2.2 Kompressionsverfahren für Standbilder

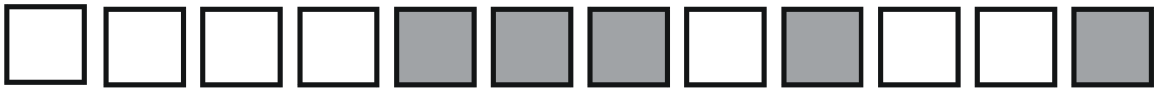
2.2.1 Fax-Kompression

Standards in der Telekommunikation werden von der International Telecommunications Union (ITU-T) entwickelt (früher: CCITT = Commitée Consultatif International de Téléphonie et Télégraphie).

Der Standard für verlustfreie Kompression von Faksimile-Anwendungen war einer der ersten Standards zur Bildkompression.

Bilder werden von Faxalgorithmen der Gruppe 3 als **bitonal** betrachtet, d.h. jedes Pixel kann entweder schwarz oder weiß sein. So kann jedes Pixel mit einem Bit repräsentiert werden. In jedem bitonalen Bild gibt es in der Regel lange Bitfolgen ("runs"), die entweder vollkommen schwarz oder vollkommen weiß sind.

Beispiel:



Lauflängen-Kodierung: 4w 3s 1w 1s 2w 1s

Fax-Standards der ITU (vormals CCITT)

Standard T.4

Erstmals verabschiedet 1980, überarbeitet 1984 und 1988 (Fax Gruppe 3) für fehleranfällige Leitungen, insbesondere Telefonleitungen.

A4-Dokumente in schwarz/weiß.

100 dots per inch (dpi) oder 3,85 Zeilen/mm vertikal

1728 Pixel pro Zeile

Ziel:

Übertragung einer Seite in einer Minute bei 4800 bit/s

Standard T.6

Erstmals verabschiedet 1984 (Fax Gruppe 4) für fehlerfreie Leitungen oder digitale Speicherung.

Standards für die Fax-Kompression

Fax Gruppe 3, ITU-T Recommendation T.4:

Codierungsansatz 1: Modified Huffman Code (MH)

- Bild wird als Folge von Pixel-Zeilen betrachtet.
- Für jede Zeile wird eine Lauflängen-Codierung ermittelt.
- Die (Lauflänge,Wert)-Beschreibung wird dann Huffman-codiert, mit standardisierter, fest-geschriebener Tabelle
- Dabei werden schwarze und weiße Runs mit separaten Huffman-Codes behandelt, da die Auftrittswahrscheinlichkeiten der Lauflängen sehr unterschiedlich sind.
- Zur Fehlererkennung wird nach jeder Zeile ein EOL (end-of-line=)-Codewort eingefügt. Dies ermöglicht das Wiederaufsetzen nach Bitübertragungsfehlern.

Fax-Kompression

Codierungsansatz 2: Modified Read (MR) Code

- Pixelwerte der vorangehenden Zeile werden für die Vorhersage der aktuellen Zeile verwendet.
- Anschließend wird eine Lauflängen-Kodierung und ein statischer Huffman-Code wie bei MH eingesetzt.
- Auch hier wird ein EOL-Codewort verwendet.

Die Kodierungsansätze MH und MR werden regelmäßig abgewechselt, um die Fortpflanzung von Fehlern zu vermeiden.

Huffman-Tabelle für Fax Gruppe 3 (Ausschnitt)

White run length	Code word	Black run length	Code word
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000

Fax Gruppe 4

Fax Gruppe 4, ITU-T Recommendation T.6

Kodierungstechnik: Modified Modified Read Code (MMR)

- Vereinfachung des MR-Codes; es werden keine Fehlererkennungsmechanismen mehr eingesetzt, um die Kompressionsrate zu erhöhen.

Kompressionsraten

	Geschäftsdokumente
Gruppe 3:	20:1
Gruppe 4:	50:1

Für Fotos u. ä. ist die Kompressionsrate schlecht, weil dort nur kurze Lauflängen vorkommen. Eine adaptive arithmetische Kodierung wäre beispielsweise besser geeignet.

2.2.2 Block Truncation Coding (BTC)

Ein einfaches Verfahren für Graustufen-Bilder. Man nimmt an, dass jedes Pixel im Original mit einem Grauwert von 0 (schwarz) bis 255 (weiß) beschrieben ist.

Algorithmus BTC

1. Zerlege das Bild in Blöcke der Größe n x m Pixel
2. Berechne Mittelwert und Standardabweichung der Pixelwerte für jeden Block

$$\mu = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} Y_{i,j}$$

$$\sigma = \sqrt{\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (Y_{i,j} - \mu)^2}$$

mit $Y_{i,j}$ = Helligkeit (Graustufe) des Pixels.

3. Weise dem Block eine Bitmatrix der Größe n x m nach folgender Regel zu:

$$B_{i,j} = \begin{cases} 1 \ldots falls \;\; Y_{i,j} \leq \mu \\ \\ 0 \ldots sonst \end{cases}$$

Algorithmus BTC (Fortsetzung)

4. Berechne zwei Graustufenwerte für den Block, *a* für den Wert der dunkleren Pixel, *b* für den Wert der helleren Pixel:

$$a = \mu - \sigma \sqrt{p/q}$$

$$b = \mu + \sigma \sqrt{q/p}$$

p ist dabei die Anzahl der Pixel, die heller als der Mittelwert μ sind, *q* die Anzahl der dunkleren Pixel.

5. Ausgabe: (Bitmatrix, *a, b*) für jeden Block

Dekompression bei BTC

Für jeden Block werden die Graustufen der einzelnen Pixel wie folgt berechnet:

$$Y'_{i,j} = \begin{cases} a \ldots falls \;\; B_{i,j} = 1 \\ b \ldots sonst \end{cases}$$

Beispiel für die Kompressionsrate

Blockgröße:	4 x 4
Original (Grauwerte)	1 Byte pro Pixel
Codierte Darstellung:	Bitmatrix mit 16 Bits + 2 x 8 Bits für *a* und *b*

=> Reduktion von 16 Bytes auf 4 Bytes, also 4:1

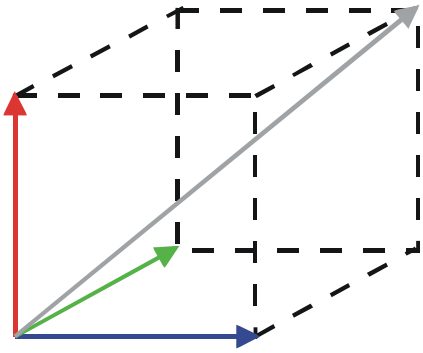
2.2.3 Color Cell Compression

Ein Verfahren zur Kompression von Farbbildern. Im Prinzip könnte man BTC für jeden der drei Farbkanäle RGB getrennt anwenden. Aber CCC liefert qualitativ bessere Ergebnisse.

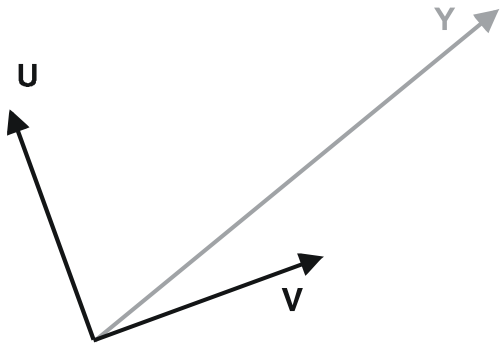
Farbmodelle

Das klassische Farbmodell ist das **RGB-Modell**. Der Farbwert eines Pixels ergibt sich aus den Intensitäten der Komponenten rot, grün und blau. Die maximale Intensität aller drei Farben ergibt weiß.

Beim **YUV-Modell** ist Y die Helligkeit (Graustufe) des Pixels, U und V sind zwei "senkrecht darauf stehende" Farbvektoren. Der Farbwert eines Pixels lässt sich leicht von dem einen in das andere Modell umrechnen.



RGB-Modell



YVU-Modell

Das YUV-Modell hat den Vorteil, dass man den Helligkeitswert eines Pixels unmittelbar kennt. So kann man leicht eine Graustufenversion des Bildes darstellen. Weiterhin kann man bei der Kompression den Helligkeitswert mit einer anderen Auflösung kodieren als die Farbwerte.

Algorithmus CCC

1. Zerlege das Bild in Blöcke der Größe m x n Pixels.
2. Berechne für jedes Farbpixel die Helligkeit gemäß:

$$Y = 0.3P_{red} + 0.59P_{green} + 0.11P_{blue}$$

Y=0 entspricht schwarz, Y=1 entspricht weiß

3. Für c = red, green, blue berechne die mittleren Farbwerte der Pixel gemäß

$$a_c = \frac{1}{q} \sum_{Y_{i,j} \leq \mu} P_{c,i,j} \,, \qquad b_c = \frac{1}{p} \sum_{Y_{i,j} \succ \mu} P_{c,i,j}$$

p ist wieder die Anzahl der Pixel, die heller als der Mittelwert sind, *q* die Anzahl der dunkleren Pixel.

Algorithmus CCC (Fortsetzung)

4. Weise dem Block eine Bitmatrix der Größe n x m nach folgender Regel zu:

$$B_{i,j} = \begin{cases} 1 \ldots falls \;\; Y_{i,\,j} \leq \mu \\ \\ 0 \ldots sonst \end{cases}$$

5. Die Werte $a = (a_{red}, a_{green}, a_{blue})$ und $b = (b_{red}, b_{green}, b_{blue})$ werden in eine Farbtabelle abgebildet, gemäß einem Nähe-Maß. Es ergeben sich die Werte *a'* und *b'* als Indizes der Farbtabelle (Color Lookup Table, CLUT).

- Ausgabe: (Bitmatrix, *a'*, *b'*) für jeden Block

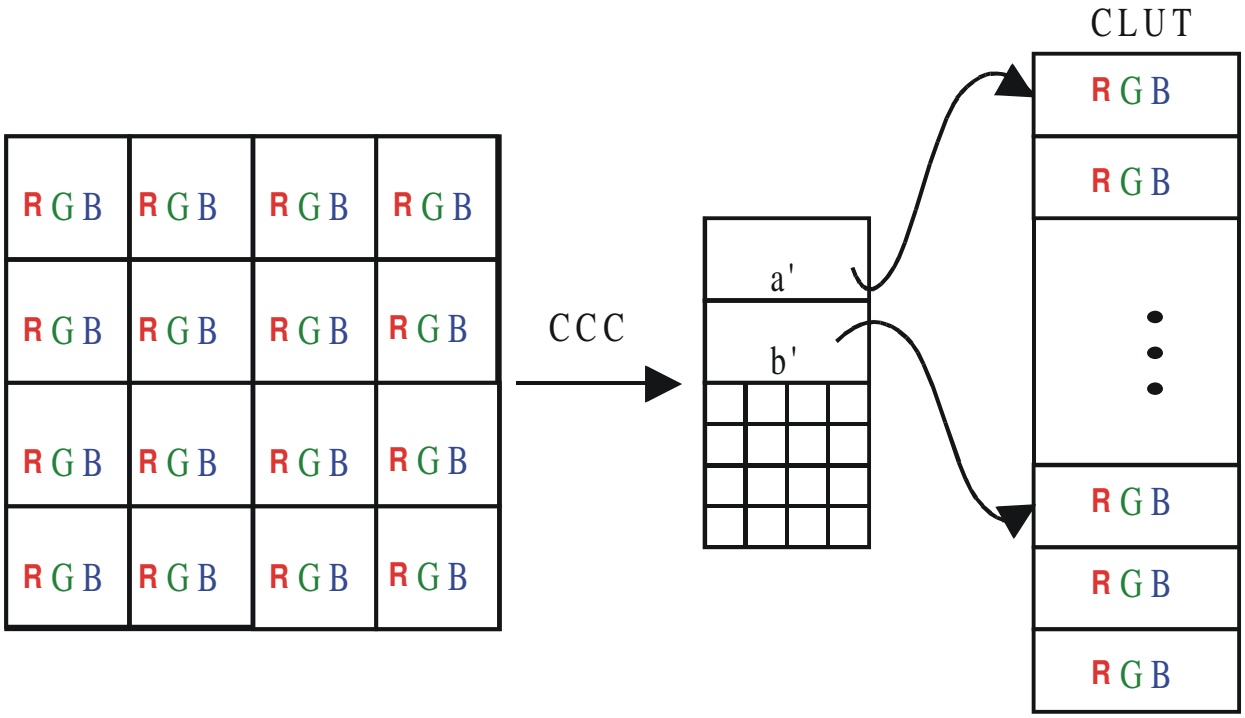
Dekompression bei CCC

Die Dekompression funktioniert für jeden Block wie folgt:

$$P'_{i,j} = \begin{cases} CLUT[a'] \ldots fallsB_{i,j} = 1 \\ CLUT[b'] \ldots sonst \end{cases}$$

Funktionsweise von CCC

Verwendung einer Farbtabelle (CLUT)

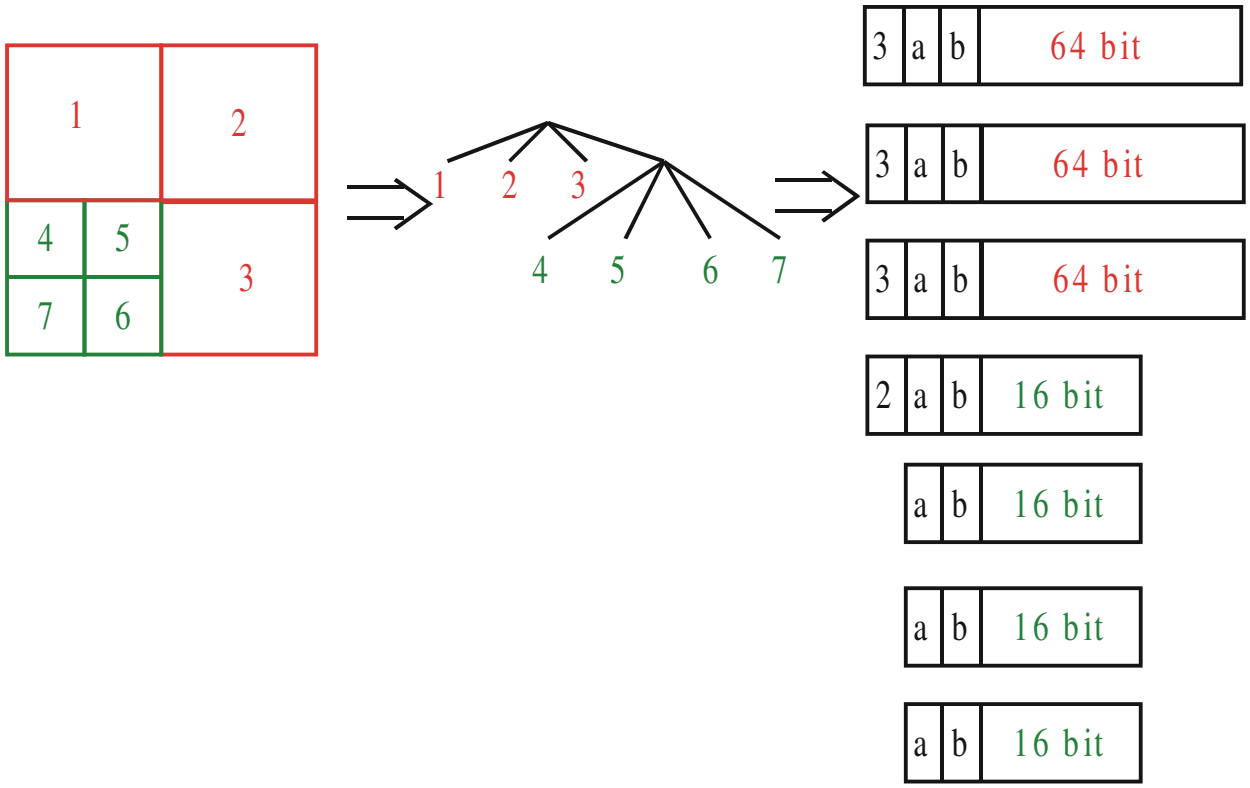


Extended Color Cell Compression (XCCC)

Eine Erweiterung von CCC zur weiteren Verbesserung der Kompressionsrate.

Idee

Hierarchie von Blockgrößen. Man versucht zunächst, einen großen Block mit CCC zu codieren. Wenn die Abweichung der tatsächlichen Farbwerte im Block von *a'* oder *b'* größer ist als ein vorgegebener Schwellenwert, wird der Block in vier Teilblöcke zerlegt. Der Algorithmus arbeitet rekursiv.



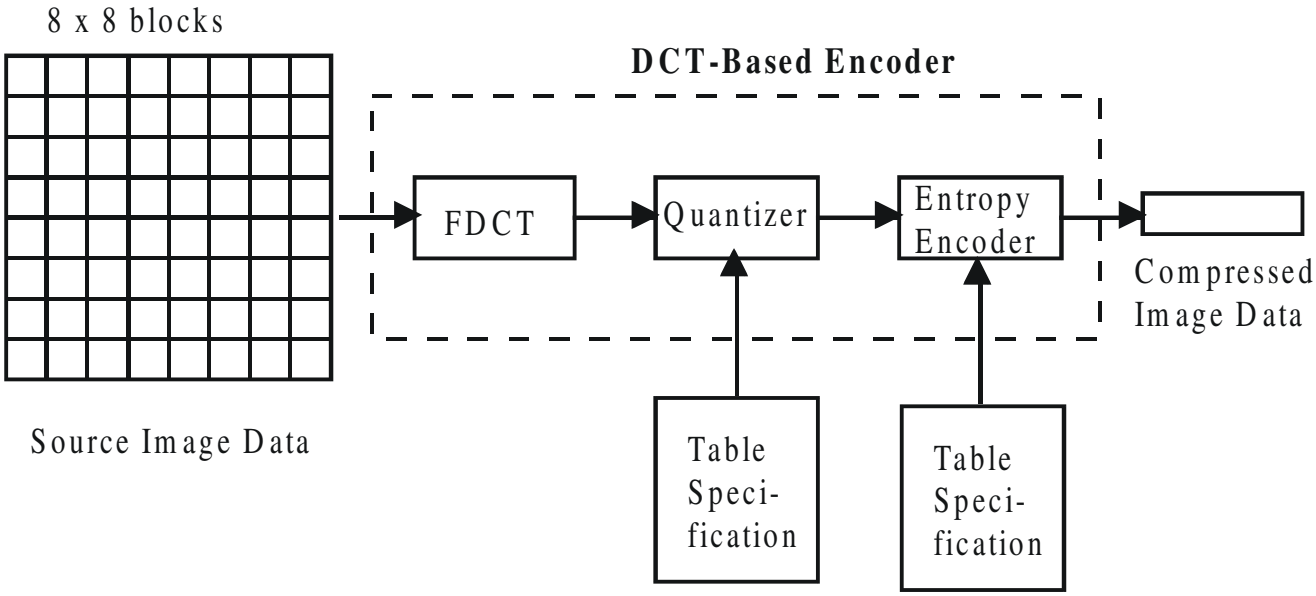
2.2.4 JPEG

Die Joint Photographic Experts Group (Komitee der ISO) hat einen sehr effizienten Kompressionsalgorithmus entwickelt, der nach ihr benannt ist. Das JPEG-Verfahren hat bei-spielsweise im WWW eine sehr weite Verbreitung gefunden.

Kompression in vier Schritten:

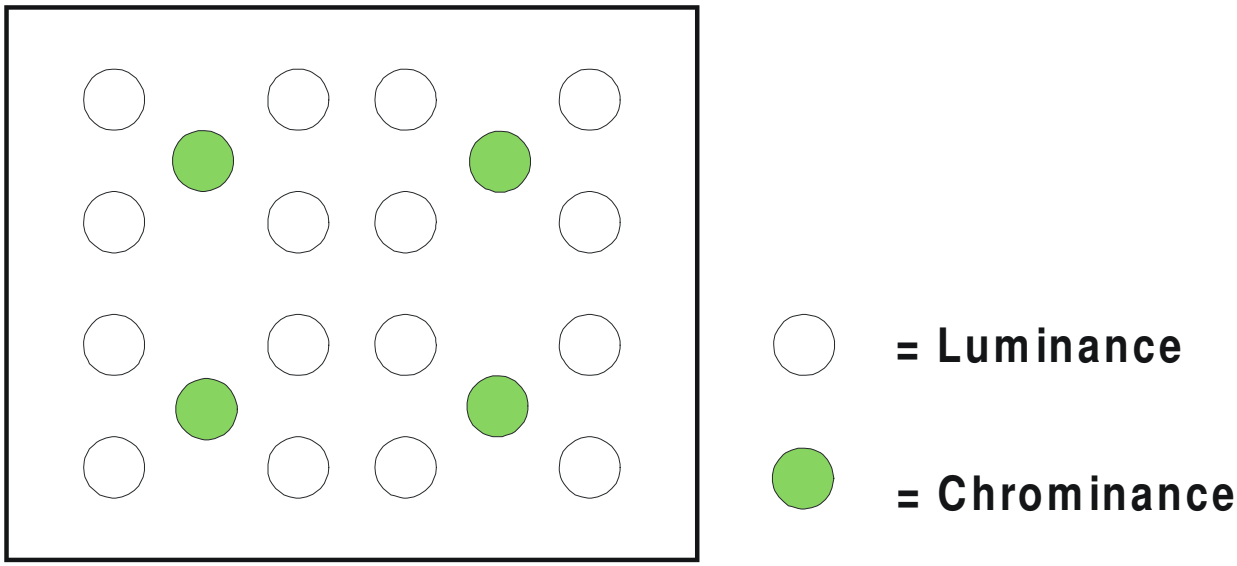
1. Bildaufbereitung
2. Diskrete Cosinus-Transformation (DCT)
3. Quantisierung
4. Entropie-Kodierung

Der DCT-basierte Codierungsprozess



Kodierung der Farbvektoren mit geringerer Auflösung ("color subsampling")

Ein Vorteil des YUV-Farbmodells ist es, dass man die Farbwerte U und V eines Pixels mit niedrigerer Auflösung kodieren kann als den Helligkeitswert Y. Das menschliche Auge reagiert nämlich auf Fehler im Hellig-keitswert empfindlicher als auf Fehler in den Farbwerten. In JPEG verwendet man dieses **color sub-sampling** in der Weise, dass für jeweils vier Helligkeitswerte Y nur je ein Chroma-Wert U und V kodiert wird.



Vier Y-Blöcke der Größe 8x8 mit einem zugehörigen U-Block der Größe 8x8 und einem V-Block der Größe 8x8 bezeichnet man als einen **Macroblock**.

