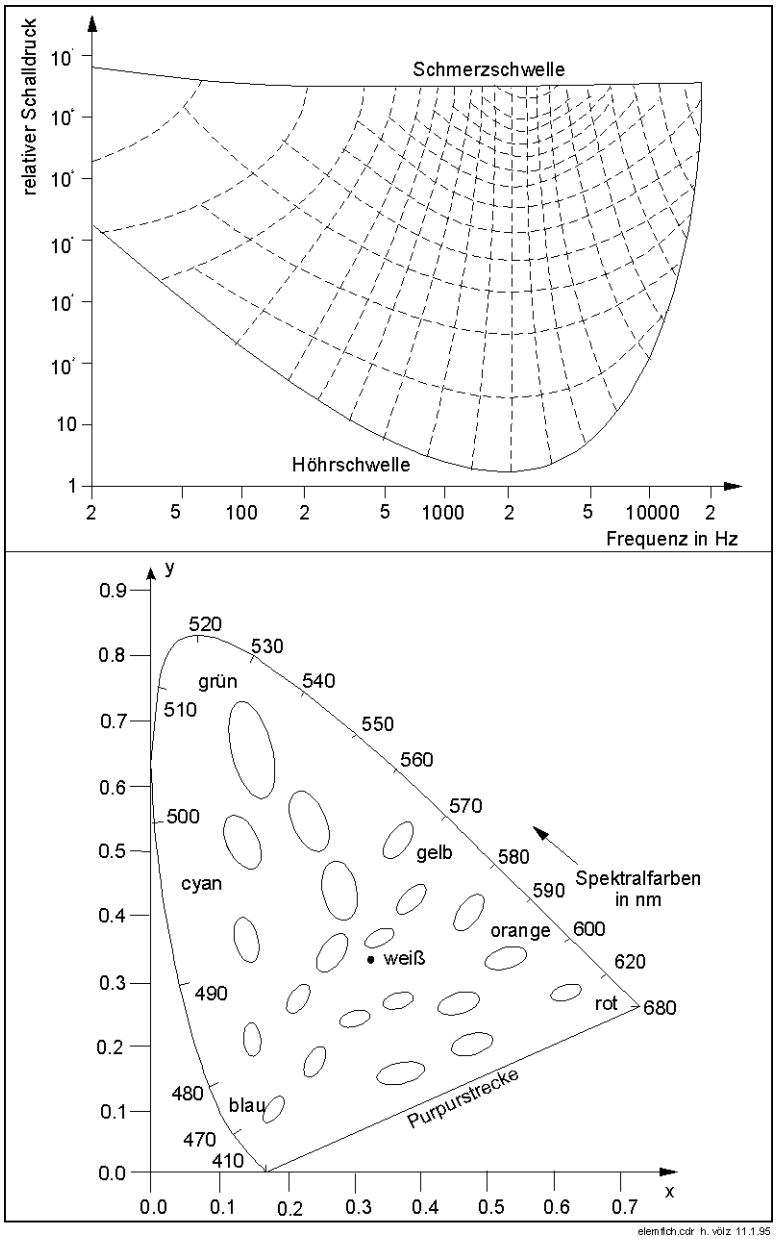


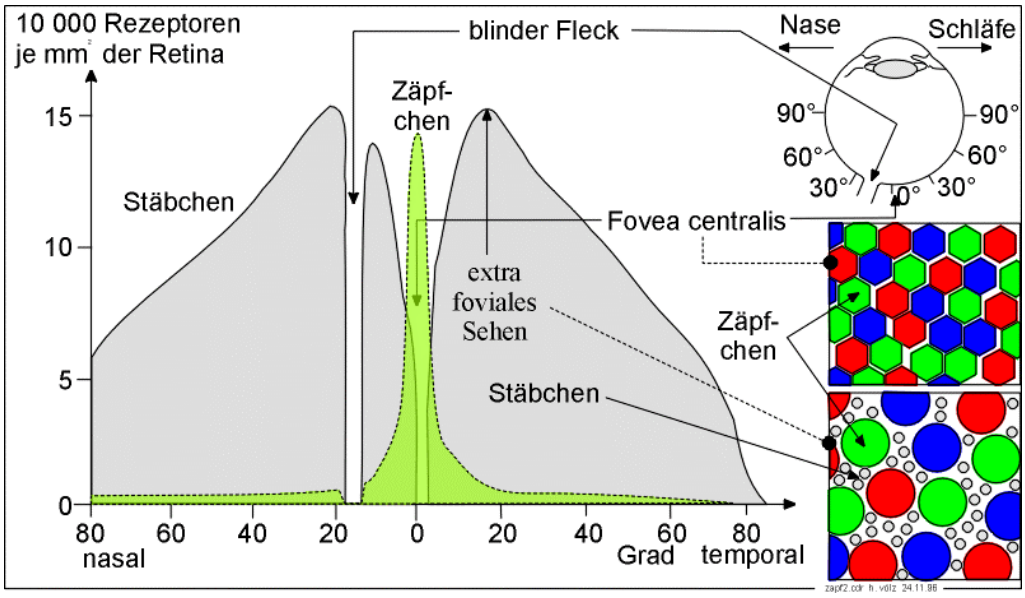


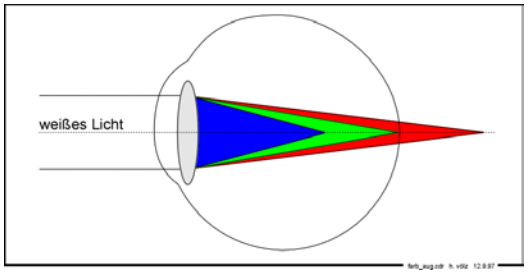

ATRAC2 Encoder
Zeit-Frequ.-Analyse
Extraction der Ton-Kompon.
Encoder Ton-Kompon.
Spectrum-Daten Enco.
Ton-Komponenten
Spectral-Komponenten
PQF Analyse-Filter
Gain-Modifier
MDCT
22,05 - 16.54 kHz
16,54 - 11,03 kHz
11,03 - 5,51 kHz
5,51 - 0 kHz
Superdisc.cdr h. völz 21.8.98