JPEG "Baseline" Mode

Ein Kompressionsverfahren auf der Basis einer Transformation aus der Zeitdomäne in die Frequenzdomäne

Bildtransformation

FDCT (Forward Discrete Cosine Transformation). Sehr ähnlich der Fourier-Transformation. Wird auf jeden Block von 8x8 Pixeln einzeln angewendet.

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

mit

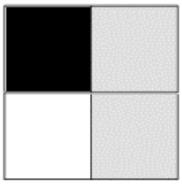
$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{2}} \text{ für } u,v=0 \\ 1 \text{ sonst} \end{cases}$$

64-mal anwenden, ergibt 64 Koeffizienten in der Frequenzdomäne

Darstellung von Bilder mit Basisbildern

Man kann ein Grauwertbild in eine Linearkombination aus Basisbildern und Koeffizienten zerlegen.

Im folgenden ist die Zerlegung eines 2x2-Bildes mit Hilfe von Basisbildern dargestellt.



=

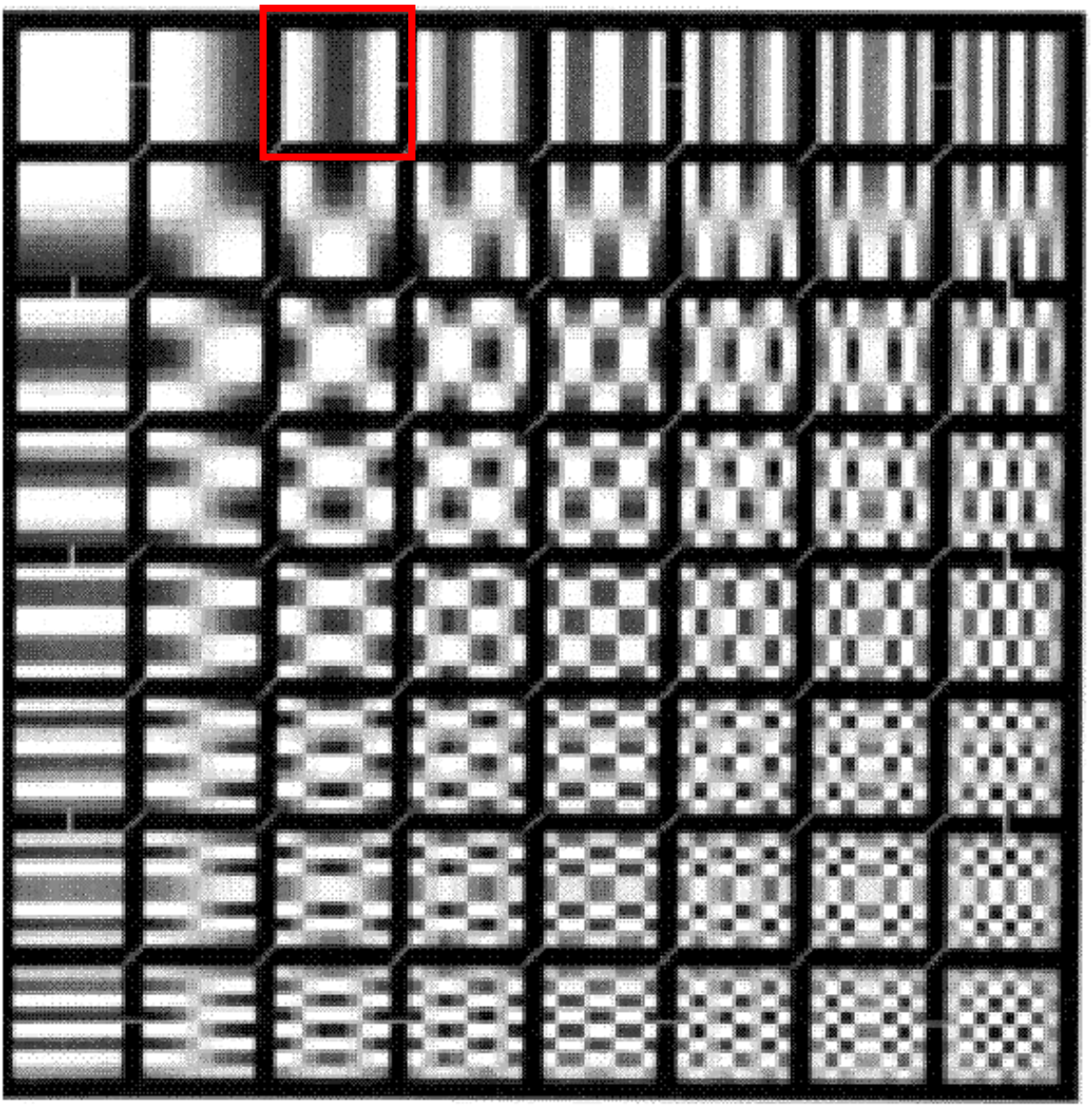
0	224
255	144

$$\begin{pmatrix} 0 & 224 \\ 255 & 144 \end{pmatrix} = 0 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + 224 \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + 255 \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + 144 \cdot \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Neben der oben dargestellten Basis aus 2x2 Basisbildern, kann man beliebige andere Basen konstruieren.

Basisbilder der 2D-DCT auf 8x8-Blöcken

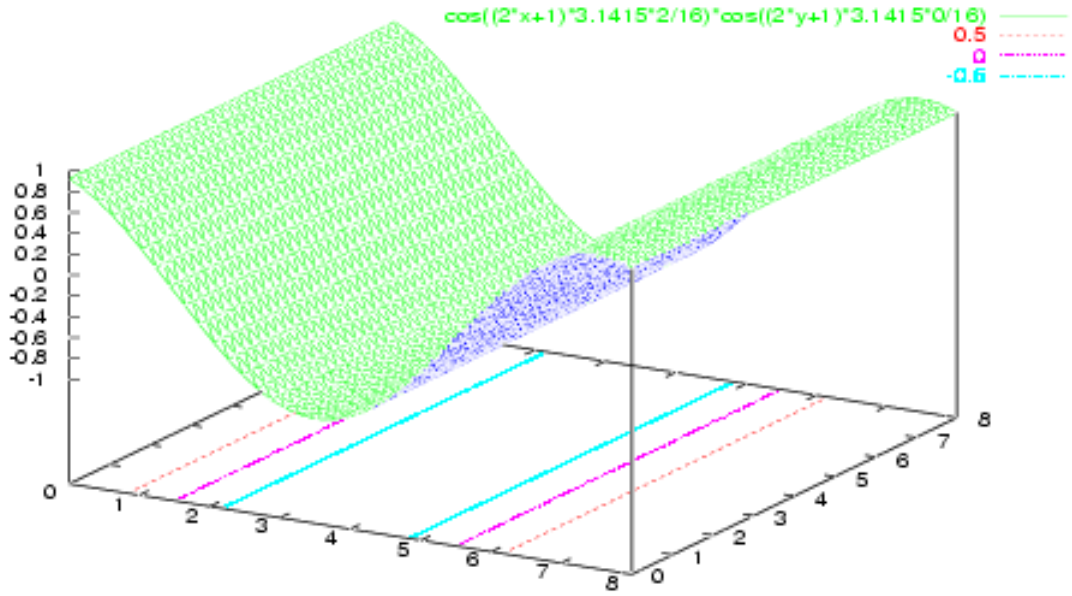
Um einen Bildblock der Größe 8x8 abzudecken, benötigt man 64 Basisbilder. Nachfolgend sind die Basisbilder dargestellt, die aus der Formel der 2D-DCT resultieren.



Kontinuierliche Darstellung eines DCT-Kernels

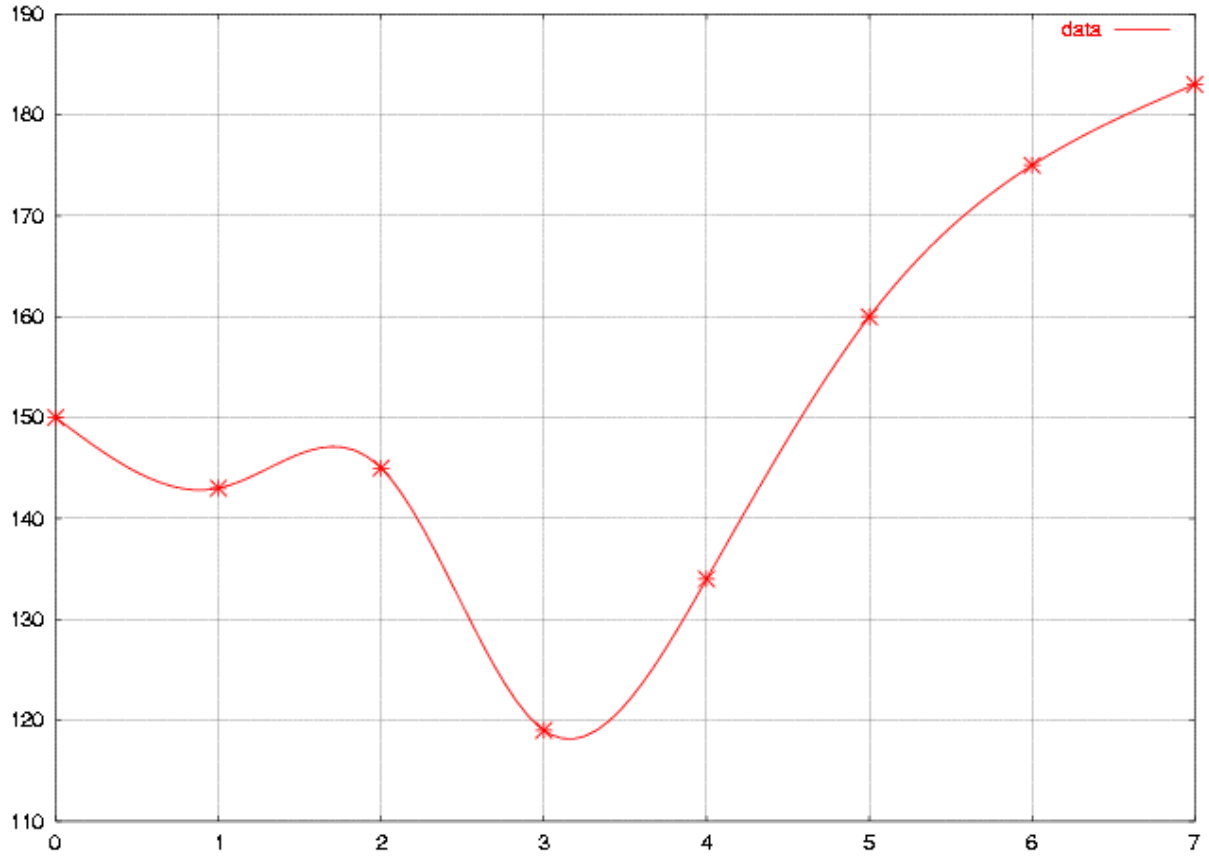
Dargestellt ist der DCT-Kernel, der das Basisbild der 1. Zeile und 3. Spalte (roter Rahmen) aus der Abbildung der vorherigen Seite ergibt.

$$\frac{\cos(2x+1)\cdot 2\pi}{16}\cdot\frac{\cos(2y+1)\cdot 0\pi}{16}$$



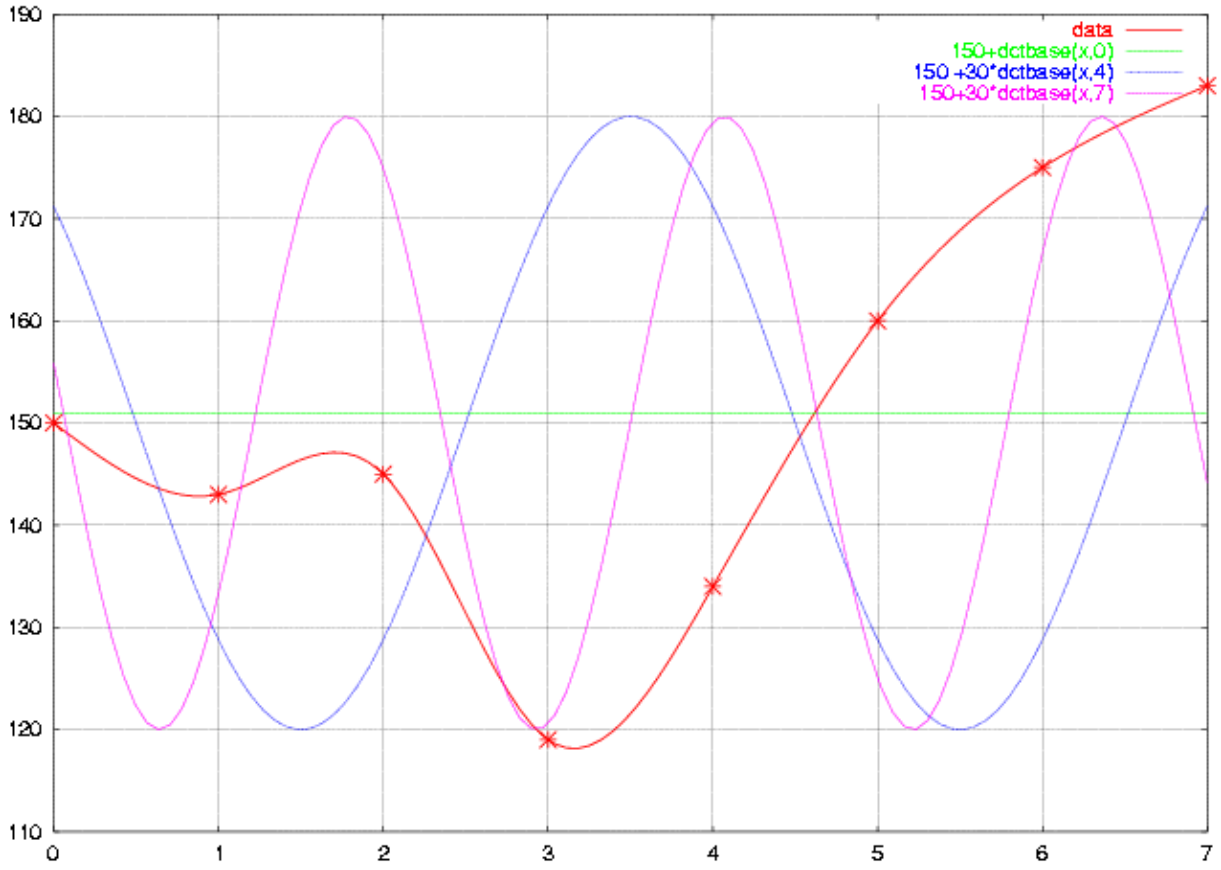
Beispiel: Kodierung eines 1D-Signals mittels 1D-DCT

Das folgende eindimensionale Signal durch Koeffizienten der 1D-DCT angenähert werden.



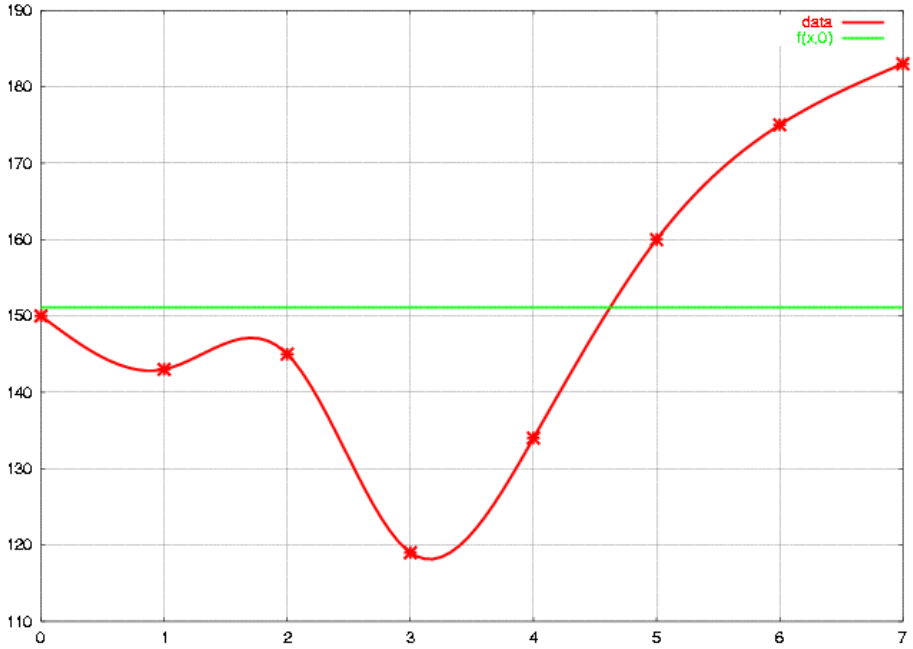
Approximation durch DCT-Koeffizienten (1)

Darstellung verschiedener eindimensionaler DCT-Kernel für die Approximation des Signals.

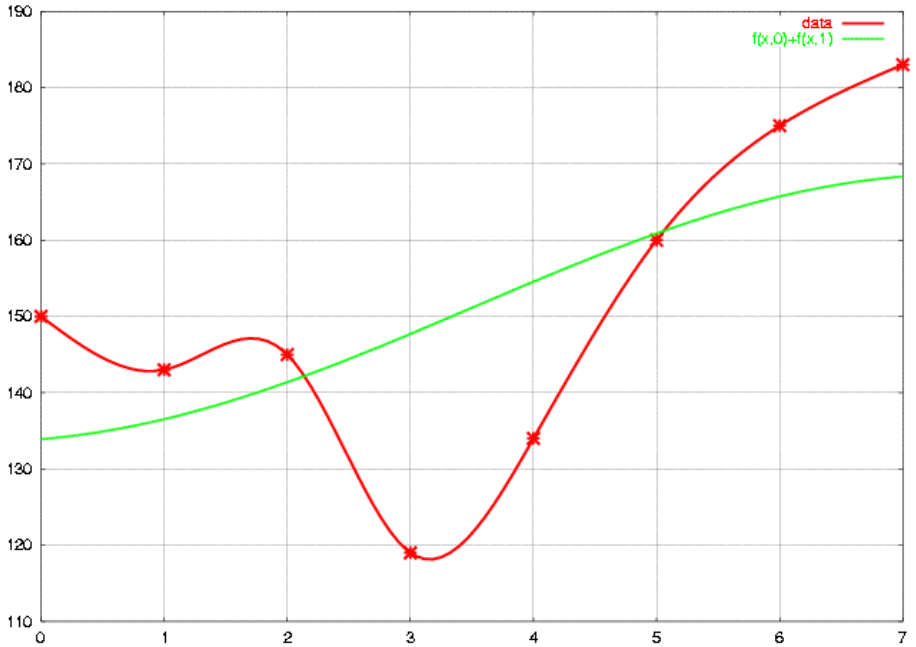


Approximation durch DCT-Koeffizienten (2)

DC-Koeffizient

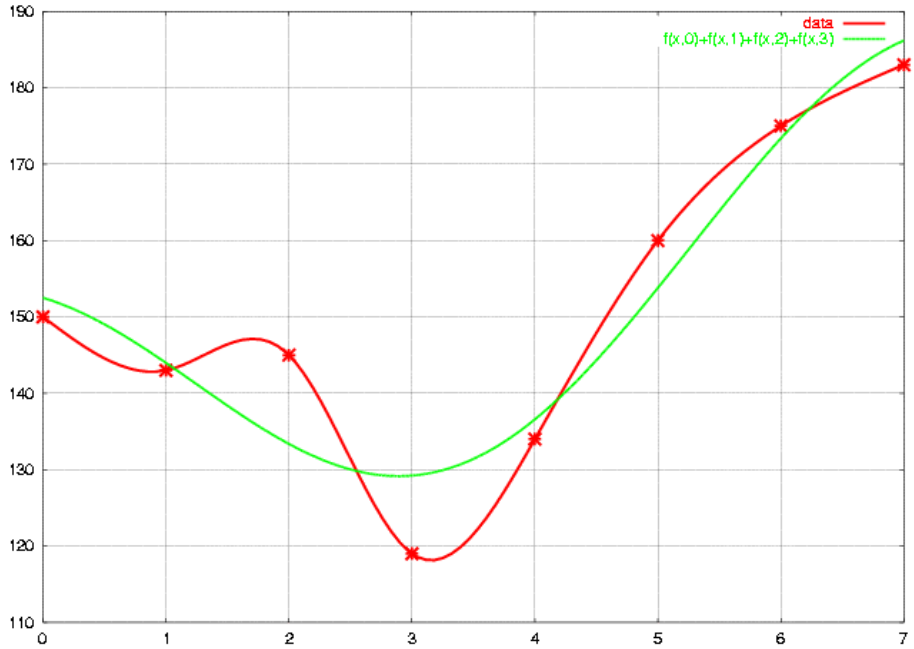


DC-Koeffizient + 1. AC-Koeffizient

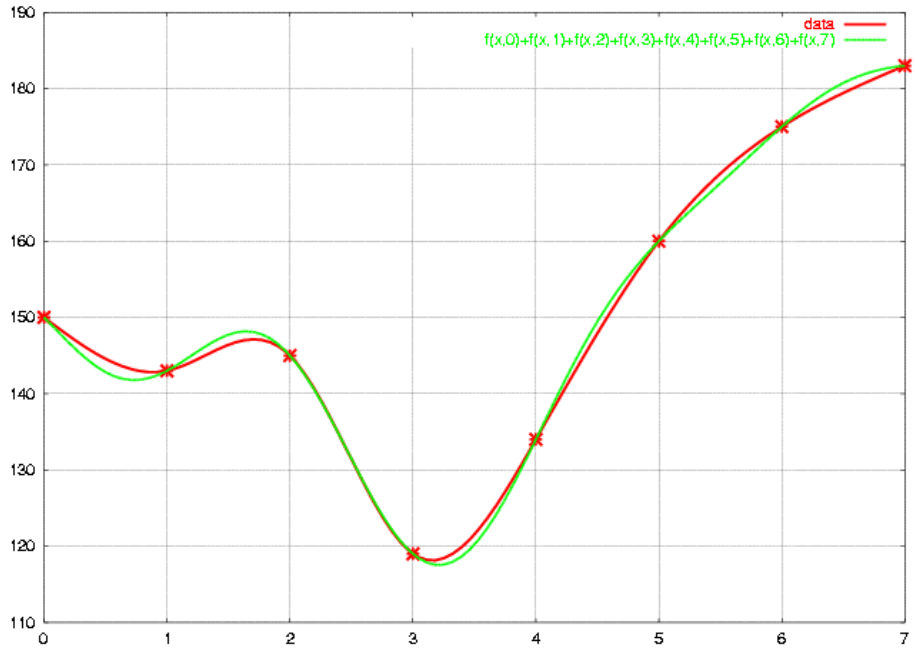


Approximation durch DCT-Koeffizienten (3)

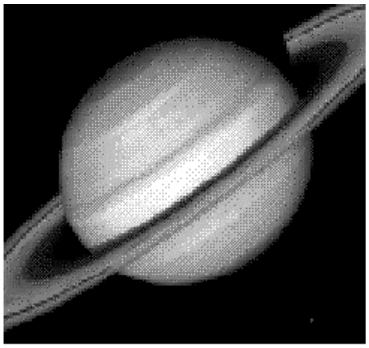
DC-Koeffizient + AC-Koeffizienten 1-3



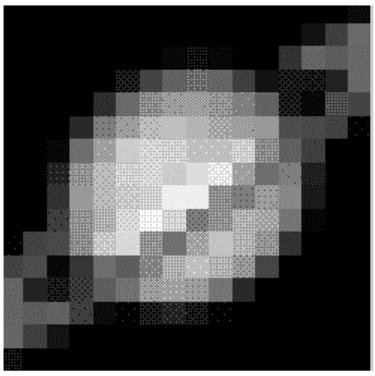
DC-Koeffizient + AC-Koeffizienten 1-7



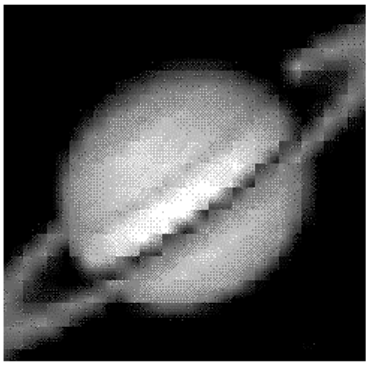
Beispiel: Kodierung eines Bildes mittels 2D-DCT auf 8x8-Blöcken



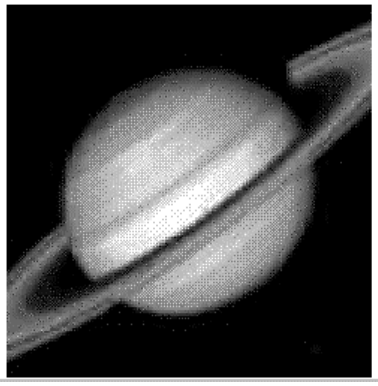
Original



1 Koeffizient



4 Koeffizienten



16 Koeffizienten

Quantisierung

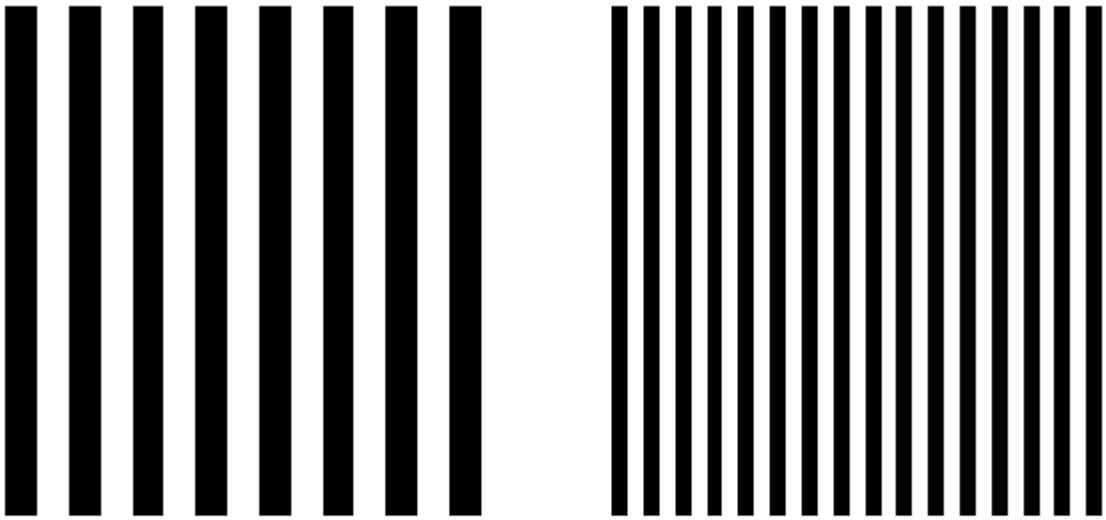
Anzahl der Quantisierungsstufen je DCT-Koeffizient einzeln einstellbar (Q-Faktor). Dies geschieht durch Angabe einer Quantisierungstabelle.

Entropie-Kodierung

1. DC-Koeffizient sehr wichtig (Grundfarbton). Codierung durch Differenzbildung zu vorherigem DC-Koeffizienten.
2. AC-Koeffizienten, Abarbeitung in „Zick-Zack“-Reihenfolge, Lauflängen-Kodierung
3. Huffman-Kodierung

Quantisierung: Wahrnehmungs-psychologischer Hintergrund (1)

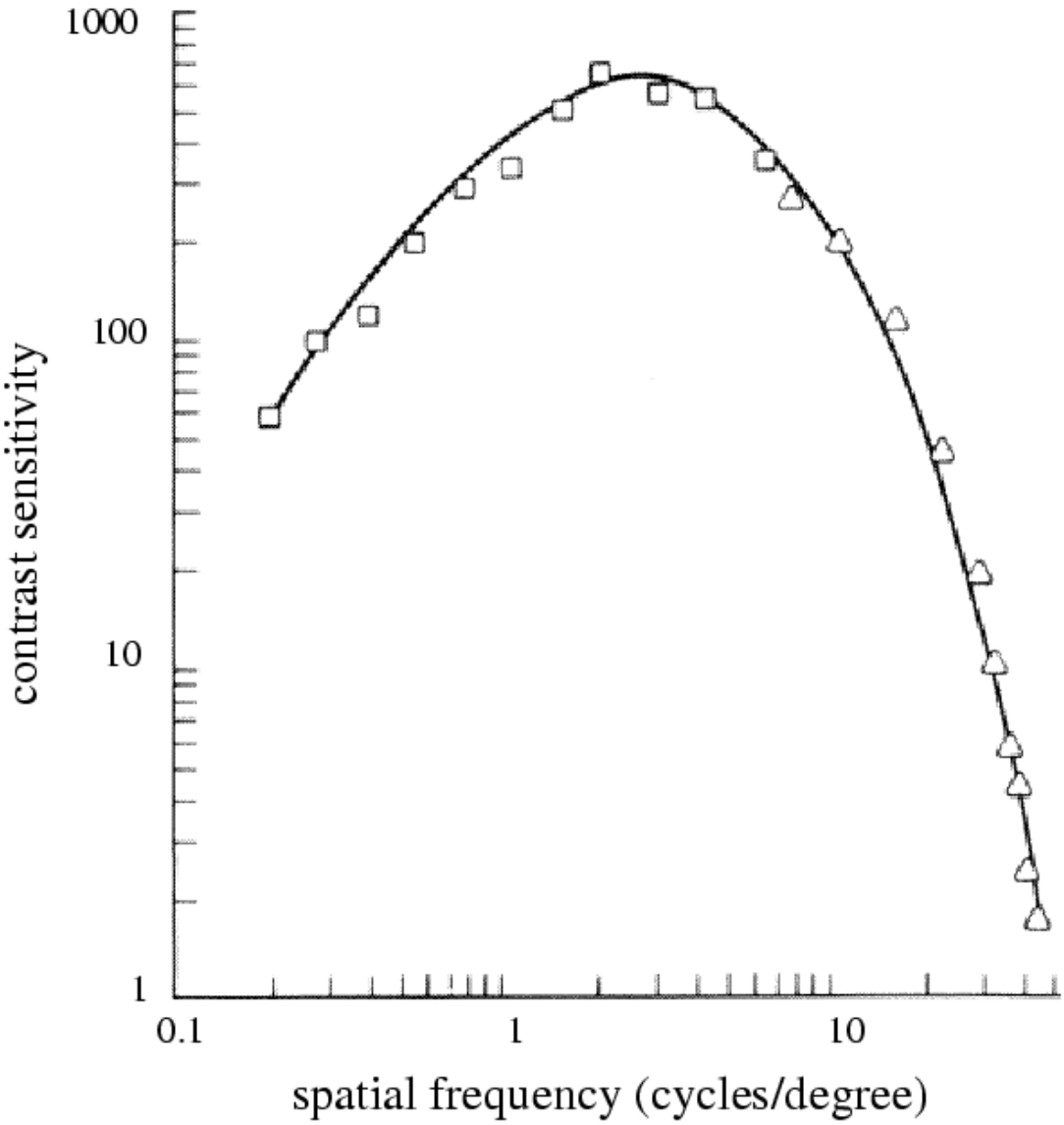
Einfluß der räumlichen Frequenz auf die menschliche Wahrnehmung von Kontrast.



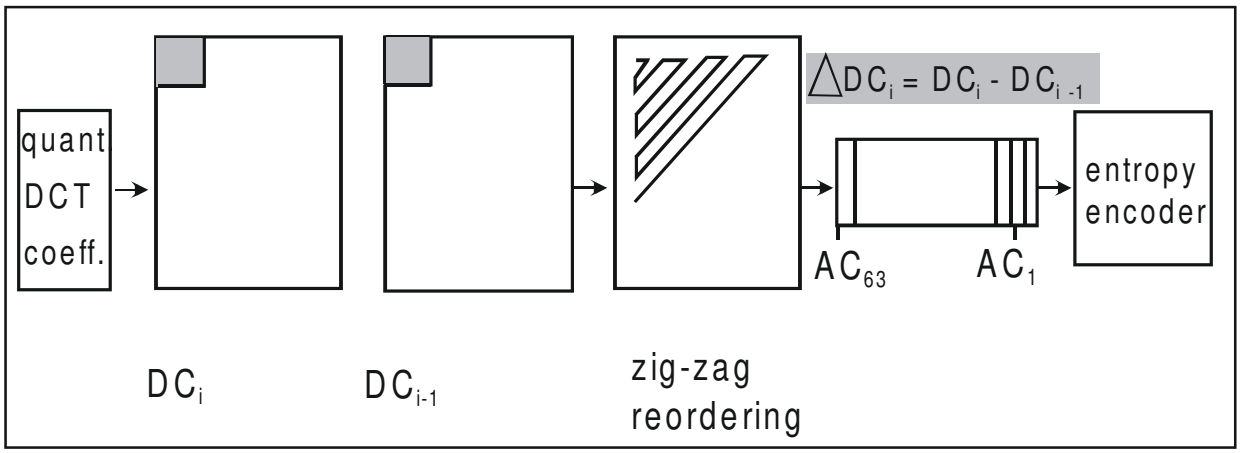
Ab einer bestimmten Distanz wirkt das rechte Muster kontrastärmer als das linke. Beide beinhalten jedoch dieselben Farben.

Quantisierung: Wahrnehmungs-psychologischer Hintergrund (2)

Kontrastempfindlichkeit der menschlichen Wahrnehmung in Abhängigkeit von der räumlichen Frequenz.

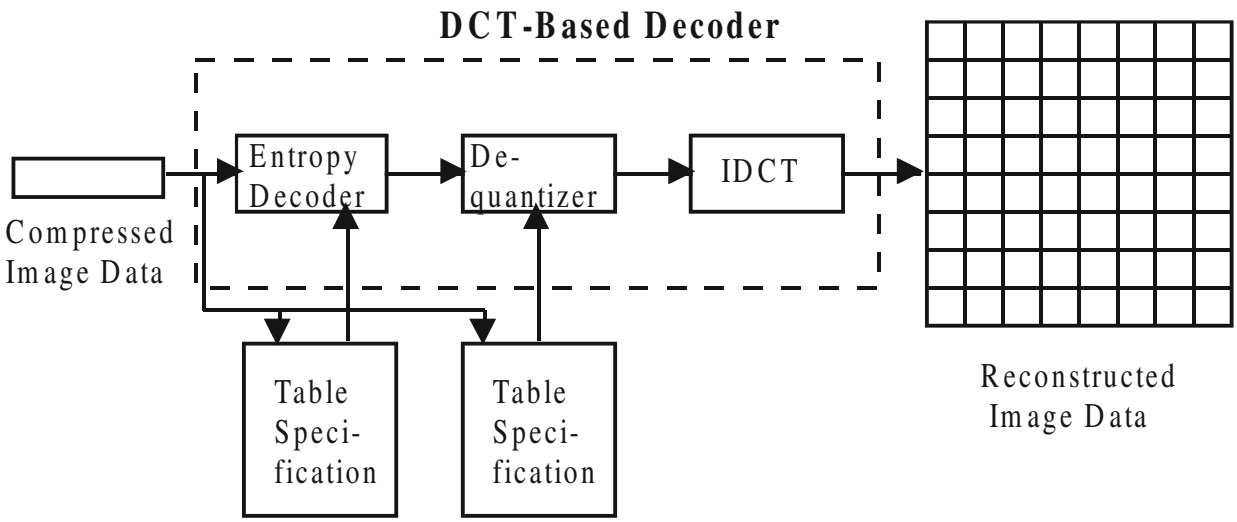


Quantisierung und Entropie-Kodierung



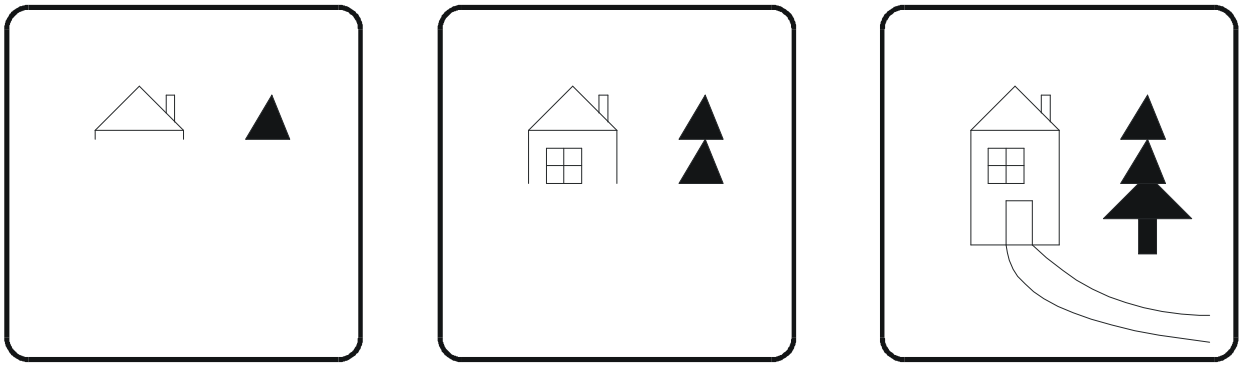
Das Zickzack-Auslesen der Koeffizienten ist besser als ein zeilenweises Auslesen. Denn: Nach der Quantisierung kommen in der rechten unteren Dreiecksmatrix viele gleiche Werte nacheinander vor, darunter viele Nullen (höhere Frequenzen entsprechen schärfer werdenden Kanten). Die Zickzack-Anordnung maximiert die Lauflängen für die anschließende Entropie-Kodierung (Lauflängen-Kodierung, gefolgt von der Huffman-Kodierung).

JPEG Decoder

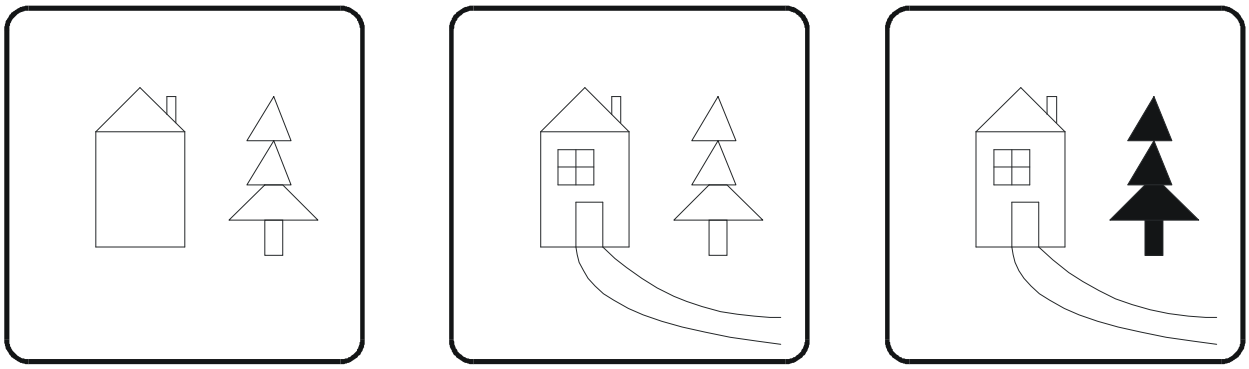


Verschiedene Modi in JPEG

JPEG Sequential Mode



JPEG Progressive Mode

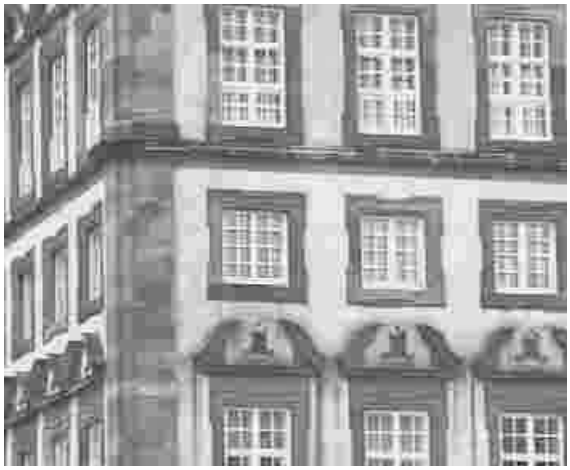


Quantisierungsfaktor und Bildqualität

Bildbeispiel Schloss

Schloss, Original

Schloss, Q=6



Bildbeispiel Schloss (Fortsetzung)

Schloss, Q=12

Schloss, Q=20

Bildbeispiel Blumen

Blumen, Original

Blumen, Q=6

Bildbeispiel Blumen (Fortsetzung)

Blumen, Q=12

Blumen, Q=20

2.2.5 Kompression mit Wavelets

Motivation

Signalanalyse und Signalkompression.

Bisher gesehen: Bildkompressionsverfahren

- direkt auf Pixelwerten beruhend (BTC; CCC; XCCC)
- auf Transformation in Frequenzraum beruhend (Fourier-Transformation, DCT)

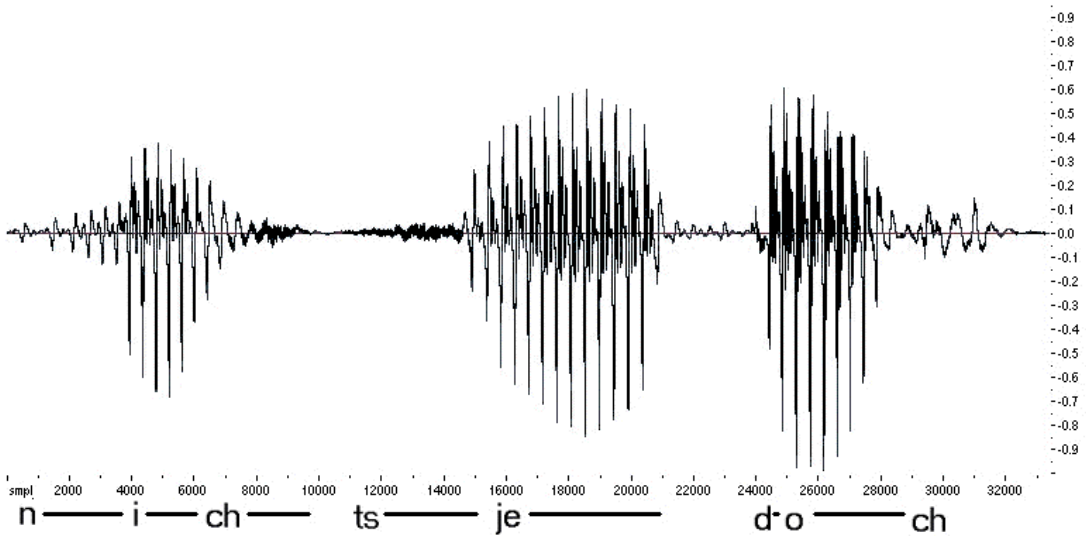
Was ist eine Transformation?

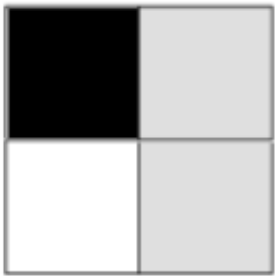
- Mathematisch: Basiswechsel
- Inhaltlich: Darstellung “auf andere Art und Weise”

Beispiel

“normale” Darstellung eines Signals:

- Audiosignal als Frequenzen über der Zeit
- Bild als Pixelwerte an Pixelplätzen





=

0	224
255	144

Das ist auch nicht das “eigentliche” Signal, aber man hat sich daran gewöhnt.

Frequenzraum

In der Darstellung, die wir gewohnt sind, handelt es sich um eine Darstellung eines Wertes in der Zeit/im Ort. Dies bezeichnet man allgemein als Zeitraum.

Im Frequenzraum fragt man sich nach der *Änderung* eines Signals.

- Wie stark variiert die Amplitude eines Audiosignals?
- Wie stark variiert der Übergang von einem Pixelpunkt zum nächsten?
- Welche Frequenzen sind in dem gegebenen Signal vorhanden?

Rückblick: Fourier-Transformation

Wir erinnern uns:

Fourier-Transformation von f:

$$\hat{f}(t) = \int f(x) e^{-2\pi i t x} dx$$

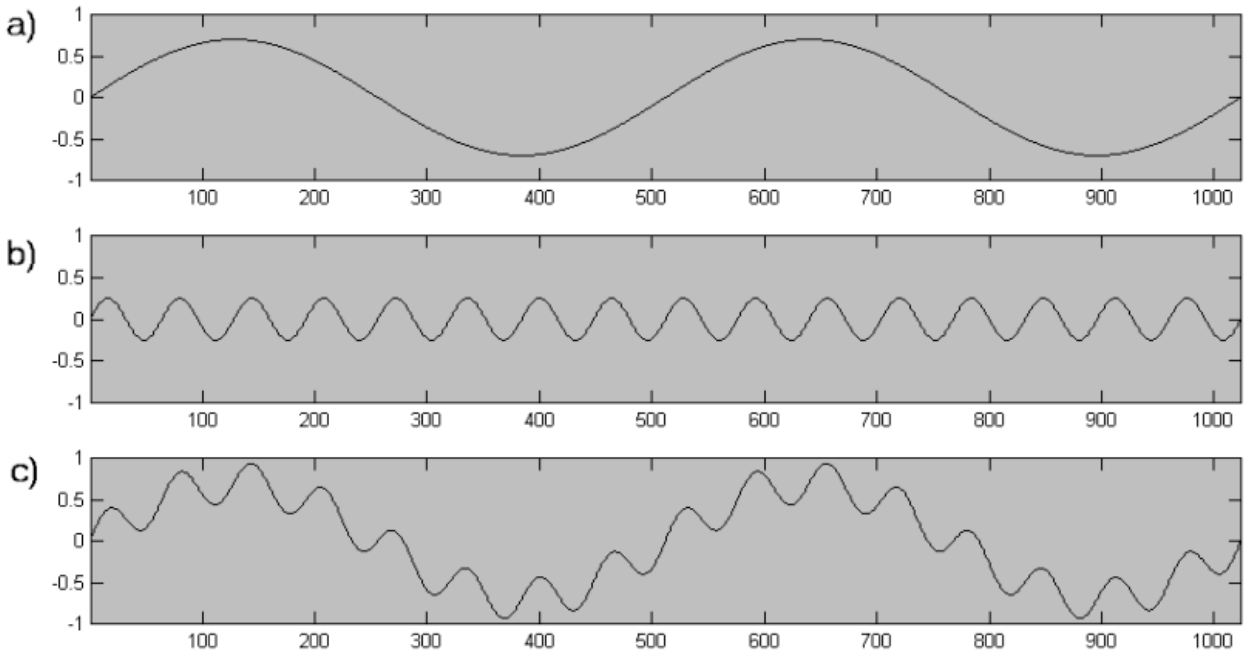
und Funktion e lässt sich schreiben als

$$e^{ix} = \cos(x) + i \sin(x)$$

Sinus und Cosinus sind bekannt: sie gehen von $-\infty$ nach ∞ .