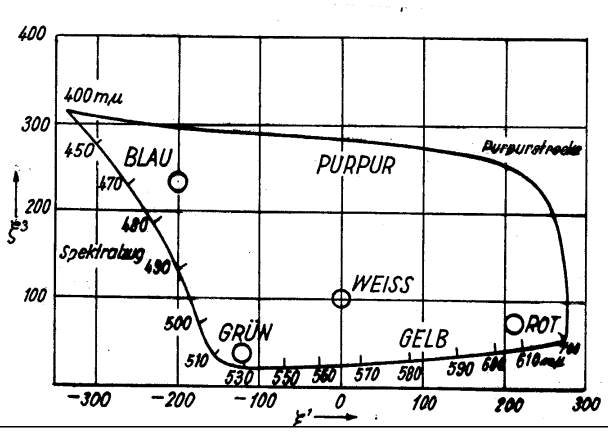

Audio-PCM 768 kBit/s
Filterbank 32 Kanäle
32 Signale
lineare Quantisierung
Bit-Packen
192 ... 96 (32) kBit/s
Herausziehen Skalenfaktoren Information
Herausziehen der Skalenfaktoren
Layer I (II)
Codieren der Seiten-Information
Erzeugung des Bit-Stromes
FFT 512 Punkte (1024 Punkte)
Psycho-akustisches Modell
dynamische Bit-Allocation
kommt bei Layer II hinzu
Audio-PCM 768 kBit/s
Filterbank 32 Kanäle
32 Signale
MDCT
576 Zeilen
ungleichmäßige Quantisierungs-Rate
Distortion-Control-loop
Huffman Codierung
Layer III
FFT 1024 Punkte
Psycho-akustisches Modell
Codieren der Seiten-Information
Erzeugung des Bit-Stromes
192 ... 32 kBit/s
au_mpeg.cdr h. völz 3.4.97