Geeignete Transformation

Eine Transformation fragt danach, wie stark einzelne Frequenzen gewichtet werden müssen, damit die Summer aller dieser Frequenzen das Ausgangssignal darstellt.

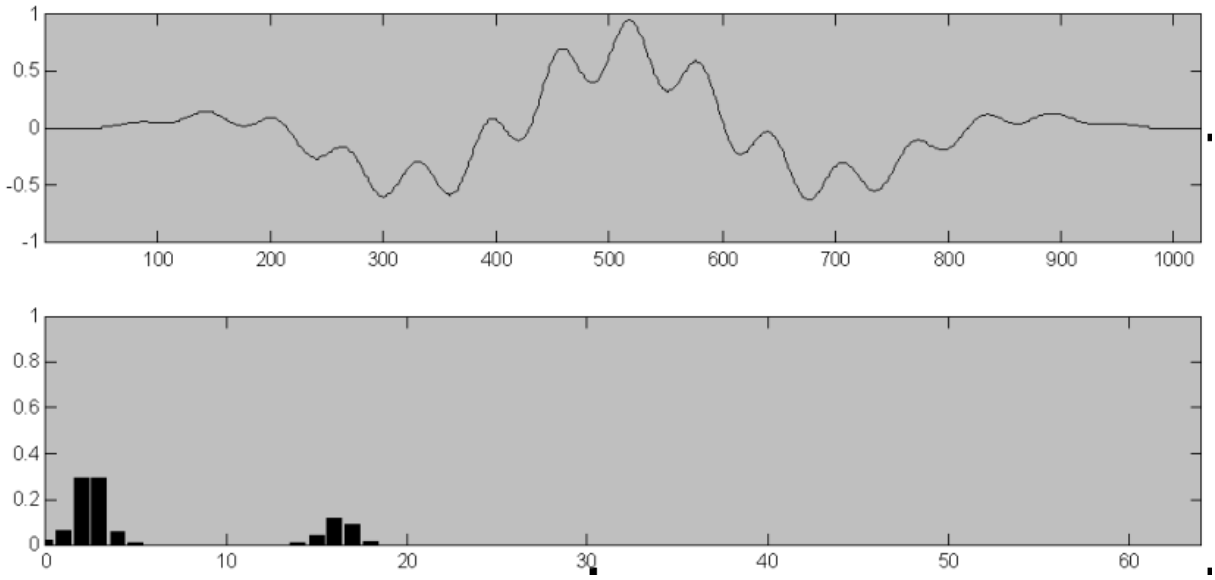


Das Ausgangssignal (c) wird dargestellt als Summe der beiden Sinusschwingungen (a) und (b).

Nebenbei bemerkt:

In JPEG wird die DCT verwendet wird - und nicht die FT - weil in $e^{ix} = \cos(x) + i\sin(x)$ der Sinus die FT "komplexwertig" macht. Wird nur Cosinus benutzt, so bleibt alles reellwertig.

Problem der Fourier-Transformation



Will man Signale darstellen, die eine hohe “Lokalität” aufweisen, so muss man sehr viele Sinus und Cosinus-Schwingungen aufaddieren. Das Beispiel stellt ein Signal dar (oberes Bild), das an den Rändern verschwindet. Es setzt sich zusammen aus Sinus-Schwingungen von 0-5 Hz und 15-19 Hz (unteres Bild).

Gesucht: Frequenzdarstellung durch Funktionen, die eine sehr hohe Lokalisierung aufweisen. Denn dann würde sich das Signal im Frequenzraum mit einigen wenigen Summanden unterschiedlicher Frequenzen darstellen lassen.

Lösung: Wavelets !

Was ist ein Wavelet?

Ein Wavelet ist eine Funktion ψ ,welche die Zulässigkeitsbedingung

$$0 < c_\psi := 2\pi \int_R \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$$

erfüllt.

Daraus folgt:

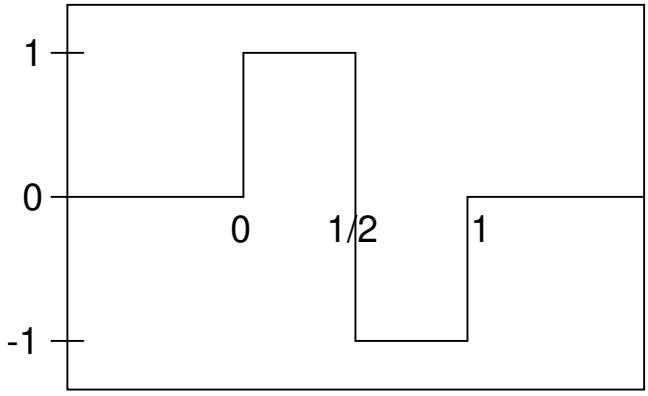
$$0 = \hat{\psi}(0) = \int \psi(x) e^{-2\pi i 0 x} dx = \int \psi(x) dx$$

Ein Wavelet ist somit eine Funktion, die nur auf einem begrenzten Intervall <> 0 ist, und die “über der Kurve” genau so viel hat wie “unter der Kurve”.

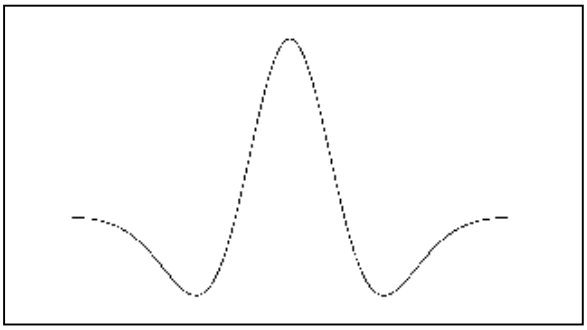
wavelet = small wave	(engl.)
ondelette = petite onde	(frz.)
Wellchen = kleine Welle	(dt.)

Beispiel-Wavelets

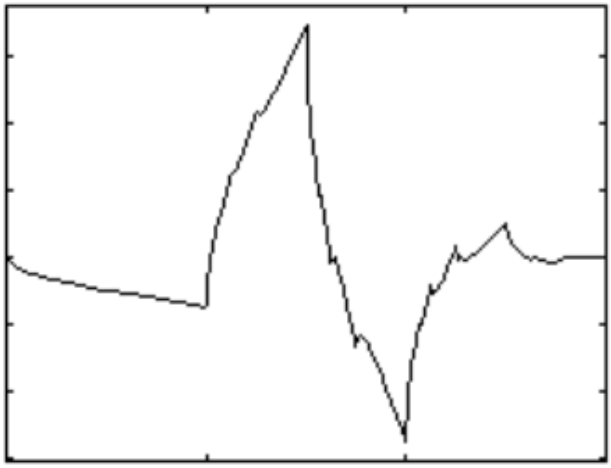
Haar-Wavelet



Mexican Hat



Daubechies-2



Praktische Anwendung

Einschränkungen:

Statt der allgemeinen Theorie betrachten wir nur noch

- diskrete Wavelet Transformationen (DWT)
- dyadische DWT, d.h. “Faktor 2”
- orthogonale Wavelets

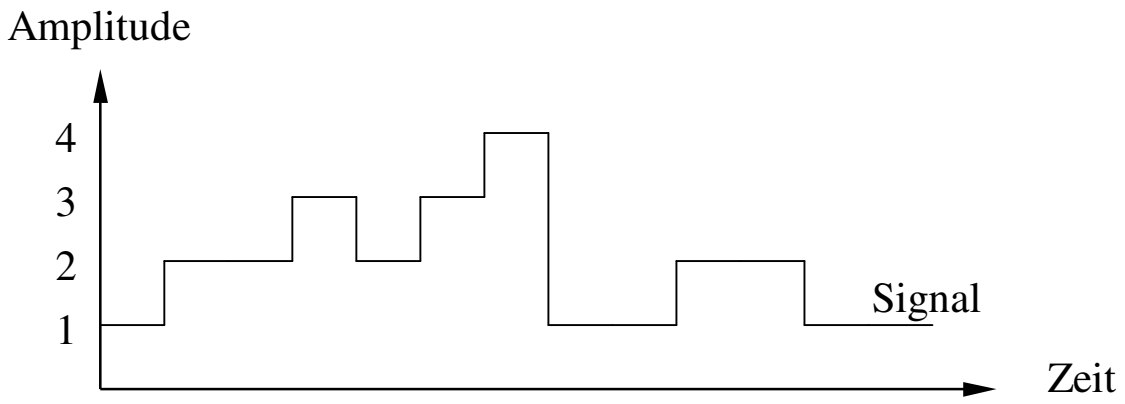
... damit wird alles jetzt ganz einfach und “Hands On” ...

Stéphane Mallat hat einen Zusammenhang hergestellt zwischen orthogonalen Wavelets und Filtern, die in der Signalverarbeitung und den Ingenieurwissenschaften schon lange verbreitet ist. Dies führt auf die Begriffe Hochpaßfilter (~Wavelet) und Tiefpaßfilter (~Scaling Function).

Beispiel: Haar-Transformation (I)

Wir führen jetzt eine Wavelet-Transformation mit dem Haar-Wavelet durch, ohne uns um die Theorie zu kümmern. Erst im Nachhinein werden wir den Zusammenhang zu dem Gelernten wieder herstellen...

Ziel: Zerlegung eines eindimensionalen Signals (z.B. Audio) in Wavelet-Koeffizienten.



(a) graphische Darstellung

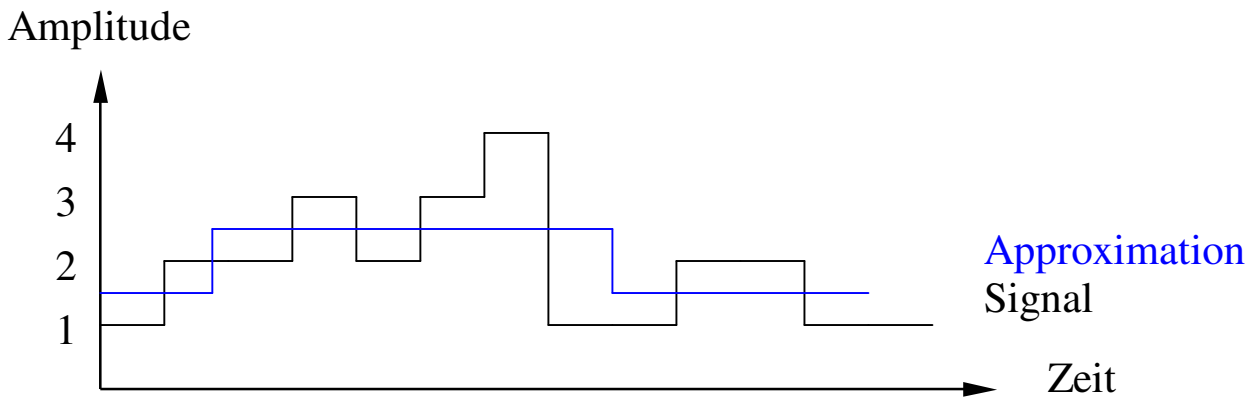
1	2	2	3	2	3	4	1	1	2	2	1	1

(b) Koeffizientendarstellung in der Zeit

Beispiel: Haar-Transformation (II)

Wie kann man das Signal anders darstellen, ohne dass Information verloren geht?

Eine gröbere Darstellung fragt (z.B.) nach dem Mittelwert zweier benachbarter Werte.



1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Approximation

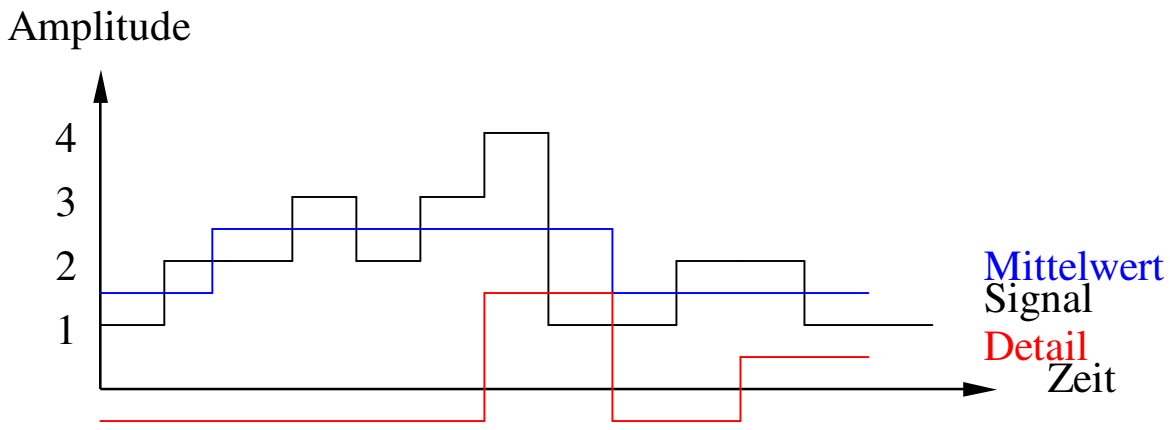
Filter zur Berechnung des Mittelwertes (Approximation):

1/2	1/2

Ein Filter wird “über das Signal gelegt”. Die Werte, die “aufeinander liegen” werden multipliziert, und alles zusammen addiert (→ Faltung).

Beispiel: Haar-Transformation (III)

Bei der Darstellung des Signals durch Approximationen ist Information verloren gegangen! Um das Signal aus den Approximationen wieder rekonstruieren zu können, müssen wir wissen, wie weit die beiden Werte, die einen Mittelwert gebildet haben, von diesem entfernt liegen.



1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Mittelwert
-0.5		-0.5		-0.5		1.5		-0.5		0.5			Detail

Filter zur Berechnung der Abweichung (Detail):

1/2	-1/2

Beispiel: Haar-Transformation (IV)

Wir haben unser Ausgangssignal zerlegt in eine andere Darstellung. Beachte: Die Anzahl der Koeffizienten, die man zur vollständigen Darstellung braucht, hat sich nicht verändert. (Das steckt auch hinter dem mathematischen Begriff "*Basis*-Transformation").

1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Mittelwert
-0.5		-0.5		-0.5		1.5		-0.5		0.5			Detail

Um das Ausgangssignal aus Approximationen und Details zu rekonstruieren, werden Synthesefilter angewandt:

1	1	Synthesefilter für 1. Wert
1	-1	Synthesefilter für 2. Wert

Damit:

1.5*1+(-0.5)*1 = 1 (Synthese des 1. Wertes)

1.5*1+(-0.5)*(-1) = 2 (Synthese des 2. Wertes)

2.5*1+(-0.5)*1 = 2 (Synthese des 1. Wertes)

2.5*1+(-0.5)*(-1) = 3 (Synthese des 2. Wertes)

usw.

Beispiel: Haar-Transformation (V)

Insgesamt haben wir also 4 Filter benötigt zur Zerlegung und Synthese des Ausgangssignals:

- Approximationsfilter für Mittelwert
- Detailfilter für Abweichung
- Synthesefilter für 1. Wert
- Synthesefilter für 2. Wert

1/2	1/2
1/2	-1/2
1	1
1	-1

Die Zerlegung des Signals in Approximationen und Details kann jetzt sukzessive mit den Approximationen als Eingangssignal fortgesetzt werden.

Bezeichnung:

Approximationsfilter =: Tiefpaßfilter

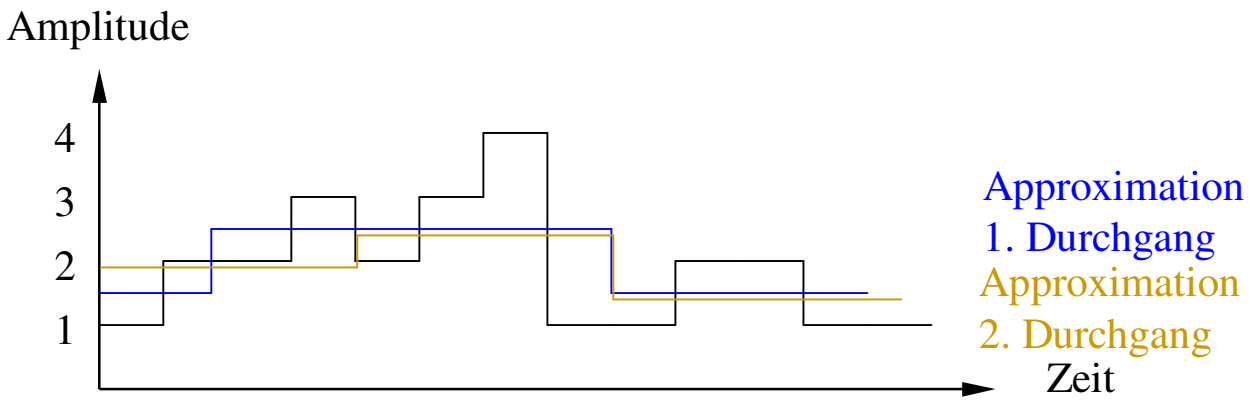
Detailfilter =: Hochpaßfilter

Betrachtung eines Signals in immer gröberen Auflösungen =: Multiskalenanalyse

Beispiel: Haar-Transformation (VI)

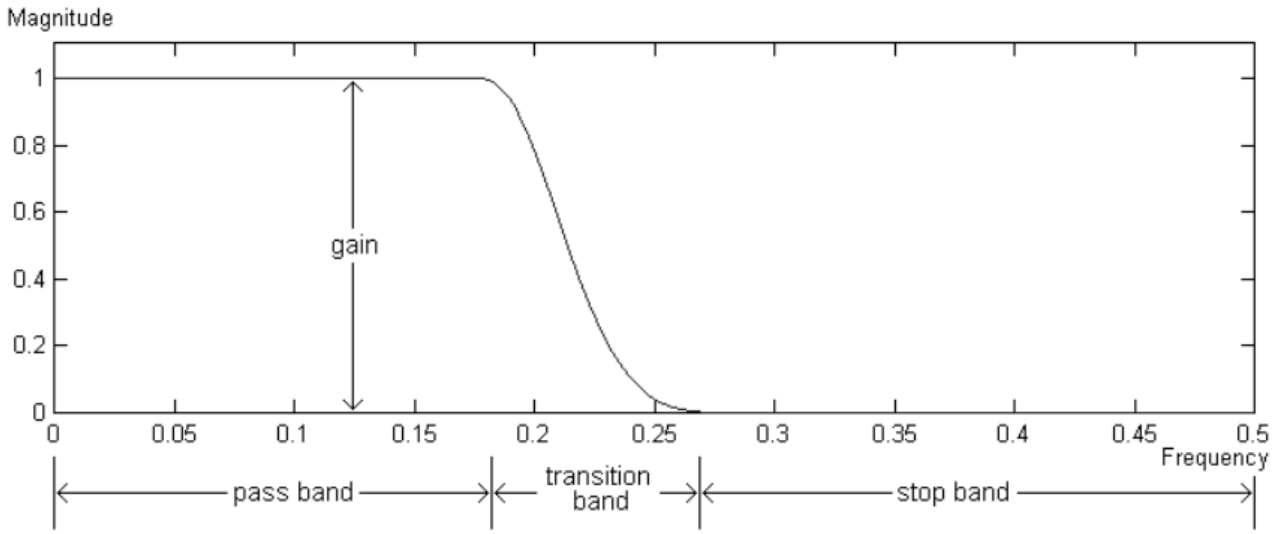
Rekursion mit jeweils errechneten Approximationen (Tiefpaß gefilterten “gröberen” Versionen des Ausgangssignals):

Speichere jeweils die Details (werden bei Synthese benötigt), und arbeite mit den Approximationen weiter.



1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Approx. 1
2				2.5				1.5					Approx. 2
2.25													Approx. 3

Hoch- und Tiefpaßfilter



Das Bild stellt einen Tiefpaßfilter dar. Ein Tiefpaßfilter lässt tiefe Frequenzen passieren (Multiplikation mit 1) und schneidet hohe Frequenzen ab (Multiplikation mit 0). Dass der Übergang vom “Pass Band” zum “Stop Band” in der Praxis oft nicht ganz streng ist, und es einen Frequenzbereich gibt, der weder gänzlich weggefiltert, noch gänzlich unverändert bleibt - das sogenannte Transition Band - soll uns hier mal nicht interessieren.

Ein Hochpaßfilter arbeitet genau anders herum.

Multiresolution Analysis

Wenn ein Signal (Eine Funktion, ein “Raum”) in sukzessive immer gröberen Skalen betrachtet wird (vgl. Beispiel Haar-Transformation), spricht man von Multiresolution Analysis.

Wir betrachten noch einmal

1	2	2	3	2	3	4	1	1	2	2	1	1	Signal
1.5		2.5		2.5		2.5		1.5		1.5			Approx. 1
2				2.5				1.5					Approx. 2
2.25													Approx. 3

Ein Koeffizient des Signals steht für einen Wert. Nach einem Durchgang der Wavelet-Transformation macht ein Koeffizient des Tiefpasses Aussagen für zwei Signalwerte. In Stufe 2 macht ein Koeffizient Aussagen für 4 Signalwerte etc.

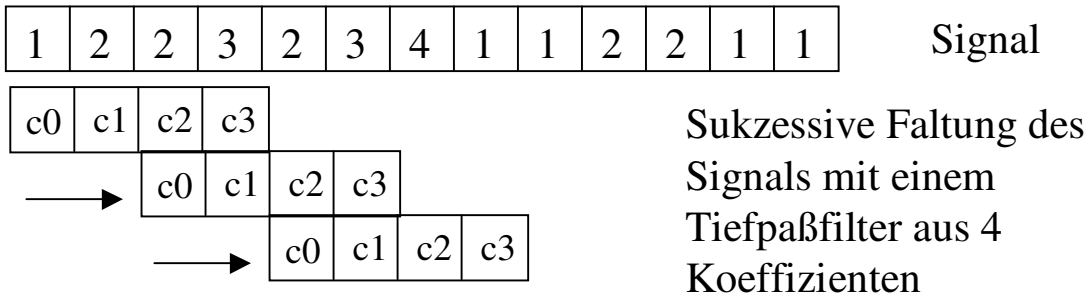
Die “Tragweite” eines Koeffizienten wird folglich in jedem Schritt gestreckt. Wir machen immer das gleiche, aber in immer anderen Auflösungen.

Allgemeine Wavelet-Transformation

Wir haben die 4 Filter einer Haar-Transformation und -Synthese kennen gelernt.

1/2	1/2
1/2	-1/2
1	1
1	-1

Allgemeine Wavelet-Filter sind länger. Zu einer vollständigen Transformation braucht man aber *immer* einen Tiefpaßfilter, einen Hochpaßfilter, und zwei Synthesefilter. Diese Filter werden immer über das Signal gelegt und mit ihm gefaltet (d.h. Multiplikation und Addition). Anschließend werden die Filter um 2 Elemente des Signals verschoben.



Beachte: Bei allen Filtern mit Filterlänge > 2 entstehen Randwertprobleme!

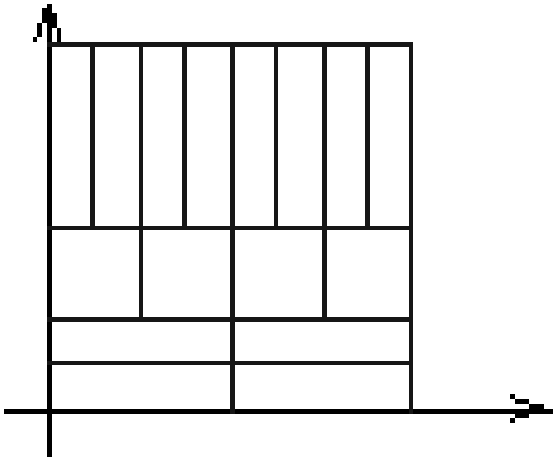
Einsatz von Wavelets: Audioanalyse

Die vorgestellte Diskrete Wavelet Transformation zerlegt ein Signal immer wieder in die Hälfte seiner Auflösung.

Ein Mensch hat einen hörbaren Frequenzbereich von 20 Hz bis 20KHz.

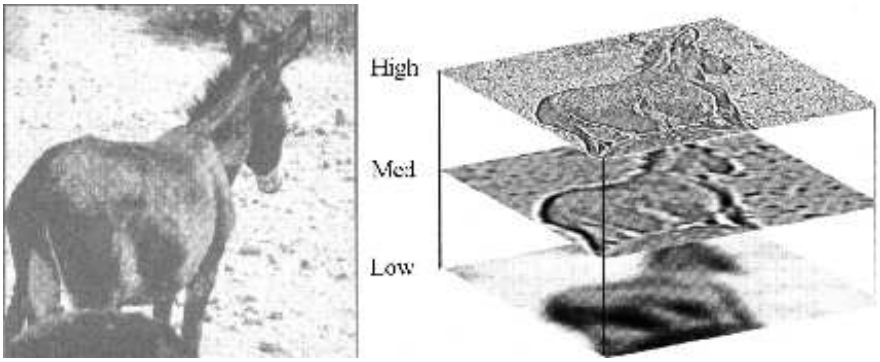
Audiobereiche werden logarithmisch wahrgenommen: Der Frequenzbereich von 100Hz bis 200Hz wird als gleich groß mit dem Bereich 5000Hz bis 10000Hz empfunden.

Eben diese Unterschiede bildet die Wavelet Transformation sehr schön nach.



Einsatz von Wavelets: Bildkompression

Es ist immer geschickt, ein Signal möglichst so darzustellen, wie ein Mensch es wahrnimmt.



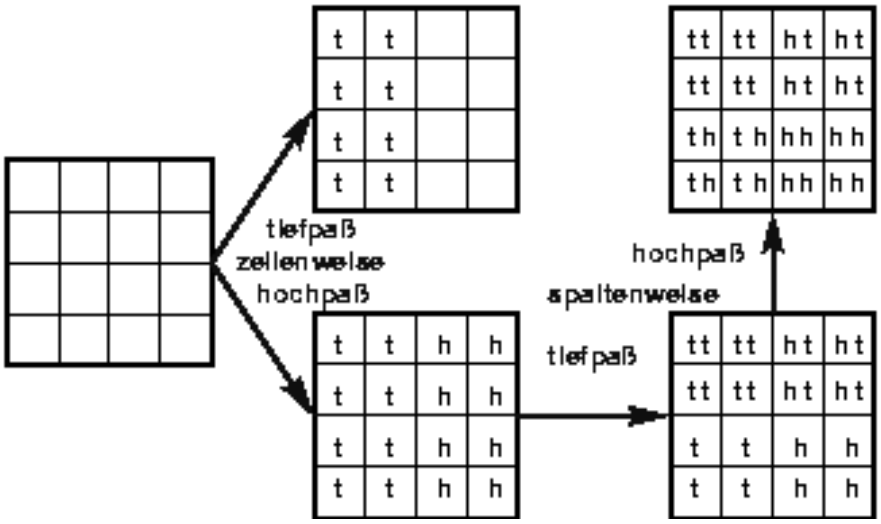
Aufteilung “von grob nach fein” ermöglicht es, ein Bild zunächst in seiner gröbsten Darstellung zu zeigen. Wenn Speicherplatz/Bitrate noch ausreichen, um weitere Verfeinerungen vorzunehmen, werden sukzessive Details mit hinzu gezogen.

Reicht aber Speicherplatz/Bitrate nicht aus, um ein Bild in voller Qualität (Verlustfrei) zu übertragen, so werden zumindest die für die Wahrnehmung wichtigsten Daten zuerst übermittelt!

Filter in mehreren Dimensionen

Anwendungen von Wavelet-Transformationen auf Bilder (2-dim.) und Videos (3-dim) müssen sich im Gegensatz zu Audio (1-dim.) Gedanken machen, wie mehrdimensionale Filter aussehen können.

Betrachtet man wiederum nur einen kleinen Teilbereich, separable Wavelets, so kann man die ein-dimensionalen Filter nacheinander auf die einzelnen Dimensionen loslassen.



Im Bild wird ein Standbild zunächst zeilenweise Hochpaß- unf Tiefpaß gefiltert, und danach die Filterung auf die Spalten angewandt.

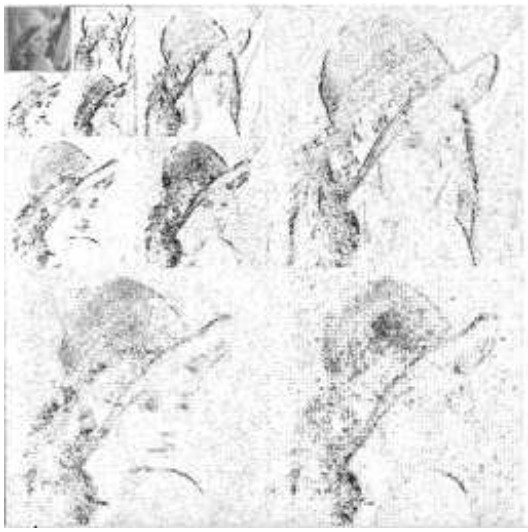
Bildkompression mit Wavelets (I)

Originalbild „Lenna“

Das 2-d Problem wird gelöst, indem man zunächst nur die Zeilen bearbeitet. Die Approximationen werden nach links geschrieben, die Details nach rechts.

Das zeilenweise gefilterte Bild wird als Ausgangsbasis für die spaltenweise Filterung genommen. Dadurch entstehen 4 Versionen innerhalb dieses einen kompletten Rekursionsschrittes.

Bildkompression mit Wavelets (II)



Während die Details gespeichert werden (d.h. nicht weiter verarbeitet), werden die Approximationen nun wiederum Tiefpaß- und Hochpaß gefiltert. Die entstehenden Details werden gespeichert, die Approximationen weiter verarbeitet...

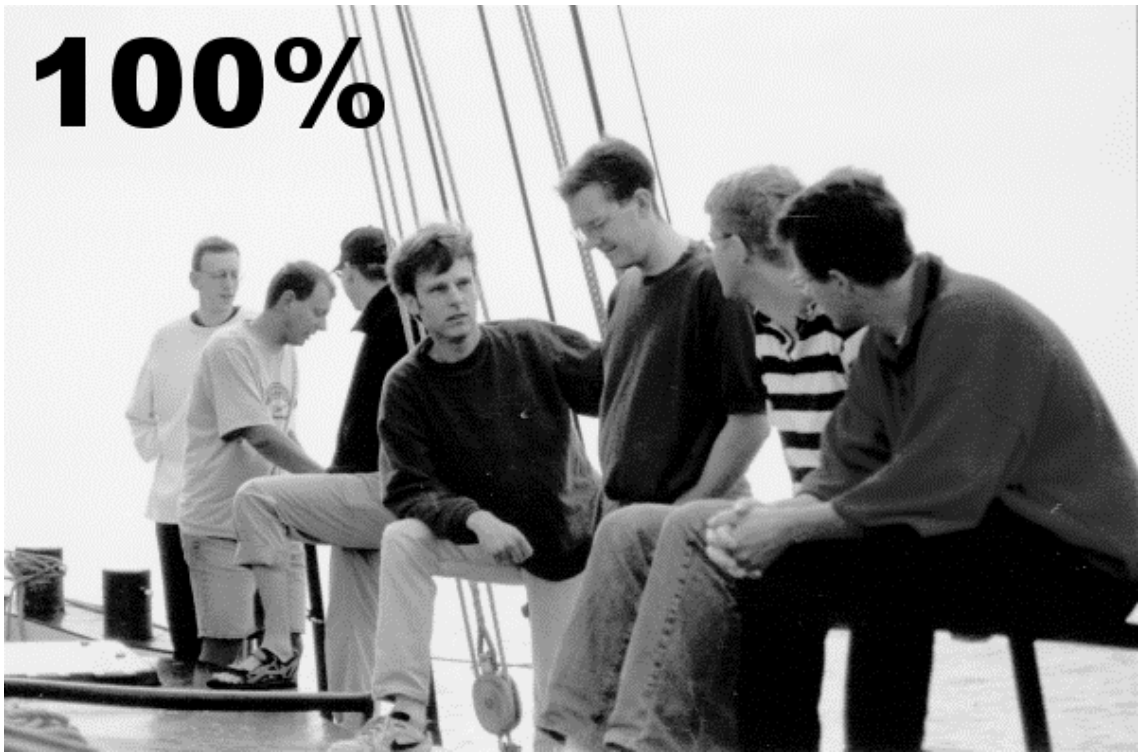
JPEG-2000 (I)

Der neue Standard JPEG-2000 beruht nicht mehr - wie JPEG - auf der DCT, sondern auf der Wavelet Transformation. Bei starker Kompression erscheinen die Artefakte (durch Informationsverlust) für den Menschen deutlich weniger störend, als die Blockbildung bei JPEG.

Es folgt:

Serie von Standbildern mit unterschiedlichen Kompressionen.

JPEG-2000 (II)

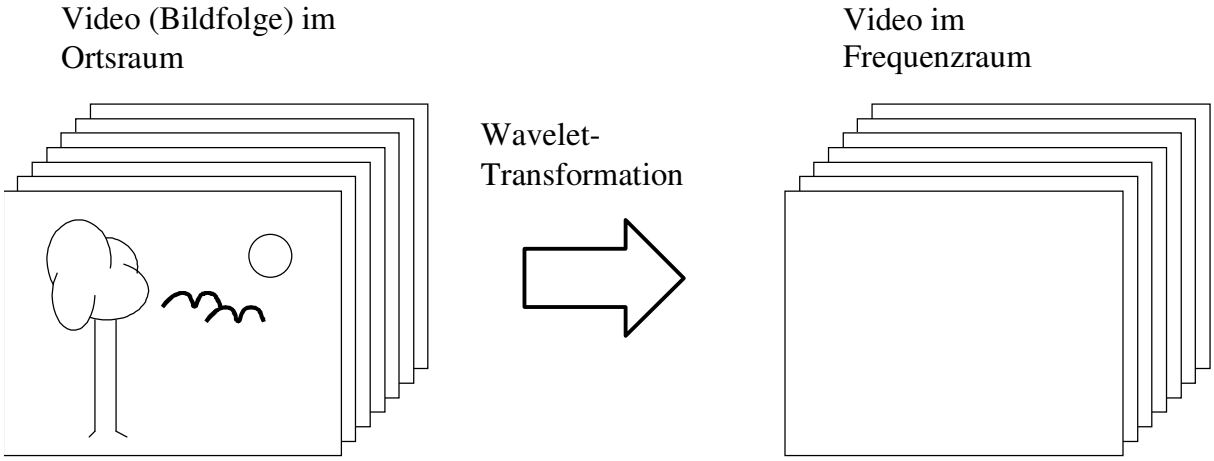


JPEG-2000 (III)

Videokompression mit Wavelets (I)

Ein Video ist ein drei-dimensionales Signal vorstellen (x-Achse, y-Achse, Zeit).

Typischerweise ist die Energie (Oszillation der Farbwerte in benachbarten Pixeln und Frames) des Signales im Ortsraum relativ gleichmäßig verteilt => keine gute Voraussetzung für die Kompression



Nach der Wavelet Transformation liegt das Signal ebenfalls in drei Dimensionen vor (Erinnerung: „Basiswechsel“), aber diese Dimensionen bedeuten überlagerte Frequenzen.

Hoffnung: Die Energie konzentriert sich in einem Bereich (→ gute Kompression).

Videokompression mit Wavelets (II)

obere Bildfolge: Originalvideo (Encodierung und Decodierung mit 100% der Wavelet Koeffizienten).
untere Bildfolge: Nur Koeffizienten oberhalb eines Grenzwertes wurden behalten. Kompression: ca. 1% der Information.

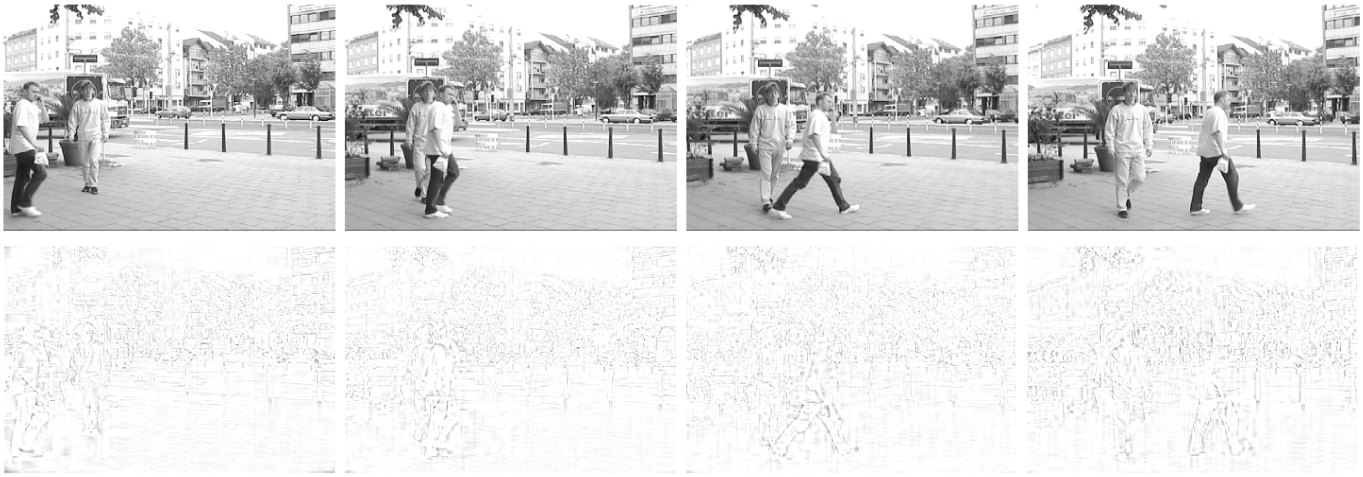
Folge: Für nur 1% der Information ist das Video noch gut zu erkennen.

Artefakte haben die Eigenschaften:

• unscharfe Einzelbilder, da viele hohe Frequenzen deutlich unter dem GW lagen. Hohe Frequenzen sind vor allem für scharfe Kanten und feine Texturen verantwortlich.

• überlagerte vergangene und zukünftige Frames, die teilweise als Schleier sichtbar sind. Grund: Es wurde nicht nur der Raum, sondern auch die Zeit transformiert.

Videokompression mit Wavelets (III)



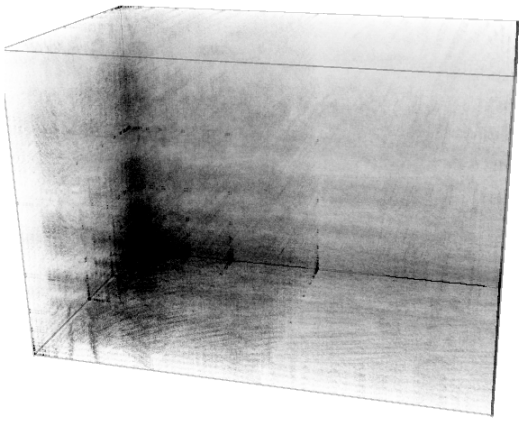
obere Bildfolge: Originalvideo (Encodierung und Decodierung mit 100% der Wavelet Koeffizienten).
untere Bildfolge: Darstellung der 99% verworfene Information.

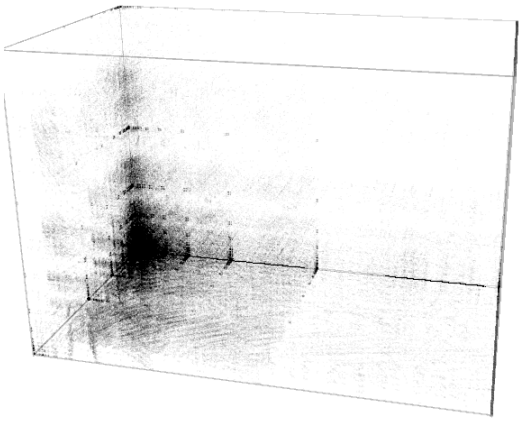
Die 99 % Wavelet Koeffizienten, die verworfen wurden, liefern die unten gezeigte Bild-Sequenz. Es wird deutlich, daß die meisten Koeffizienten eine für das Video unwesentliche Information enthalten.

Die hier gezeigte naive Transformation des Videos in den Frequenzraum bietet zwar starke Kompressionsmöglichkeiten (100:1), ist jedoch immer noch nicht besser als typische MPEG Kompressionsraten. Die Verwendung von Richtungsvektoren komprimiert Bilder immer noch besser. Trotzdem glaubt man, daß Wavelets das größere Potenzial für die Videokompression bieten.

Videokompression mit Wavelets (IV)

Darstellung einer Wavelet transformierten Videosequenz mit 150 Frames im Frequenzraum (besser: scale space). Je größer der Absolutbetrag des Koeffizienten, desto dunkler die Einfärbung.





Links: alle Koeffizienten. Rechts: 1% aller Koeffizienten sind größer als der Grenzwert.

Beobachtung: Tatsächlich clustern sich die hohen Werte um den Ursprung (linke hintere untere Ecke). Große Teile des Datenwürfels sind dagegen nur schwach besetzt.

Wavelet Transformationen - offene Fragen

Wir haben die Grundprinzipien der Wavelet Transformation vorgestellt. Abgesehen vom Anfang (Motivation und Exkurs Fourier Transformation) haben wir dabei die mathematischen Grundlagen gänzlich außer Acht gelassen.

Dadurch haben wir ein schön einfaches Kochrezept für die Anwendung von Wavelet Transformation bekommen (Stichwort: Filter und Faltung)

Erweiterungsmöglichkeiten gibt es noch zuhauf:

• diskrete WT versus kontinuierliche WT

• Art der Rekursion in mehreren Dimensionen (warum sollten nicht auch gemischte Hochpaß-Tiefpaß gefilterte Bereiche weiter zerlegt werden?)

• Wavelet für Audio ist gut verstanden, Wavelets für Standbilder ebenso. Beide legen unterschiedliche Gewichtung auf die Achsen, da Zeit etwas anderes ist, als x-Achse und y-Achse. Wie kann man diese Unterscheidung geschickt für Video ausnutzen?

• Konstruktion und Anwendung anderer als separabler Wavelets in höheren Dimensionen.

2.2.6 Bildkompression mit Fraktalen

Theorie der Fraktale = Theorie der **Selbstähnlichkeit**. Selbstähnlichkeit ist mathematisch beschreibbar! (B. Mandelbrot).

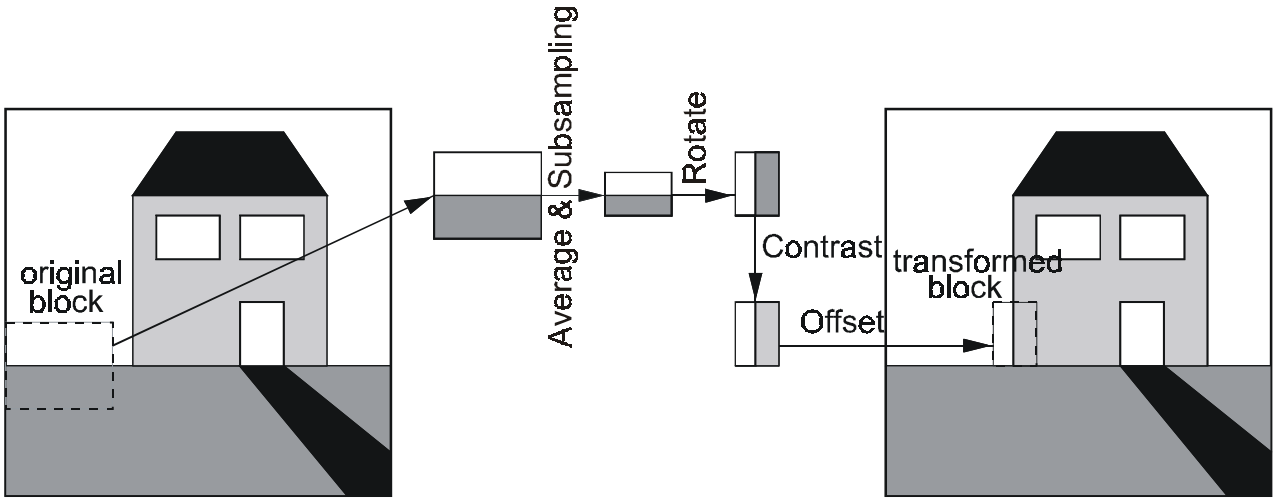
Beispiel aus der Natur: Küstenlinie einer Insel

Idee für die Bildkompression

- Häufig ist ein Ausschnitt aus einem Bild einem anderen Ausschnitt ähnlich. Genauer: Er kann unter Anwendung von einfachen mathematischen Operationen aus dem anderen Ausschnitt errechnet werden (Translation, Rotation, Skalierung)
- Kodierung: Voll-Kodierung des ersten Bildausschnitts, für die ähnlichen Bildausschnitte Ausgabe der **Transformationsoperatoren.**

Bildkompression mit Fraktalen

Beispiel



Literatur:

M.F. Barnsley, L.P. Hurd: Bildkompression mit Fraktalen, Vieweg-Verlag, 1996

2.3 Kompressionsverfahren für Video

2.3.1 MPEG

Moving Picture Experts Group (Komitee der ISO).