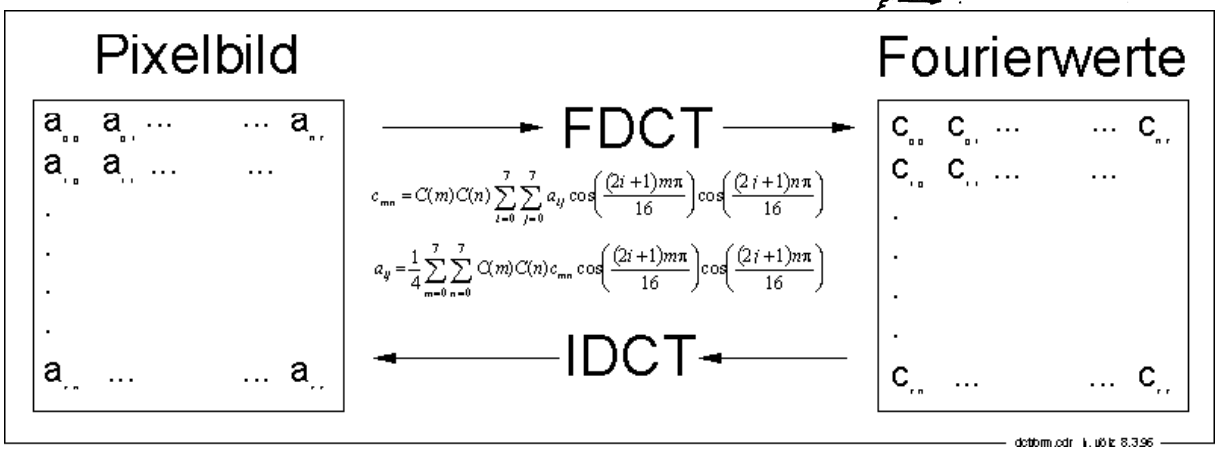

BP
GI
TP
Osc
VCA
Kanal
vocoder.cdr h. völz 3.2.96

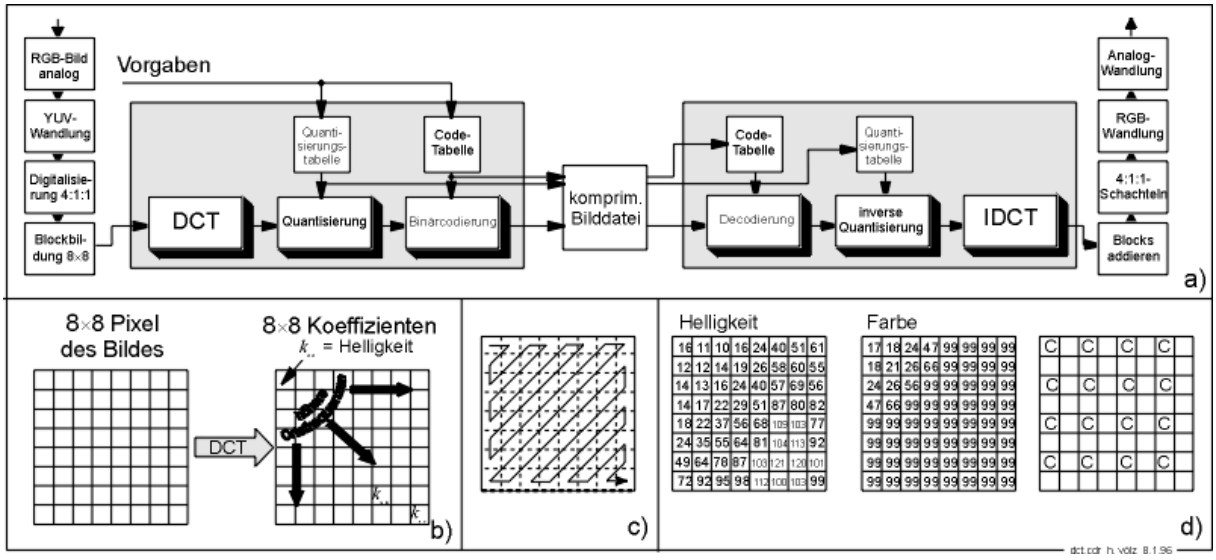

relativer Schalldruck
Schmerzschwelle
Höhrschwelle
Frequenz in Hz
y
x
520
530
540
550
560
570
580
590
600
620
680
510
500
490
480
470
410
grün
gelb
orange
rot
cyan
blau
weiß
Spektralfarben in nm
Purpurstrecke
elemfich.cdr h. völz 11.1.95

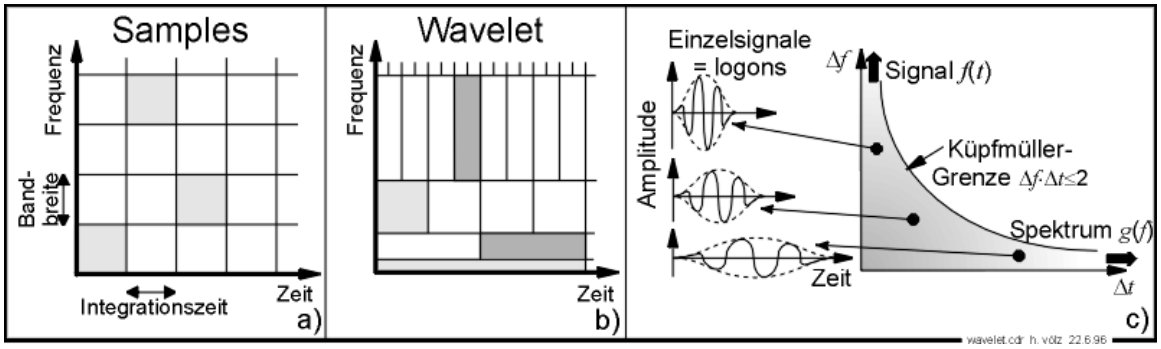

10 000 Rezeptoren je mm² der Retina
blinder Fleck
Nase
Schläfe
Zäpf-chen
Stäbchen
Fovea centralis
extra foviales Sehen
Zäpf-chen
Stäbchen
90°
60°
30°
0°
nasal
Grad
temporal
zapf2.cdr h. völz 24.11.96

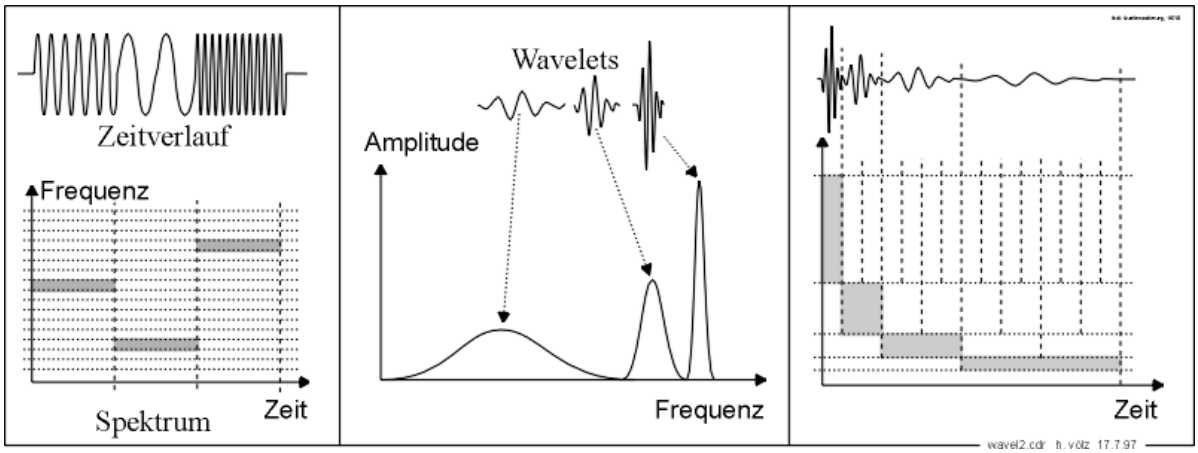

weißes Licht

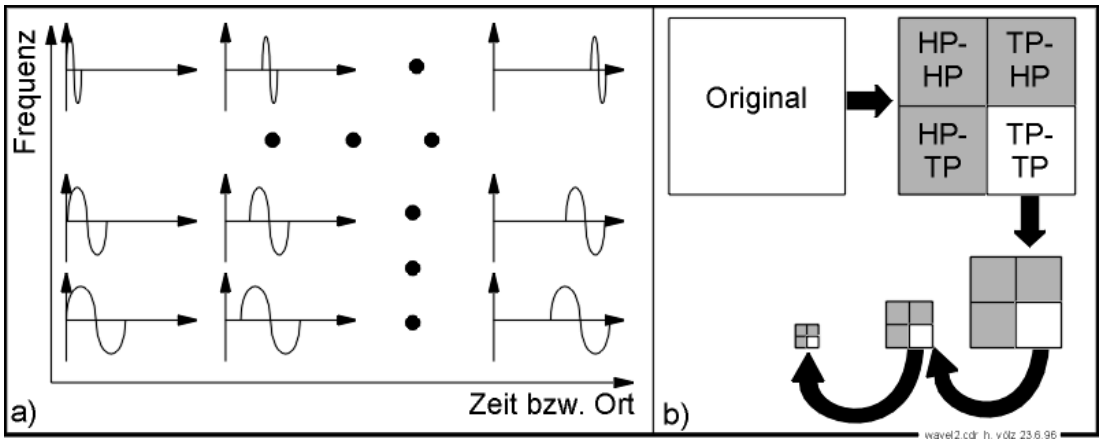

400
300
200
100
400 mμ
450
BLAU
470
480
490
500
510
530
550
560
570
580
590
610 mμ
PURPUR
Purpurgerade
Spektralzug
WEISS
GRÜN
GELB
ROT
−300
−200
−100
0
100
200
300

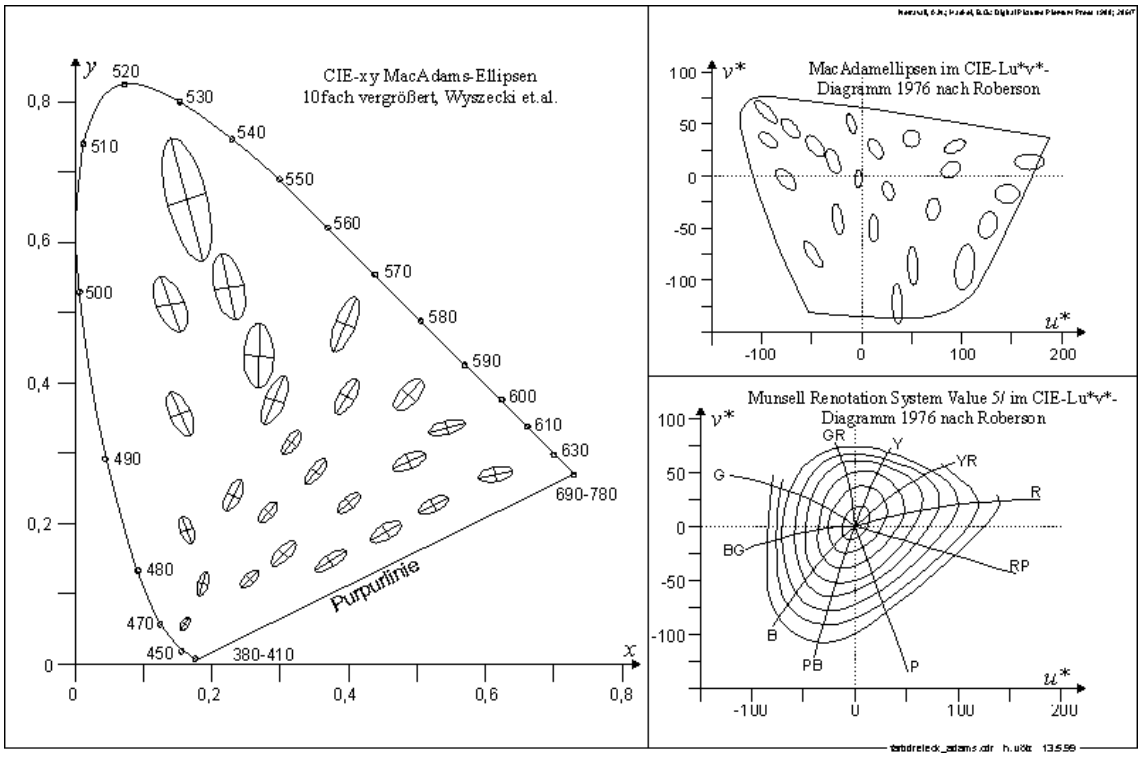

Pixelbild
Fourierwerte
FDCT
IDCT
$c_{mn} = C(m)C(n)\sum_{i=0}^{7}\sum_{j=0}^{7} a_{ij}\cos\left(\frac{(2i+1)m\pi}{16}\right)\cos\left(\frac{(2i+1)n\pi}{16}\right)$
$a_{ij} = \frac{1}{4}\sum_{m=0}^{7}\sum_{n=0}^{7} C(m)C(n)c_{mn}\cos\left(\frac{(2i+1)m\pi}{16}\right)\cos\left(\frac{(2i+1)n\pi}{16}\right)$