MPEG konzentriert sich nicht nur auf Videodaten, sondern behandelt auch die damit verbundenen Audiodaten. Ziel von MPEG-I: ein Videosignal soll samt Audio bei annehmbarer Qualität auf 1,5 Mbit/s komprimiert werden. (Datenrate einer T1-Verbindung in den USA, Datenrate eines Bitstroms von einer Audio-CD).

Entwurfsziele des MPEG-1-Videokompressionsalgorithmus

- Wahlfreier Zugriff innerhalb von 0,5 s ohne signifikanten Qualitätsverlust
- schnelle Vorwärts-/Rückwärtssuche
- Bildfolge kann rückwärts abgespielt werden
- gute Editierbarkeit

MPEG-Videokodierung

Vier Frame-Typen:

I-Frame (Intra Frame)

Intra-kodiertes Vollbild, sehr ähnlich dem JPEG-Standbild, kodiert mit DCT, Quantisierung, Lauflänge und Huffman

P-Frame (Predicted Frame)

Bezugnahme auf vorhergehende I- und P-Bilder.

Delta-Kodierung, DPCM-kodierte Makroblöcke

Bewegungsvektoren möglich

B-Frame (Interpolated Frame)

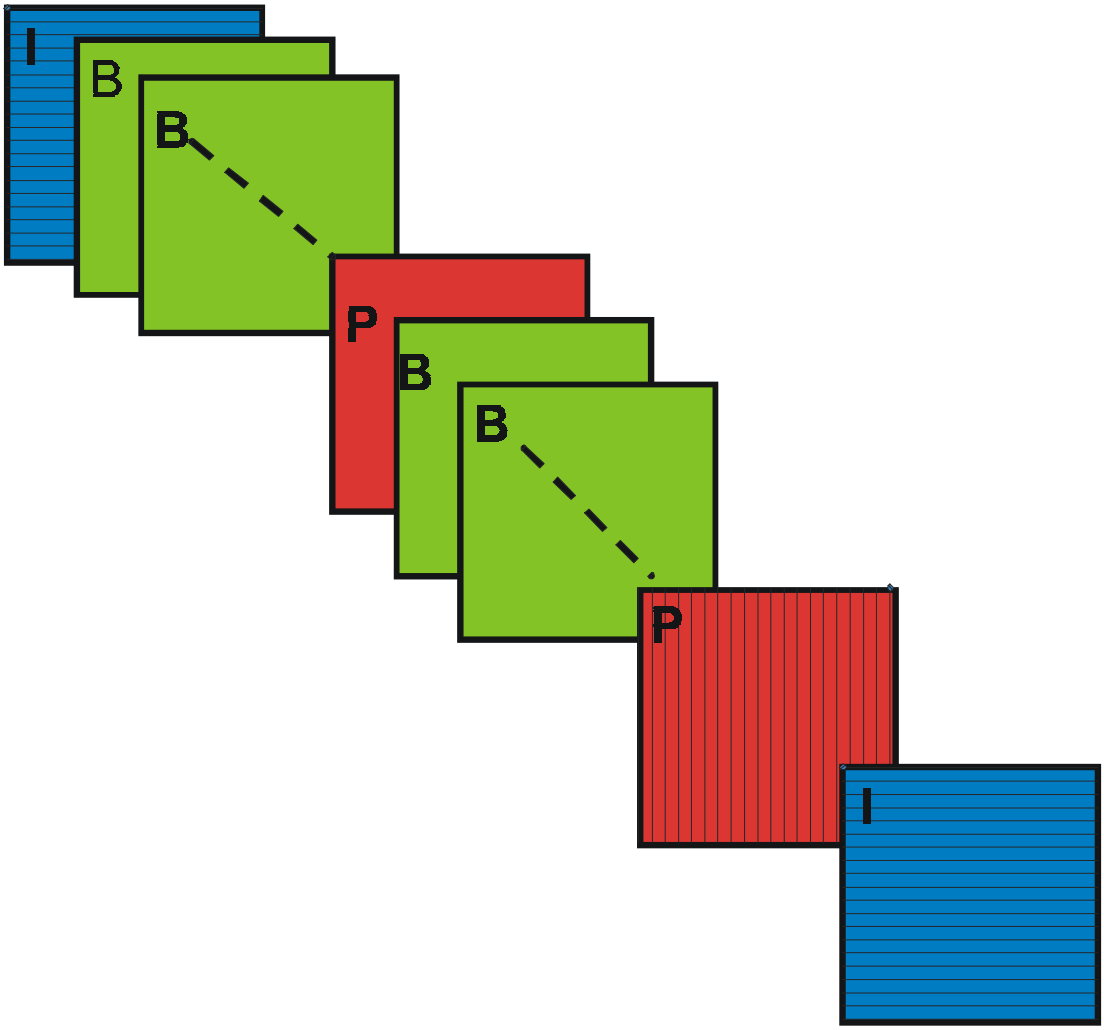
"bidirectionally predictive coded pictures", Bezugnahme auch auf nachfolgende Bilder, auch Interpolation zwischen Makroblöcken möglich

D-Frame

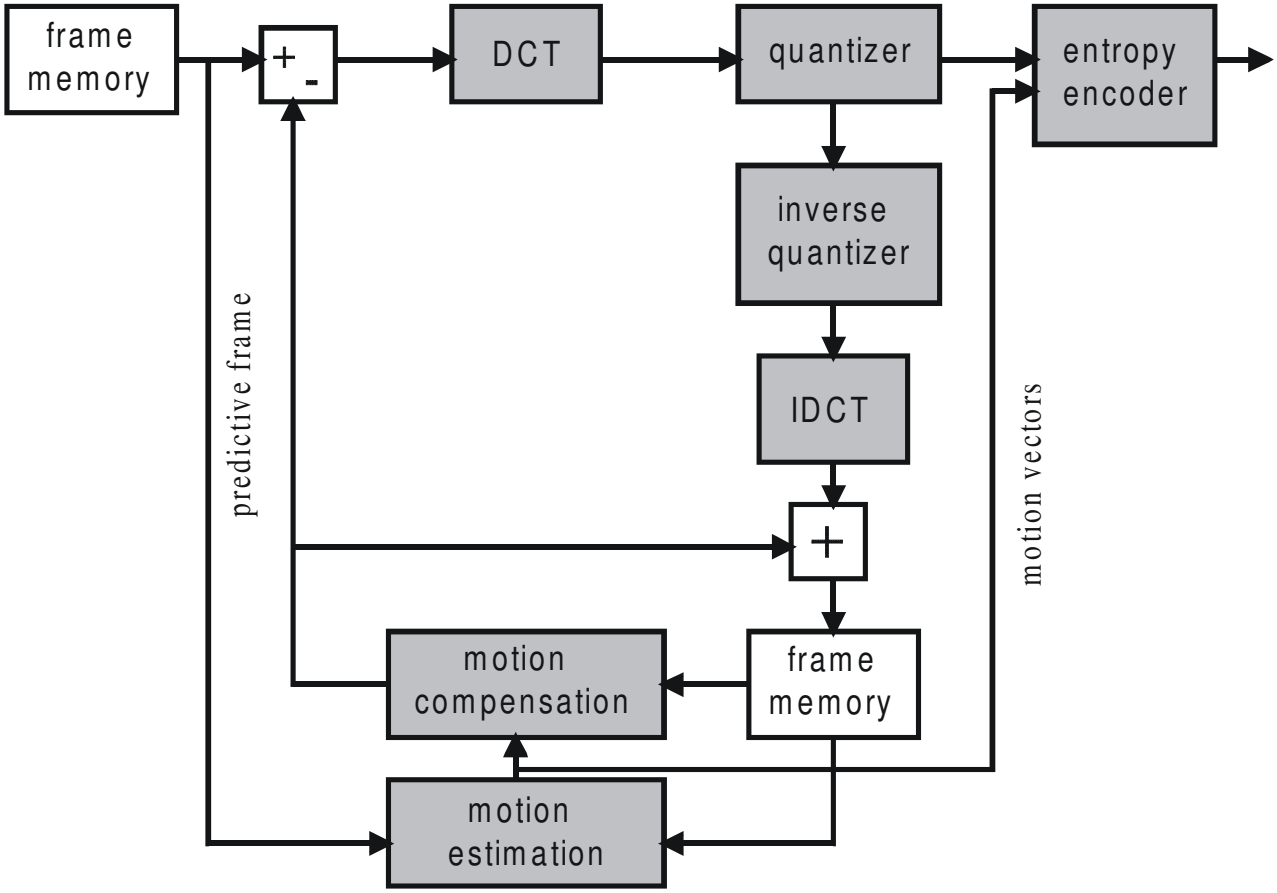
"DC coded picture", nur DC-Koeffizienten der Blöcke (linke obere Ecke der Koeffizienten-Matrix) für schnellen Vorlauf)

„Group of Pictures“ in MPEG

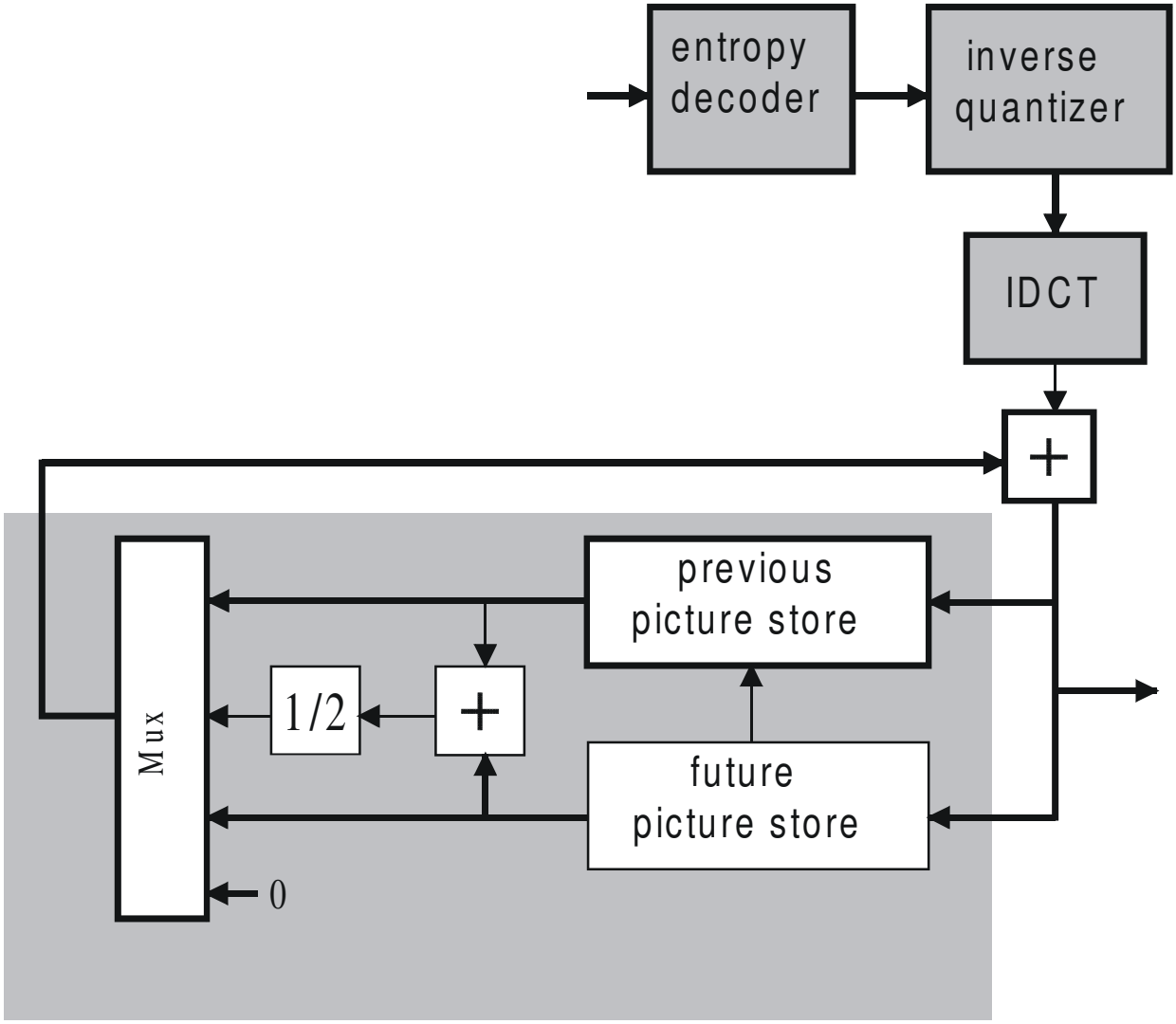
Die Folge von I-, P. und B-Frames ist nicht standardisiert, sondern kann von der Anwendung bestimmt werden. So lassen sich Bildqualität und Kompressionsrate anwendungsabhängig wählen.



MPEG-Encoder



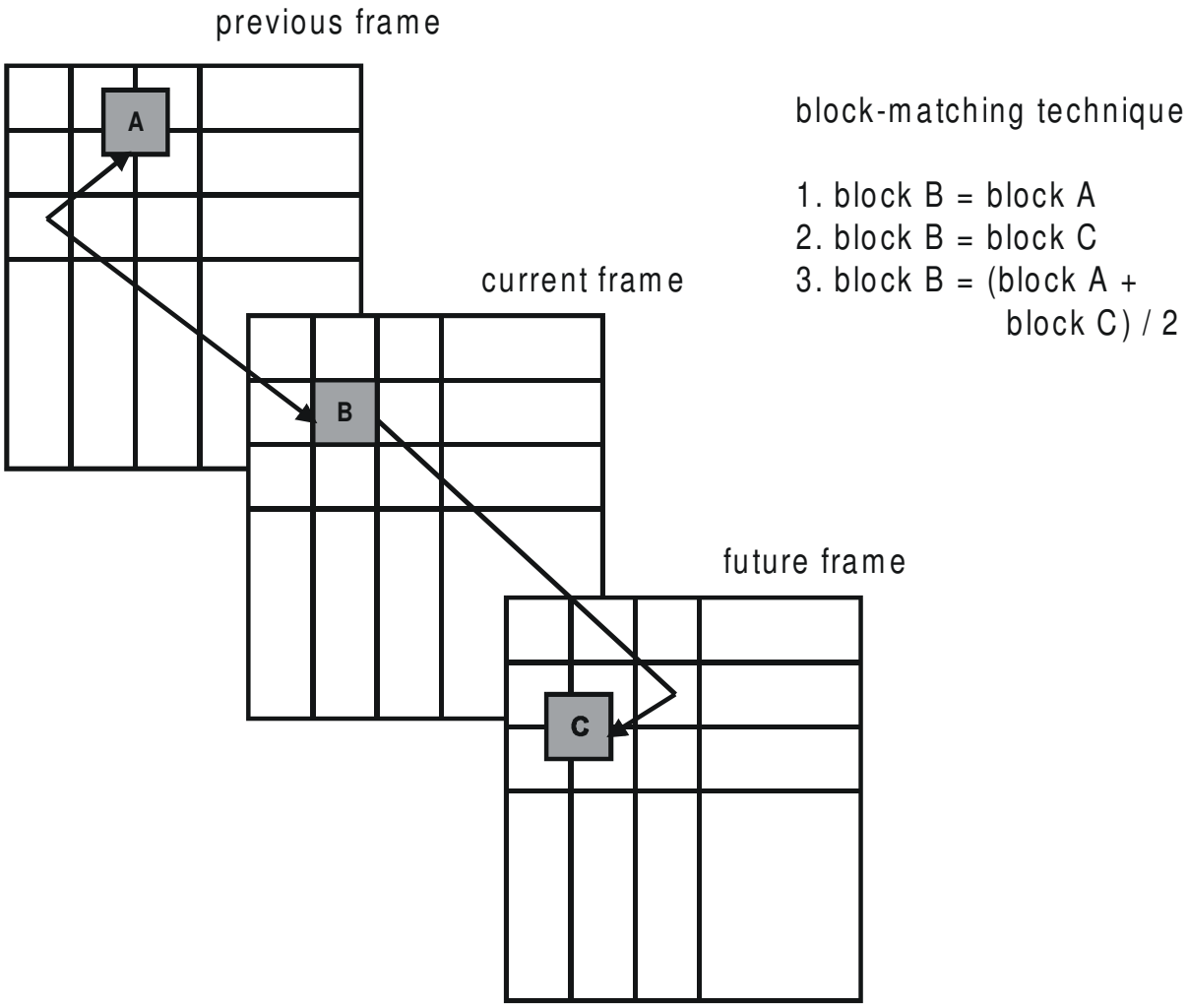
MPEG-Decoder



motion compensation

Zeitliche Redundanz und Bewegungsvektoren

"Motion Compensated Interpolation"



Der Suchradius (search range) kann bei der MPEG-Codierung als Parameter eingestellt werden. Je größer der Suchradius, desto besser das Kompressionspotential, aber desto länger auch die Laufzeit.

MPEG-2

MPEG-2 erweitert MPEG-1 für höhere Bandbreiten und bessere Bildqualität, bis hin zu HDTV. Es wurde gemeinsam von ISO und ITU entwickelt (H.262).

Erstmals sieht MPEG-2 auch skalierbare Datenströme vor, die zur Bedienung von Empfängern verschiedener Leistungsklassen eingesetzt werden können.

Skalierbarkeit in MPEG-2 (1)

- **"SNR scalability"** wird durch die Codierung jedes Bildes in mehreren Schichten ("layers") erreicht. Wer nur die Basis-Schicht dekodiert, erhält eine schlechte Bildqualität; wer auch höhere Schichten decodiert, erhält Schicht für Schicht eine bessere Bildqualität. Ein Beispiel ist die Übertragung der U- und V-Komponenten in der Basis-Schicht mit einem Viertel der Auflösung von Y, die höheren Schichten enthalten dann die fehlenden Farbwerte und ermöglichen eine bessere Farbtreue.

- **"Spatial scalability"** bedeutet eine Codierung des Bildes mit verschiedenen Pixel-Auflösungen. So könnten beispielsweise die Bildauflösungen für ein Standard-Fernsehgerät und für ein hochauflösendes HDTV-Fernsegerät in einem gemeinsamen Datenstrom übertragen werden.

Skalierbarkeit in MPEG-2 (2)

- **"Temporal scalability"** bedeutet die Übertragung von wenigen Frames/s in der Basis-Schicht und von zusätzlichen Frames/s in den höheren Schichten. Wer also die höheren Schichten decodiert, erhält eine höhere Frame-Rate.

- **"Data partinioning"** ermöglicht eine Skalierbarkeit bezüglich der Fehler-Resistenz. Die wichtigsten Bestandteile des MPEG-Stromes werden in der Basis-Schicht übertragen, weniger wichtige Teile in den höheren Schichten. In die Basis-Schicht legt man zum Beispiel die niedrigen Koeffizienten der DCT und die Bewegungsvektoren. Die Basis-Schicht kann dann mit mehr Redundanz zur Fehlersicherung versehen werden und wird somit fehlerärmer empfangen als die höheren Schichten, bei denen sich Übertragungsfehler weniger gravierend auswirken.

MPEG-2 Video-Profile

	Simple profile no B frames not scalable	Main profile B frames not scalable	SNR scalable profile B frames SNR scaling	Spatially scalable profile B frames spatial scaling	High profile B frames spatial or SNR scaling
High level 1920x1152x60		<=80 Mbits/s			<=100 Mbits/s
High-1440 level 1440x1152x60		<=60 Mbits/s		<=60 Mbits/s	<=80 Mbits/s
Main level 720x576x30	<=15 Mbits/s	<=15 Mbits/s	<=15 Mbits/s		<=20 Mbits/s
Low level 352x288x30		<=4 Mbits/s	<=4 Mbits/s		

Skalierbare Kodierung = Kodierung in Schichten (layers)

base layer: niedrigste Qualität

enhancement layers: Nachbesserung der Qualität

SNR scaling: enhancement layer enthält beispielsweise Chroma-Daten in höherer Auflösung

Spatial scaling: Erhöhung der Pixeldichte

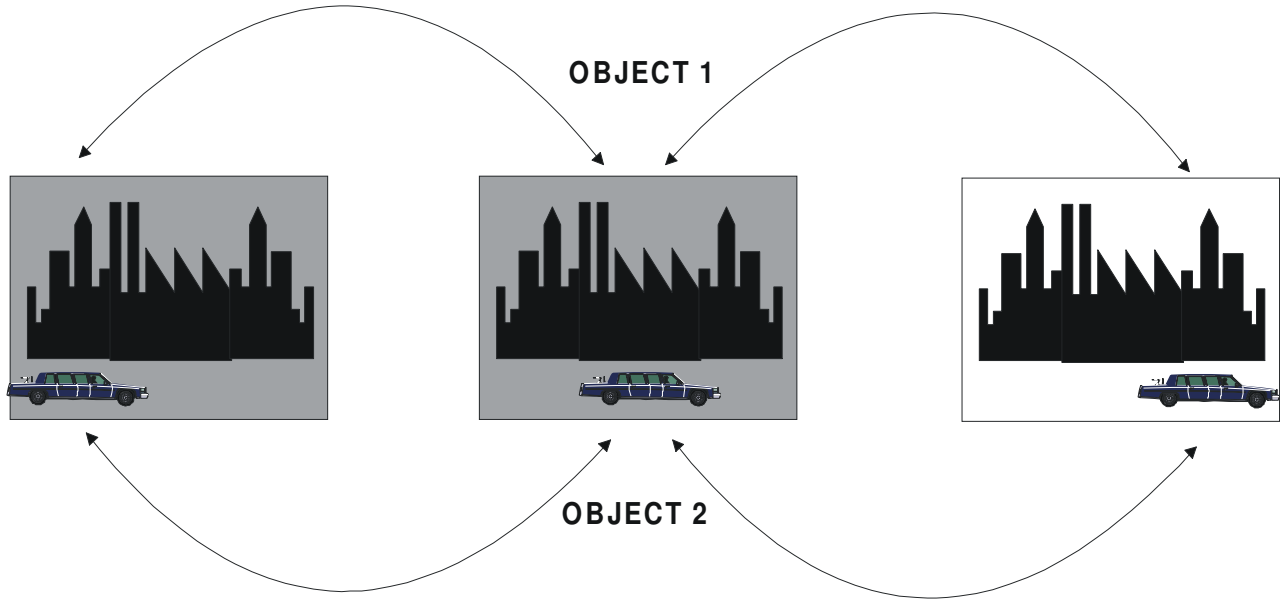
MPEG-4 (1)

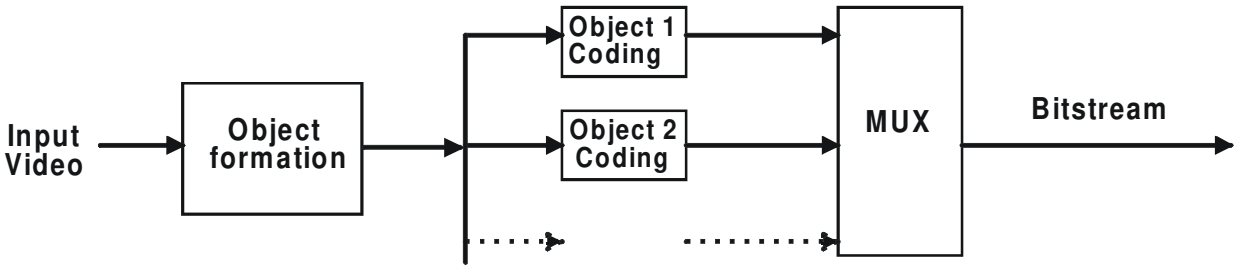
Ursprünglich planten ISO und ITU einen Standard MPEG-3 für HDTV-Qualität bei sehr hohen Datenraten. Dieser wurde später in MPEG-2 integriert und ist somit hinfällig.