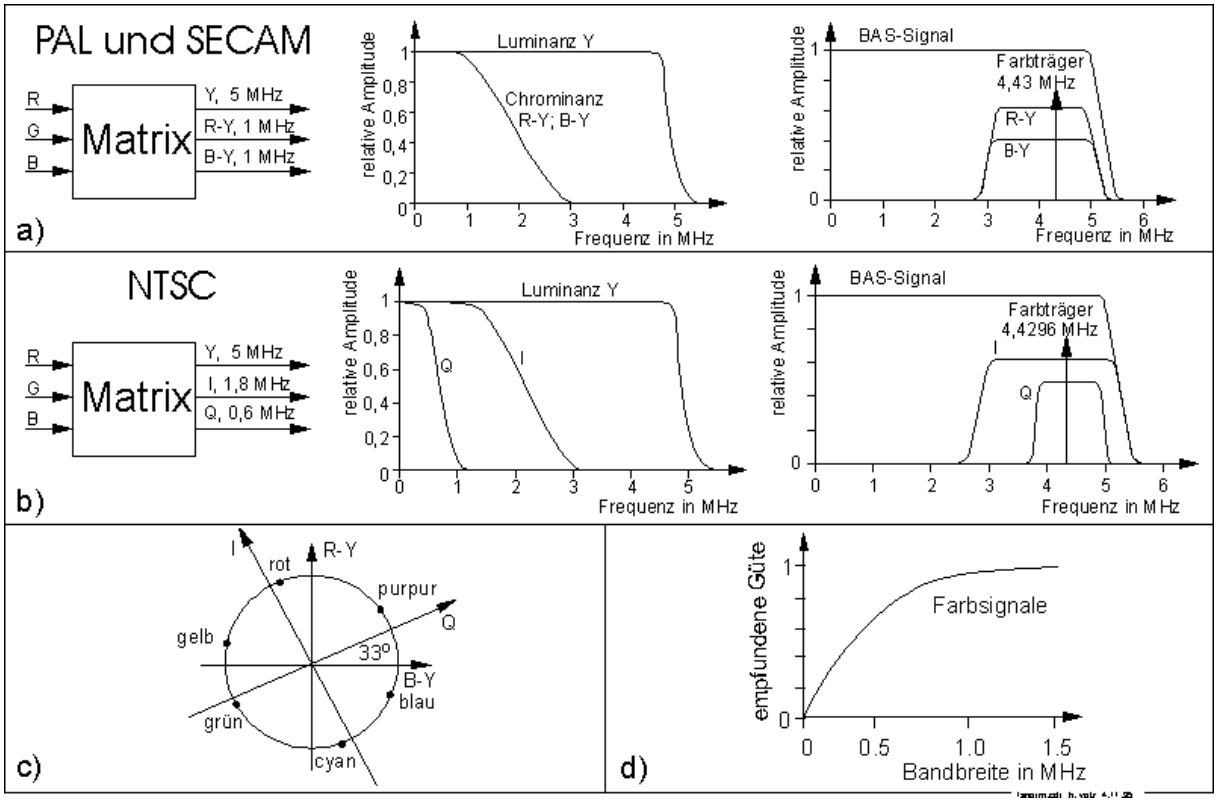

RGB-Bild analog
YUV-Wandlung
Digitalisierung 4:1:1
Blockbildung 8×8
Vorgaben
Quantisierungstabelle
Code-Tabelle
DCT
Quantisierung
Binarcodierung
komprim. Bilddatei
Decodierung
inverse Quantisierung
IDCT
Analog-Wandlung
RGB-Wandlung
4:1:1-Schachteln
Blocks addieren
a)
8×8 Pixel des Bildes
8×8 Koeffizienten
k_{00} = Helligkeit
DCT
b)
c)
Helligkeit
Farbe
d)


Samples
Frequenz
Band-breite
Integrationszeit
Zeit
a)
Wavelet
Frequenz
Zeit
b)
Einzelsignale = logons
Amplitude
Signal $f(t)$
Küpfmüller-Grenze $\Delta f \cdot \Delta t \geq 2$
Spektrum $g(f)$
Zeit
c)


Zeitverlauf
Frequenz
Spektrum
Zeit
Wavelets
Amplitude
Frequenz
Zeit
wavel2.cdr h. völz 17.7.97


Frequenz
Zeit bzw. Ort
a)
Original
HP-HP
TP-HP
HP-TP
TP-TP
b)
wavel2.cdr h. völz 23.6.96


CIE-xy MacAdams-Ellipsen
10fach vergrößert, Wyszecki et.al.
Purpurlinie
MacAdamellipsen im CIE-Lu*v*-
Diagramm 1976 nach Roberson
Munsell Renotation System Value 5/ im CIE-Lu*v*-
Diagramm 1976 nach Roberson



Kompression und orthogonale Funktionen

orthogonale Funktionsklassen

$$\int f(i,x) \cdot f(j,x) = 0 \quad \text{für } i \neq j$$

i sind ganzzahlige Laufparameter
x ist Zeit- oder Ortskoordinate

Beispiele sind:

Fourier-, Wavelet-,
Tschebyscheff- und
Legendre-Transformation

Durch die Transformationen werden zunächst nur neue Eigenschaften zugänglich, z.B. Spektrum statt Zeitfunktion

Mit der zu komprimierenden Funktion $g(x)$ werden dann Koeffizienten gebildet

$$k_n = \int g(x) \cdot f(n,x)$$

Sie beschreiben vollständig $g(x)$

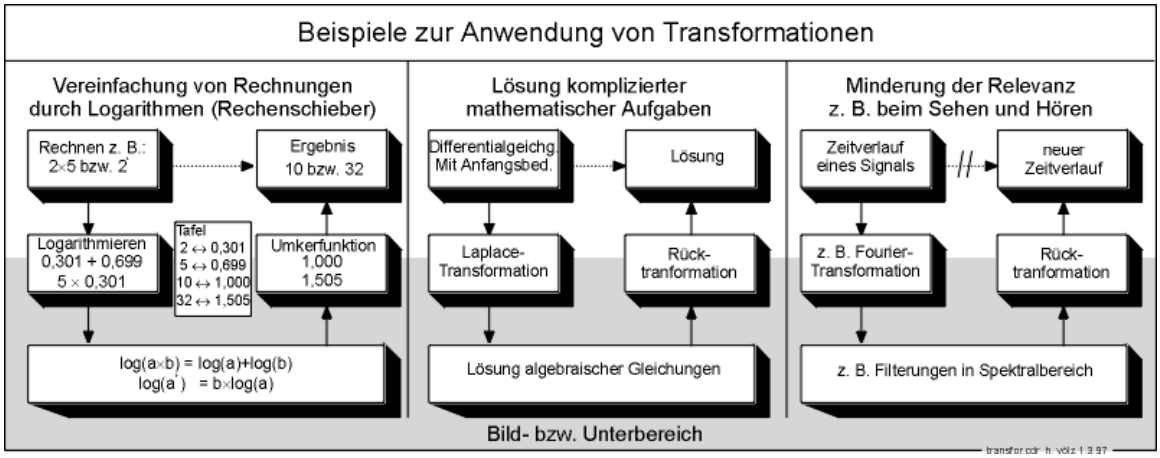
Die **Kompression** läßt jedoch im neuen Funktionsraum **kleine Koeffizienten** weg. Sie berücksichtigen oft weentlich besser die Eigenschaften des Empfangssystems.