Für MPEG-4 war zunächst an Video für sehr niedrige Bandbreiten gedacht (z.B. Multimedia für Mobilfunk). Inzwischen basiert MPEG-4 nach einer radikalen Kehrtwendung im Normungsgremium auf einer **Einzelobjekt-Kodierung**. Das Bild wird in einzelne Objekte zerlegt. Für jedes Objekt kann die Kodierungstechnik getrennt gewählt werden. Die objektorientierte Kodierung eröffnet zugleich völlig neue Möglichkeiten zur Weiterverarbeitung beim Empfänger.

MPEG 4 (2)

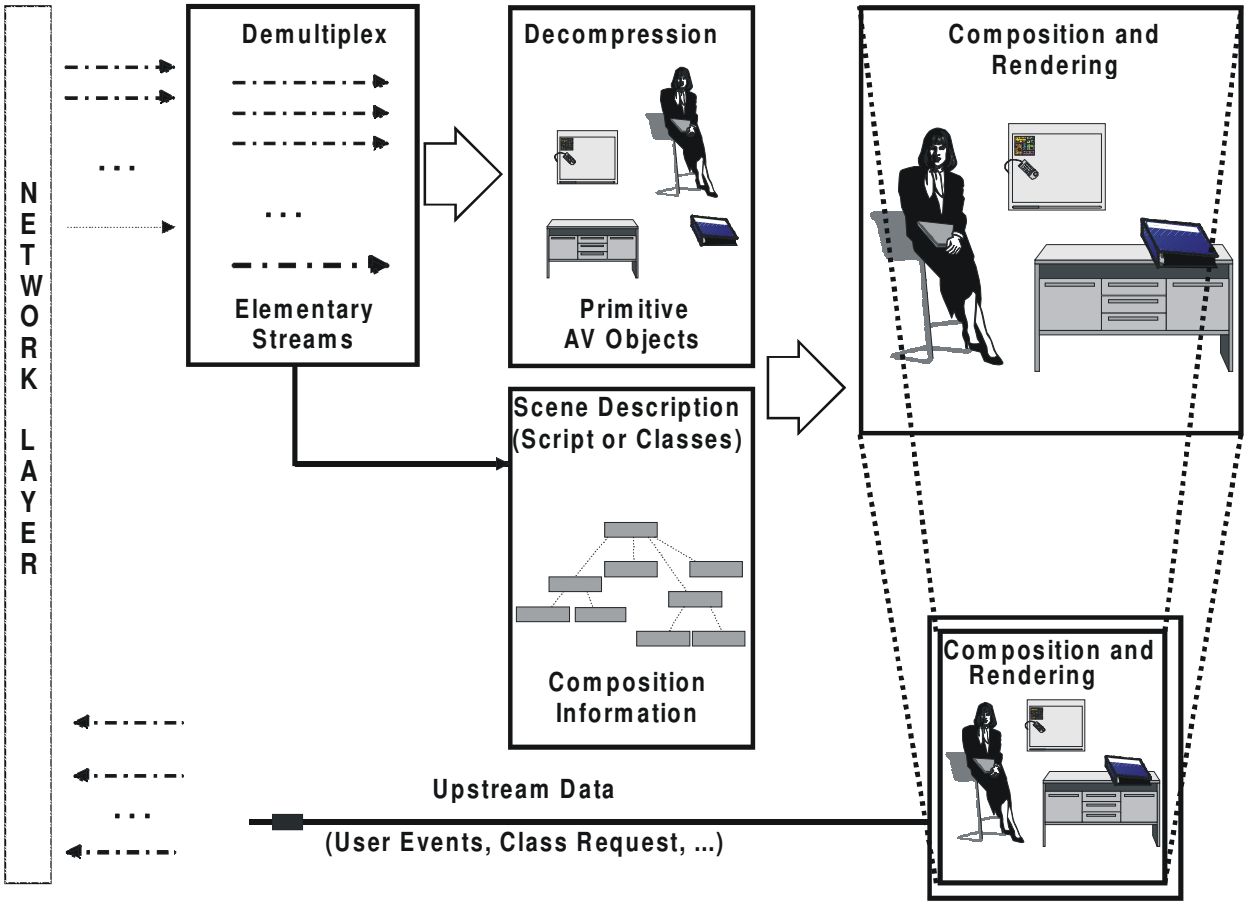
Getrennte Codierung des Hintergrund-Objektes und des Vordergrund-Objektes. Der Hintergrund ist statisch.





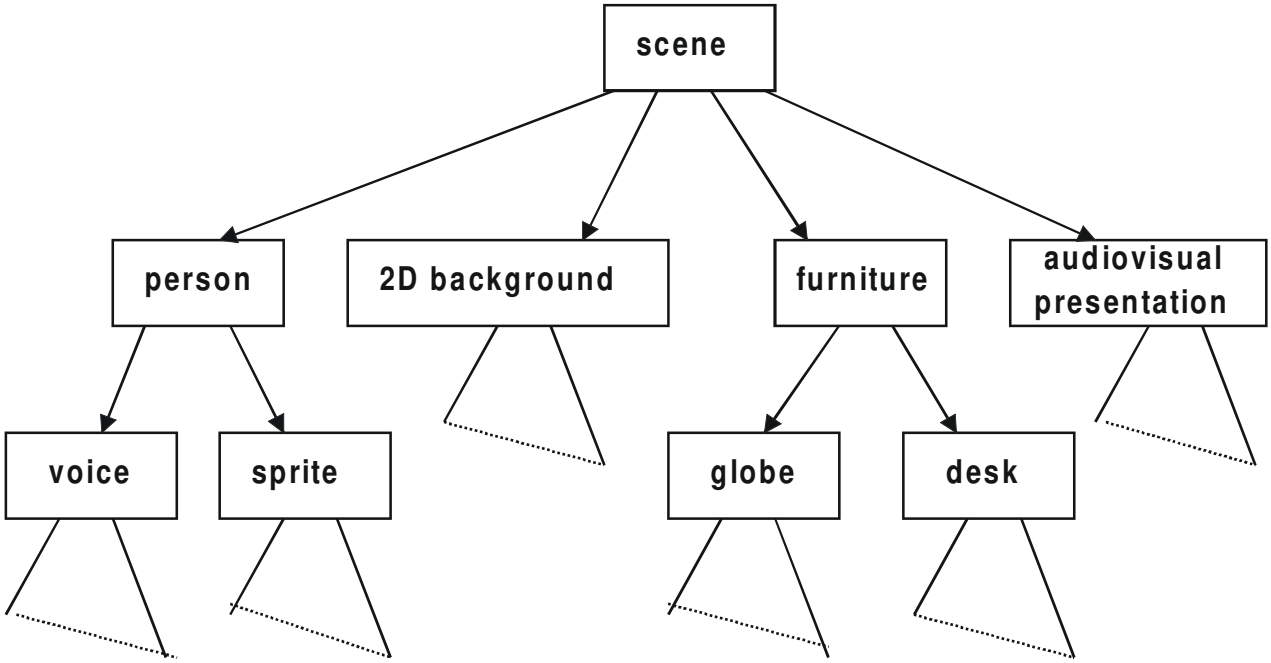
MPEG 4 (3)

Decodierung eines MPEG-4-Systemstromes



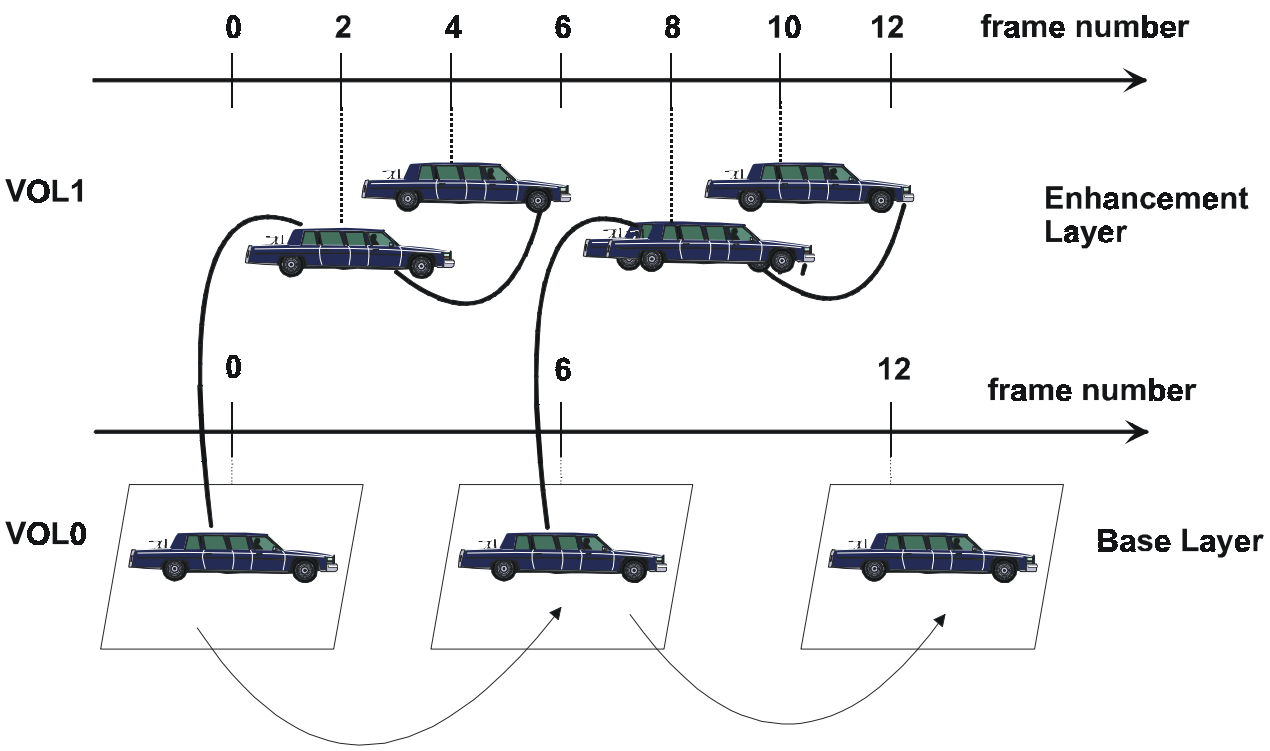
MPEG 4 (4)

Objekthierarchie zum Decodierungsbeispiel



MPEG 4 (5)

Skalierung durch „layered encoding“ in MPEG-4



2.3.2 ITU Recommendation H.261

Auch bekannt als „p*64kbit/s“

- Ein Verfahren zur Videokodierung für audiovisuelle Dienste bei einer Bitübertragungsrate von p x 64 kbit/s. Nur für Video, nicht für Audio.
- Ursprünglich entwickelt für ISDN
- Für den Parameter p gilt:
  - p ist aus [1,30]
  - p klein (p=1, p=2): Bildtelefon (schlechte Qualität)
  - p groß (ab p=6): auch Videokonferenz (mittlere Qualität)
- heute gebräuchlich: p=2, p=6
- Intraframe-Coding: DCT-basiert (wie JPEG)
- Interframe-Coding: ähnlich den P-Frames in MPEG-1

Videokodierung in H.261

Die fundamentalen Ideen sind denen in JPEG und MPEG gleich. H.261 ist weniger flexibel, dafür aber leichter standardkonform implementierbar.

Es werden genau drei Komponenten kodiert, nämlich Y, C_b und C_r, jeweils blockweise mit 8 x 8 Pixeln pro Block. Die Helligkeit Y wird mit voller Auflösung kodiert, die beiden Chroma-Komponenten jeweils mit halber Auflösung in horizontaler und vertikaler Richtung (4:2:0-Modell).

Es gibt nur zwei Bildformate:

CIF	(Common Intermediate Format):	352x 288
QCIF	(Quarter CIF):	176x144

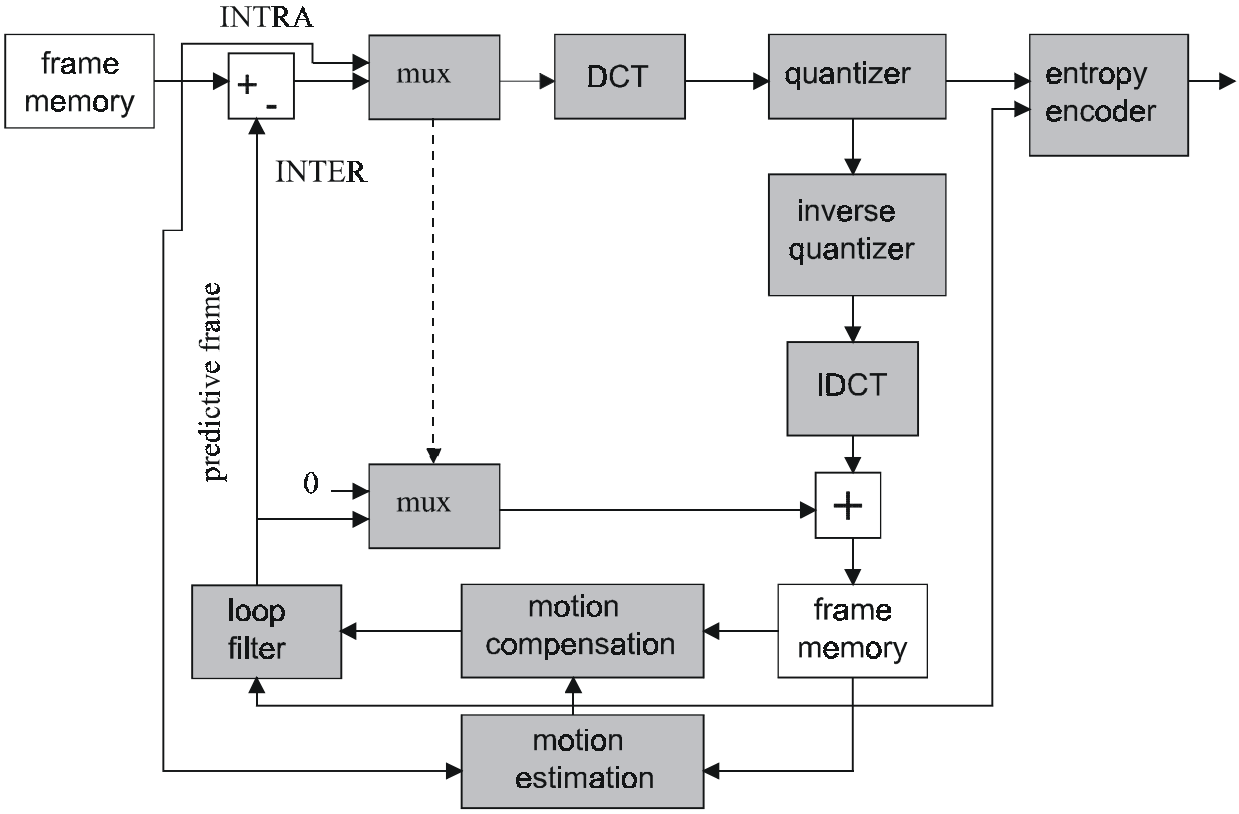
Die wichtigsten Parameter von H.261

	CIF (Breite x Höhe)	**QCIF (Breite x Höhe)**
Y	352 x 288	176 x 144
C_b	176 x 144	88 x 72
C_r	176 x 144	88 x 72

Die Hierarchie der Blockstrukturen

Strukturelement	**Beschreibung**
picture	Einzelbild (frame)
group of blocks	33 Macroblöcke
macro block	16 x 16 Y, 8 x 8 C_b, C_r
block	8 x 8 Pixel (ein Block für die DCT)

Die Funktionsweise eines H.261-Encoders



Status von H.261

Implementierungen in Hardware und Software verfügbar. Fast alle Bildtelefon-Hersteller haben inzwischen proprietäre Lösungen aufgegeben und auf H.261 und/oder den Nachfolgestandard H.263 umgestellt.

PC-Lösungen sind gegenüber teurer, Spezial-Hardware auf dem Vormarsch.

H.263

H.263 ist eine Weiterentwicklung von H.261, wobei vor allem die Erfahrungen aus MPEG einbezogen wurden. Einige Unterschiede zu H.261 sind:

- Es gibt fünf Bildgrößen statt zwei.
- Es gibt eine bidirektionale Interpolation, wobei immer genau ein B-Frame nach einem P-Frame platziert wird.
- Es gibt aushandelbare Optionen, die den Algorithmus für bestimmte Anwendungen optimieren, zum Beispiel kann in der Entropie-Codierung eine arithmetische Kodierung an Stelle der Lauflängen-/Huffman-Codierung verwendet werden.

2.4 Audio-Kompression

2.4.1 Pulse-Code-Modulation

Audiosignale sind analoge Wellenverläufe. Die charakteristischen Eigenschaften werden im Wesentlichen von der Frequenz (Tonhöhe) und von der Amplitude (Lautstärke) bestimmt.

Vor der weiteren Verarbeitung im Rechner müssen die analogen Wellenverläufe in digitale Signale transformiert werden. Dies geschieht durch **Abtastung**.

Abtastung

Für die Zeitdiskretisierung muss eine Abtastung der Analogverläufe erfolgen. Praktisch wichtig ist vor allem die periodische Abtastung.

Der zum Abtastzeitpunkt vorliegende Momentan-Wert des Analogsignals wird der Analog-Digital-Umsetzung unterworfen.

Abtast-Theorem von Shannon und Raabe (1939)

Zur fehlerfreien Rekonstruktion des Signalverlaufs des Analogsignals ist eine Mindestabtastfrequenz f_A erforderlich (bei periodischem Abtastzyklus). Diese hängt von der höchsten im analogen Signal vorkommenden Frequenz ab. Für rauschfreie Kanäle gilt das folgende

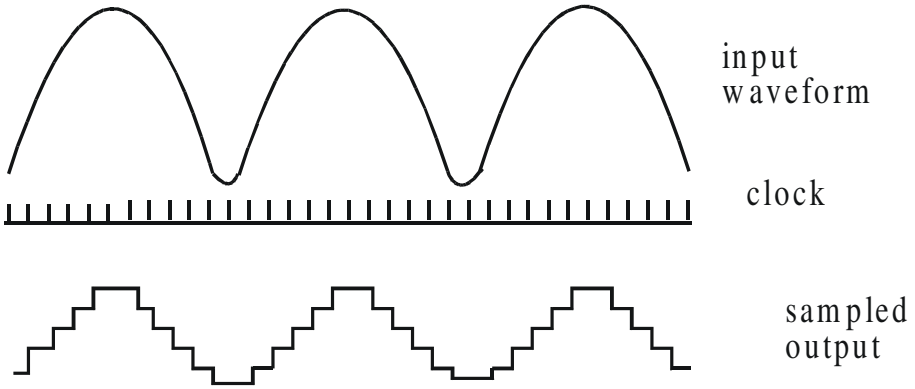
Abtast-Theorem

Die Abtastfrequenz f_A muss doppelt so hoch sein wie die höchste im abzutastenden Signal vorkommende Frequenz f_S:

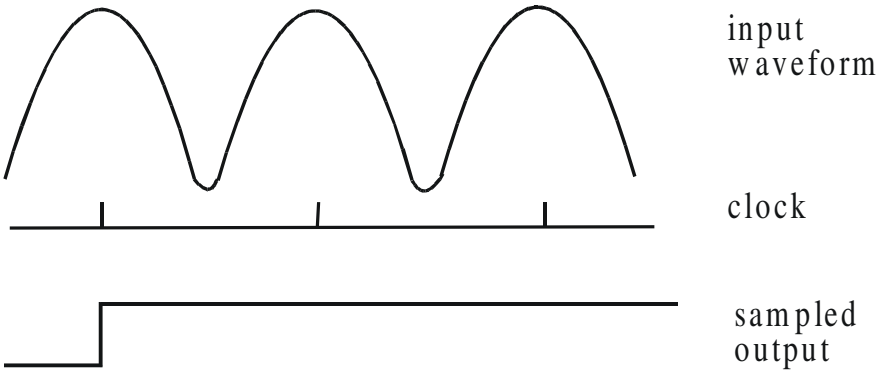
$$f_A = 2\, f_S$$

Abtastung und Quantisierung sind voneinander unabhängig zu betrachten. Eine exakte Rekonstruktion des Zeitverlaufs (bzw. des Frequenzspektrums) sagt nichts über den Fehlergrad bei der Signalwertdiskretisierung (Quantisierung) aus.