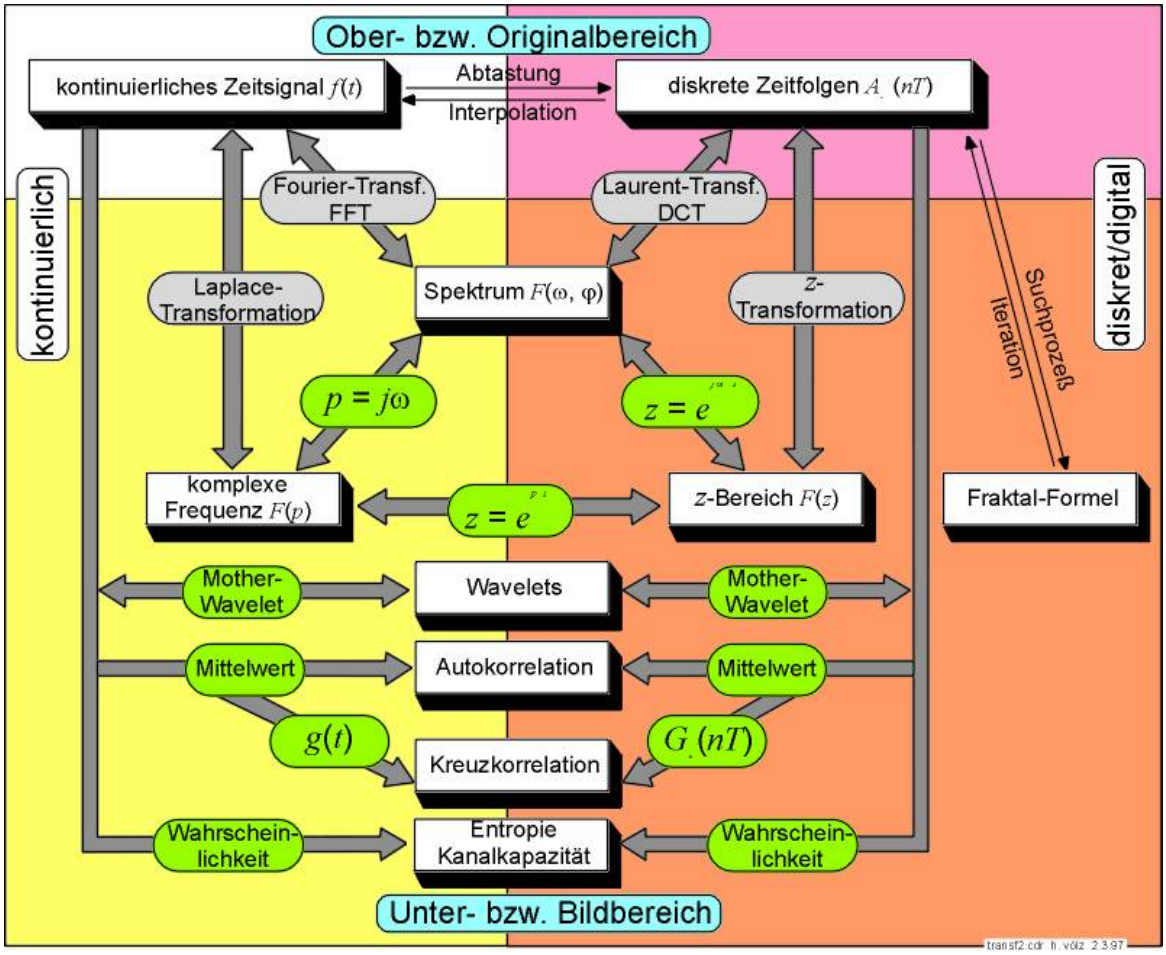
Da der **Aufwand** für die Berechnung ganzer Bilder, Bildfolgen oder Sounddateien oft unerträglich hoch wird, hat sich ein “blockweises” Bearbeiten eingeführt,
z. B. 8×8 oder 16×16 Pixel.

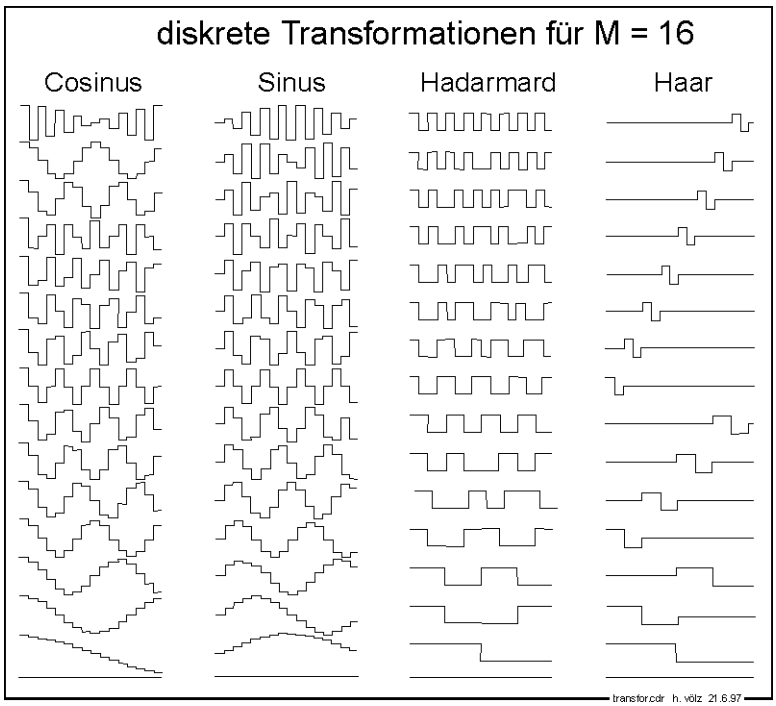
In solchen Fällen bestehen aber **weitere Möglichkeiten**

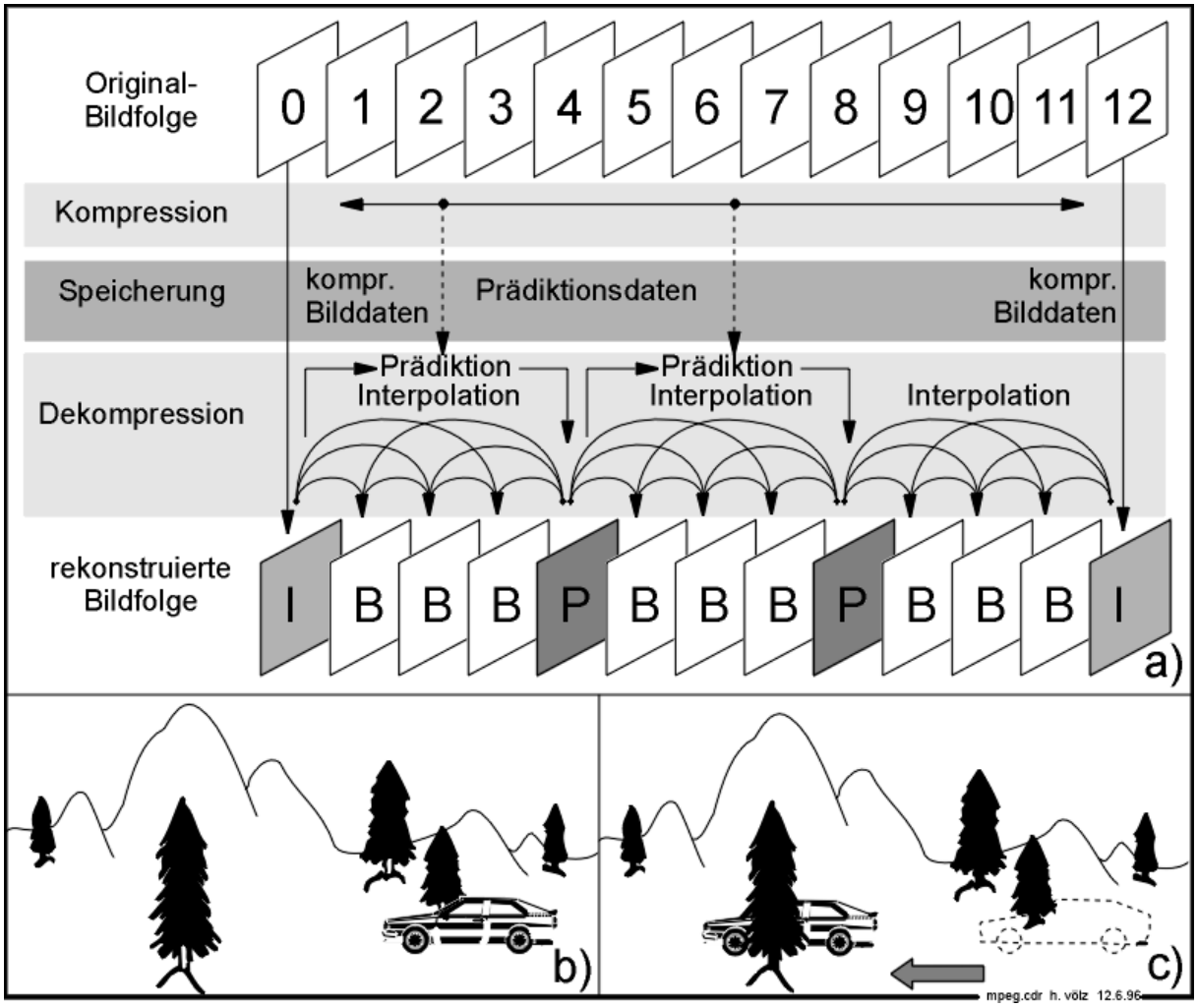
- festgelgtes Codebuch
- Faktoren zur Belegung wichtiger und unwichtiger Bereiche
- Schwellenwerte, darunter entfällt Faktor
- Approximation durch annähernde Funktion (analog Arithmetik Taylerreihe)

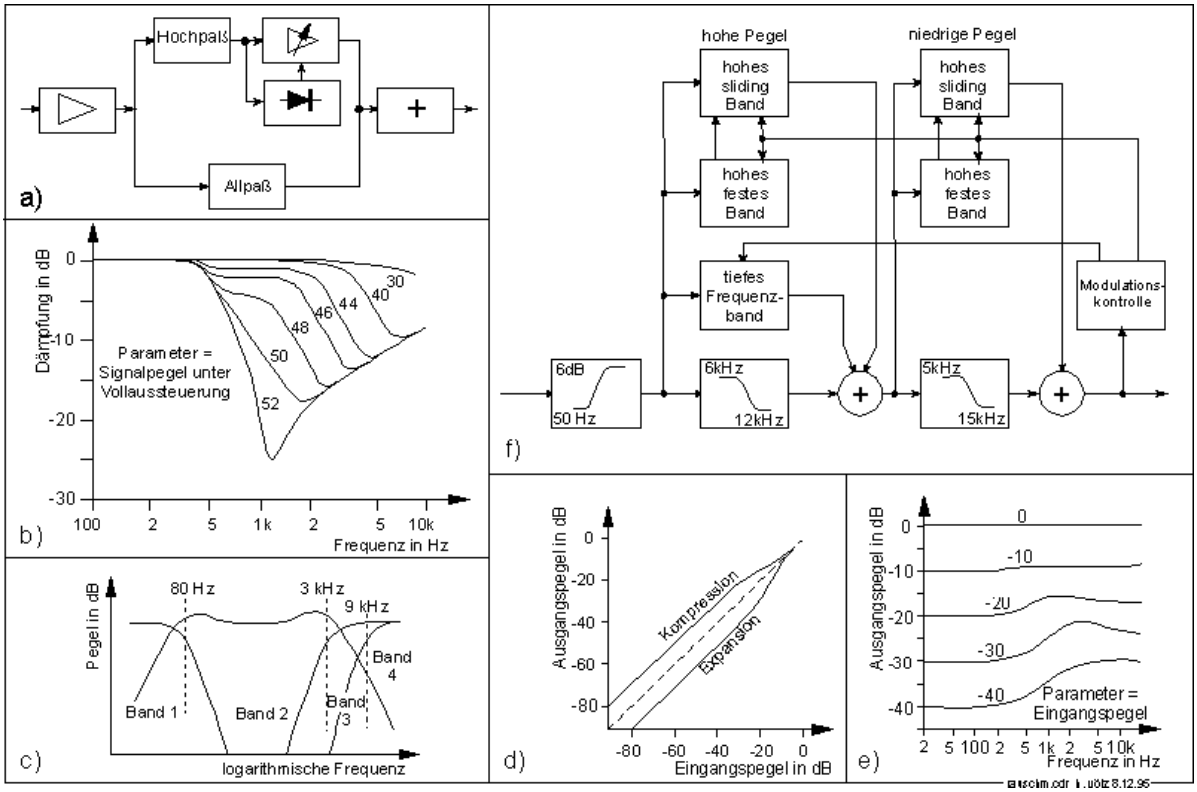
kompr_ortho.cdr h. völz 12.12.98

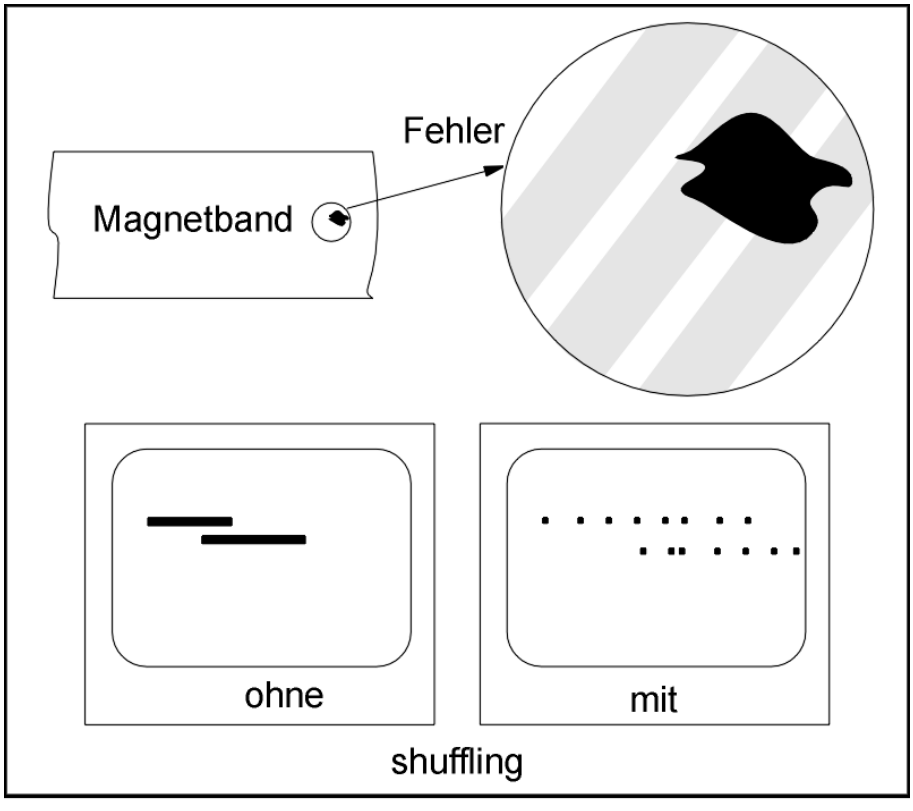

Beispiele zur Anwendung von Transformationen
Vereinfachung von Rechnungen durch Logarithmen (Rechenschieber)
Rechnen z. B.: 2×5 bzw. 2
Ergebnis 10 bzw. 32
Logarithmieren 0,301 + 0,699 5 × 0,301
Tafel 2 ↔ 0,301 5 ↔ 0,699 10 ↔ 1,000 32 ↔ 1,505
Umkerfunktion 1,000 1,505
log(a×b) = log(a)+log(b) log(a^) = b×log(a)
Lösung komplizierter mathematischer Aufgaben
Differentialgeichg. Mit Anfangsbed.
Lösung
Laplace-Transformation
Rück-tranformation
Lösung algebraischer Gleichungen
Minderung der Relevanz z. B. beim Sehen und Hören
Zeitverlauf eines Signals
neuer Zeitverlauf
z. B. Fourier-Transformation
Rück-tranformation
z. B. Filterungen in Spektralbereich
Bild- bzw. Unterbereich


Ober- bzw. Originalbereich
kontinuierliches Zeitsignal f(t)
Abtastung
Interpolation
diskrete Zeitfolgen A_ (nT)
kontinuierlich
diskret/digital
Fourier-Transf. FFT
Laurent-Transf. DCT
Laplace-Transformation
Spektrum F(ω, φ)
z-Transformation
Suchprozeß Iteration
p = jω
z = e^
komplexe Frequenz F(p)
z-Bereich F(z)
Fraktal-Formel
Mother-Wavelet
Wavelets
Mittelwert
Autokorrelation
g(t)
G_(nT)
Kreuzkorrelation
Wahrschein-lichkeit
Entropie Kanalkapazität
Unter- bzw. Bildbereich


diskrete Transformationen für M = 16
Cosinus
Sinus
Hadarmard
Haar
transfor.cdr h. völz 21.6.97


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
a)
b)
c)
mpeg.cdr h. völz 12.6.96





shuffling.cdr h. völz 10.4.95