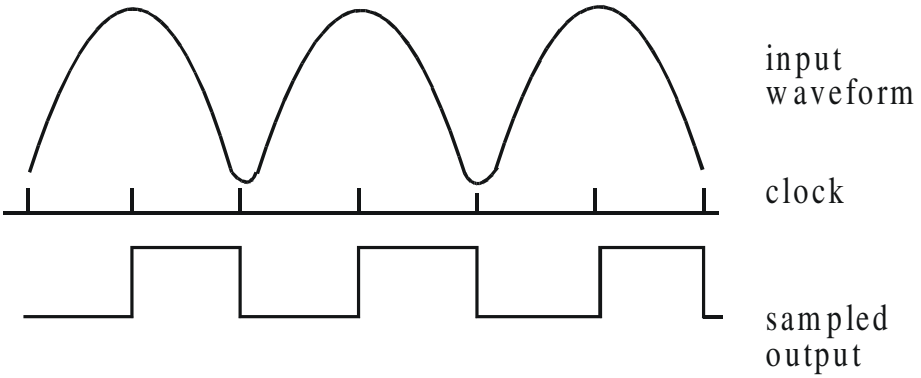
Beispiel: Abtasten bei verschiedenen Taktraten



(a) Sampling rate is much higher than signal frequency



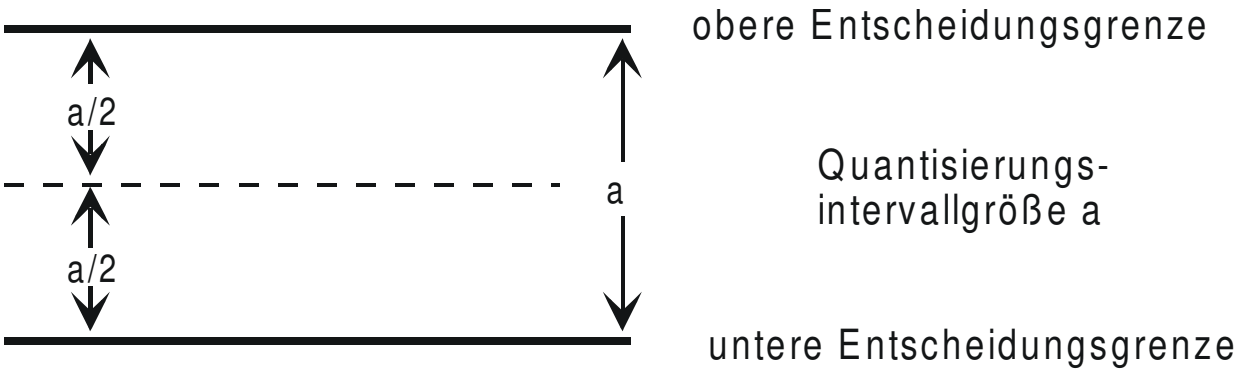
(b) Sampling rate is lower than signal frequency



(c) Sampling rate is at the Nyquist limit

Quantisierung

Der gesamte Wertebereich des Analogsignals wird in eine endliche Anzahl von Intervallen (Quantisierungs-intervalle) eingeteilt, denen jeweils ein fester diskreter Wert zugeordnet wird. Da alle in ein Quantisierungs-intervall fallenden Analogwerte demselben diskreten Wert zugeordnet werden, entsteht ein Quantisierungs-fehler.



Rückwandlung

Beim Empfänger wird ein Analogwert rückgewonnen (Digital-Analog-Umsetzung), der dem in der Mitte des Quantisierungsintervalls liegenden Analogwert ent-spricht.

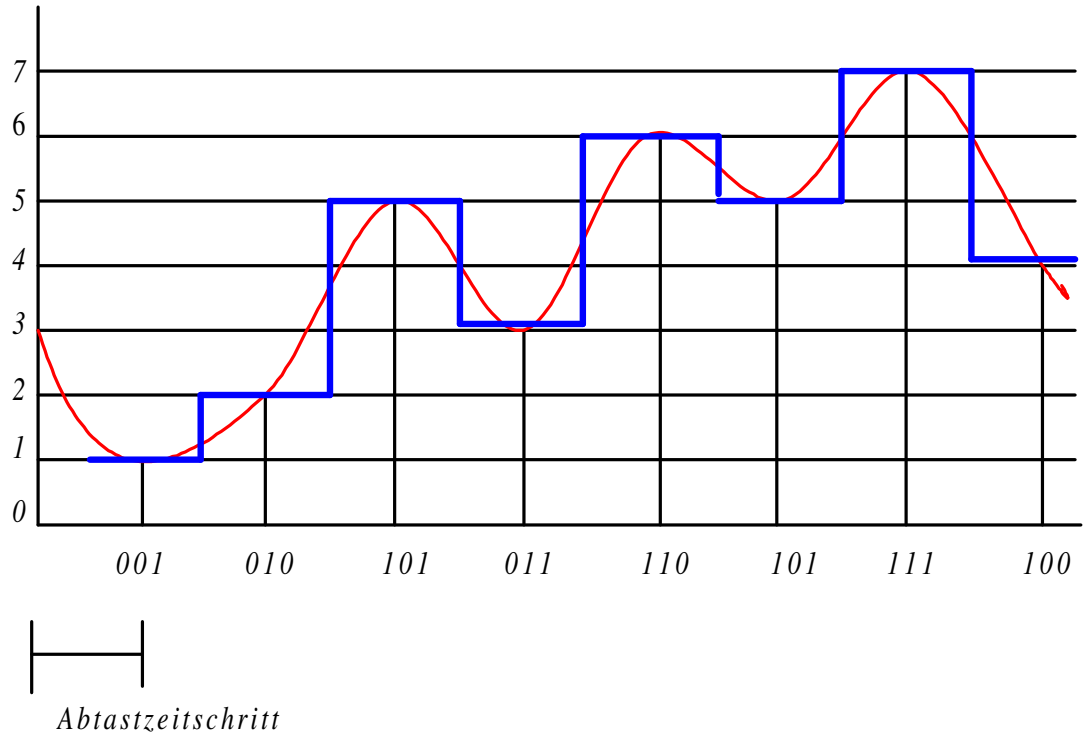
Maximaler Quantisierungsfehler: a/2

Codierung

Die quantisierten Werte werden durch die Zuordnung eines - frei wählbaren - (Binär-)Codes gekennzeichnet. Anstelle des ursprünglichen Analogsignals wird der digitale Codewert übertragen.

Im einfachsten Fall wird ein reiner Binärcode als Codierung des diskreten Digitalwertes gewählt (Darstellung als Binärzahl).

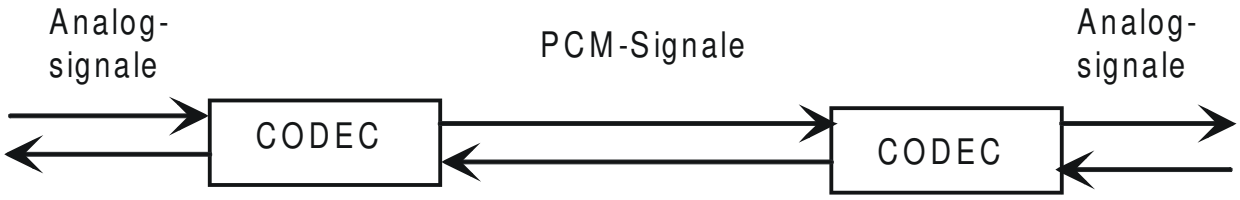
Zusammenfassende Darstellung



Die Zusammenfassung der Schritte Abtastung, Quantisierung und Codierung bezeichnet man als **Pulse-Code-Modulation (PCM).**

CODECs

Die A/D-Umsetzung (Abtastung/Quantisierung) und Codierung sowie die Rückkonvertierung erfolgt im sogenannten CODEC (Codierer/Decodierer).



PCM-Fernsprechkanal

Abtastrate

Ausgangspunkt: Analoger CCITT-Fernsprechkanal

Frequenzlage 300-3400 Hz

Bandbreite 3100 Hz

Abtastfrequenz: $\mathbf{f_A = 8\ kHz}$

Abtastperiode: $\mathbf{T_A = 1/\ f_A = 1/8000\ Hz = 125}$ **Mikrosekunden**

Die vom CCITT gewählte Abtastfrequenz ist höher als nach Shannon-Abtasttheorem erforderlich: 3400 Hz obere Bandgrenze ergibt 6800 Hz Abtastfrequenz. Für die höhere Abtastfrequenz gibt es technische Gründe (Filtereinfluss, Kanaltrennung usw.).

Amplitudenquantisierung

Die Zahl der benötigten Quantisierungsintervalle wird bei der akustischen Sprachkommunikation (Fernsprechen) durch den Grad der Silbenverständlichkeit beim Empfänger bestimmt.

Mit „Sicherheitszuschlag“ wurden vom CCITT 256 Quantisierungsintervalle genormt.

Bei binärer Kodierung ergibt dies 8 Bits pro Abtastung.

Übertragungsgeschwindigkeit für PCM-Kanal

Die Übertragungsgeschwindigkeit (Bitrate) für einen digitalisierten PCM-Fensprechkanal ist demnach

Bitrate	**=**	**Abtastfrequenz**	**x**	**Codewortlänge**
kbit/s	**=**	**8000/s**	**x**	**8 bit**

= 64 kbit/s

Ungleichförmige Quantisierung

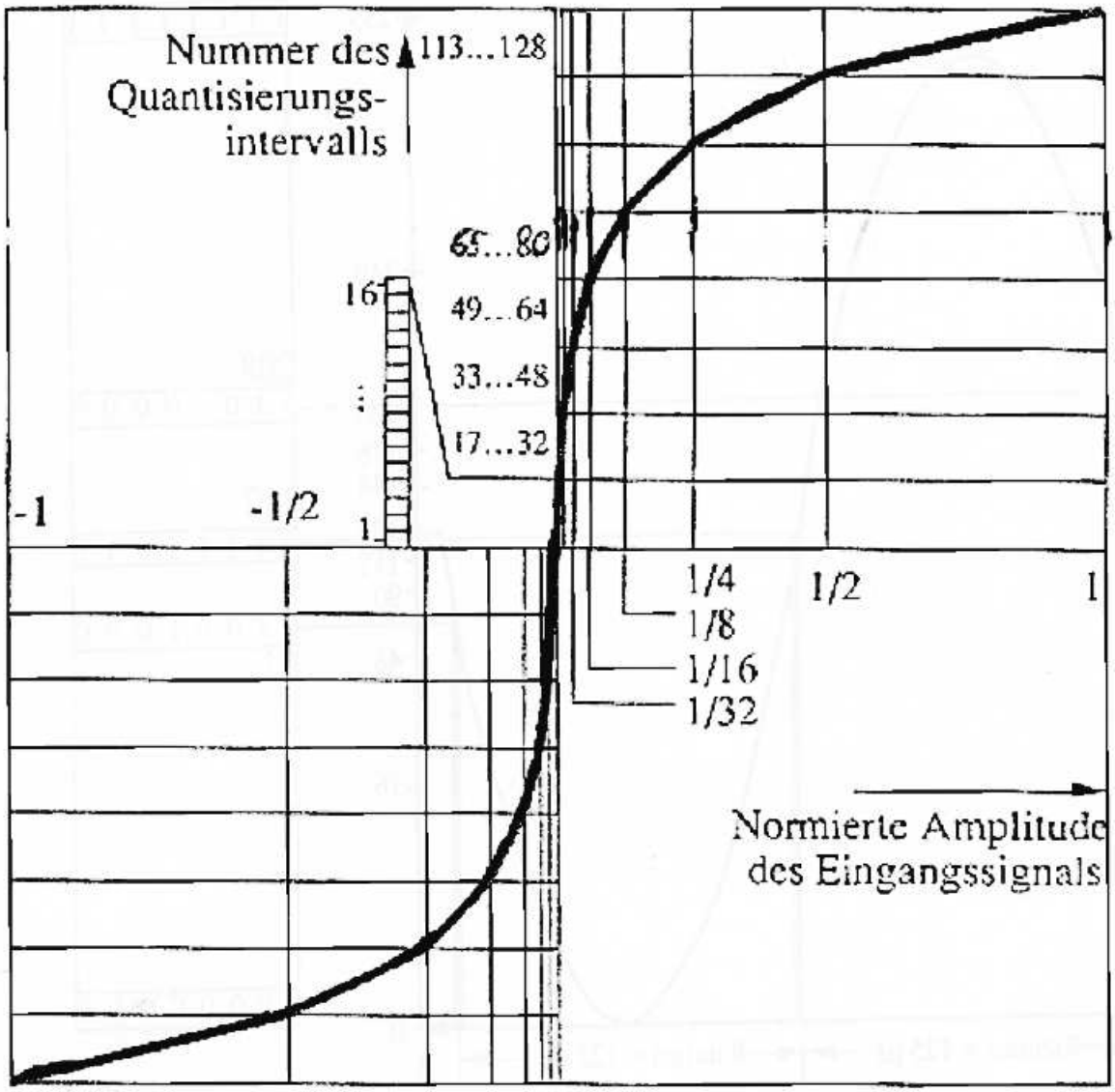
Bei gleichförmiger Quantisierung sind alle Intervalle gleich groß und vom Momentanwert des Signals unabhängig. Quantisierungsfehler machen sich bei gleichförmiger Quantisierung bei kleinen Signalwerten sehr stark bemerkbar (Quantisierungsrauschen).

Bei ungleichförmiger Quantisierung sind die Quantisierungsintervalle bei großer Signalamplitude größer und bei kleiner Amplitude kleiner.

Die ungleichförmige Intervallgröße wird durch einen dem Quantisierer vorgeschalteten Signal-Kompressor erzielt. Auf der Empfangsseite wird in inverser Funktion ein Expander eingesetzt. Er dient zur Wiederherstellung der ursprünglichen Größenverteilung der Signale (Dynamik der Signale).

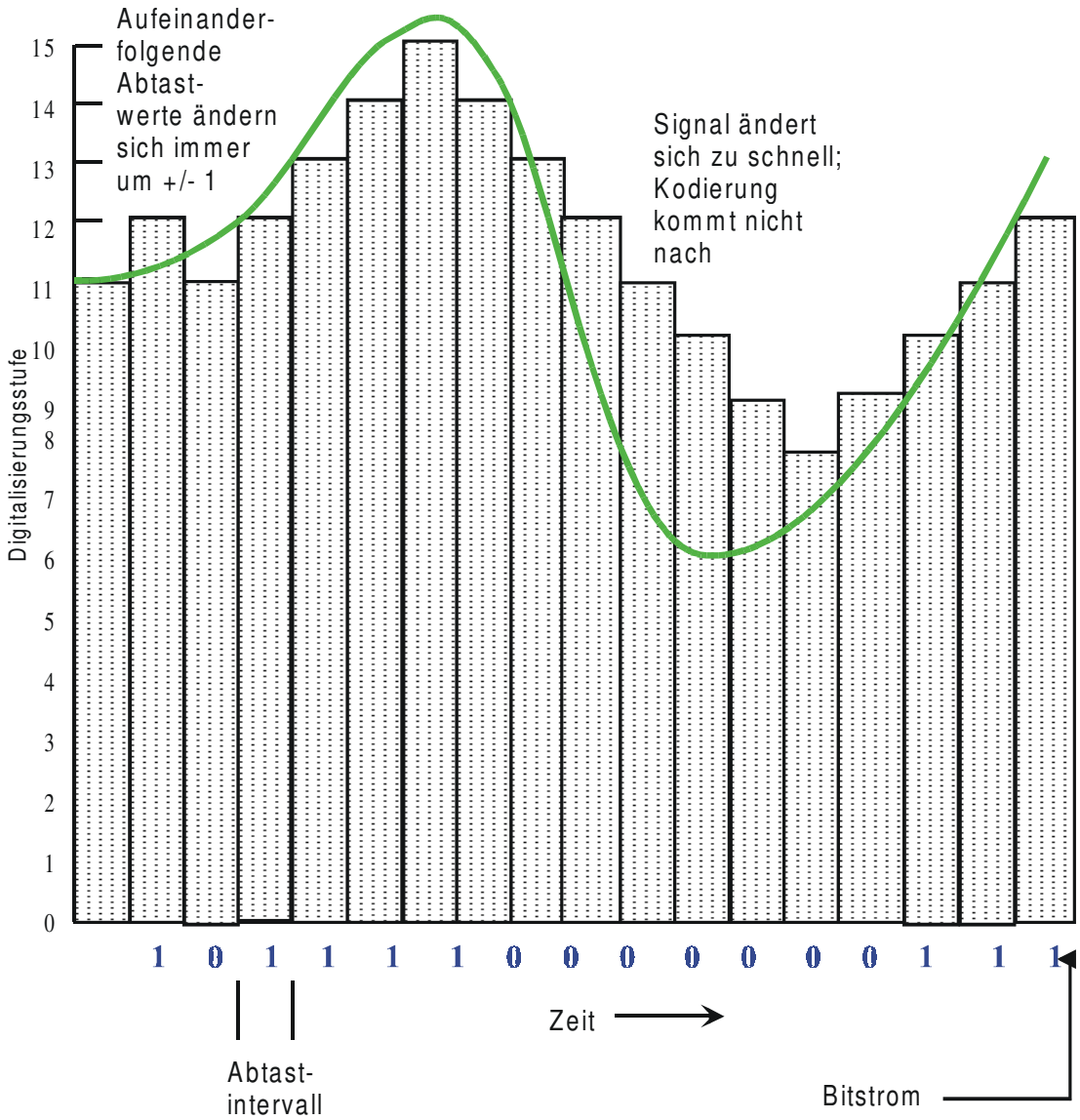
Als Kompressionskennlinien werden logarithmische Kennlinien verwendet, die schaltungstechnisch durch lineare Teilstücke approximiert werden.

13-Segment-Kompressorkennlinie



Delta-Modulation

Statt der Absolutwerte der Amplitude werden die **Wert-differenzen** in Stufen von +1 oder -1 codiert.



Kodierung:	1	=	steigendes Signal
	0	=	fallendes Signal

Differenzielle PCM (DPCM)

Es wird die echte Differenz zweier PCM-Werte codiert. Dies ergibt eine Bitrate, die zwischen derjenigen der Absolutwert-PCM und derjenigen der Delta-Codierung liegt.

Adaptive DPCM (ADPCM)

Die Quantisierungstabelle ändert sich mit den tatsächlichen Signalschwankungen. Der Empfänger kann diese Tabellenänderung dynamisch nachvollziehen. Bei kleinen Signalschwankungen wird mit weniger Bit/s übertragen als bei großen Signalschwankungen. Dadurch kann beispielsweise eine Kompression von 1,4 Mbit/s auf 0,2 Mbit/s bei vergleichbarer Qualität möglich werden.

Bekannte ADPCM-Verfahren sind µ-law und A-law.

Gebräuchliche Sampling-Parameter

Abtastrate

8 kHz	Telefon-Standard, µ-law encoding, SUN Audio
32 kHz	Digital Radio Broadcast
44,1 kHz	Audio-CD
48 kHz	Digital Audio Tape (DAT)

Quantisierung

8 bit	256 Amplituden-Stufen (Sprache)
16 bit	65536 Amplituden-Stufen (HiFi)

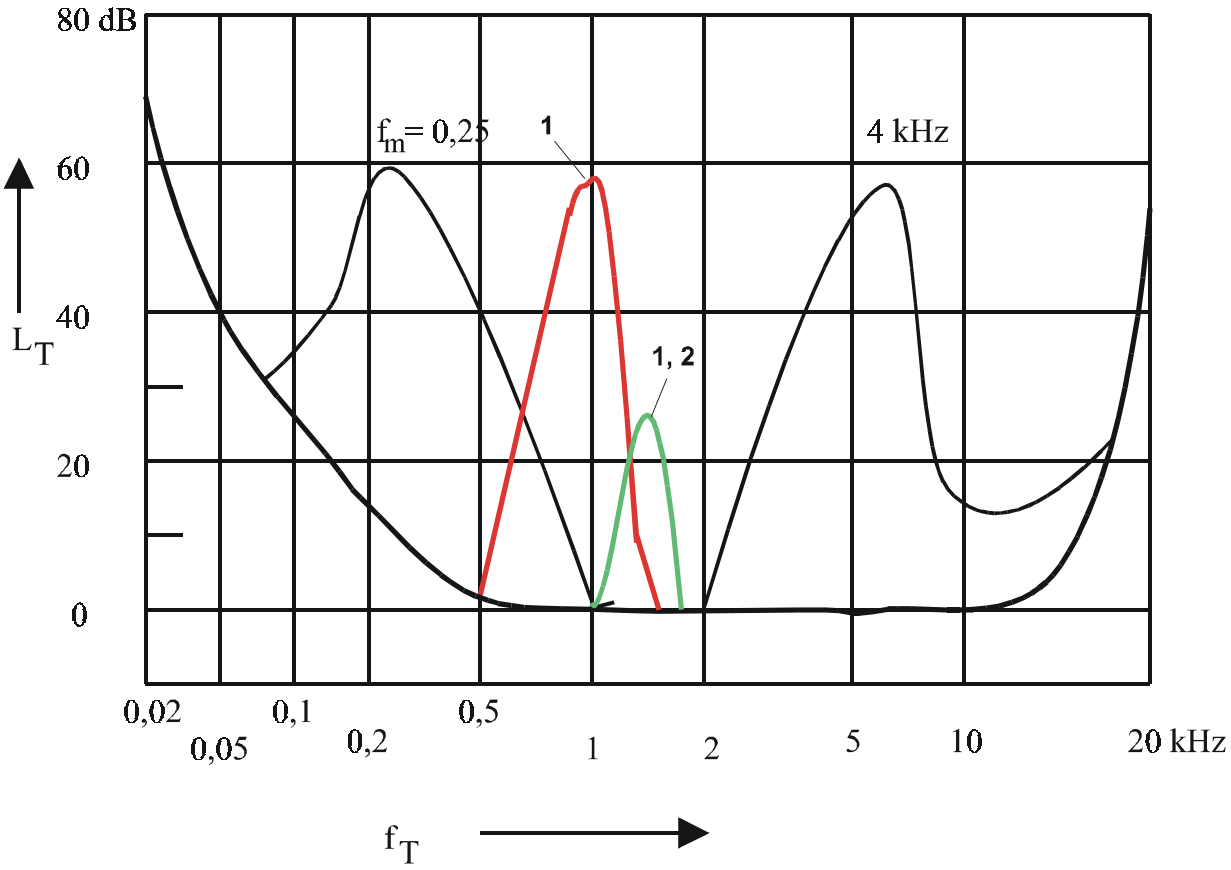
2.4.2 Audiocodierung mit psycho-akustischen Modellen

"Irrelevanzreduktion"

Ausnutzung des Auflösungsvermögens des menschlichen Ohres

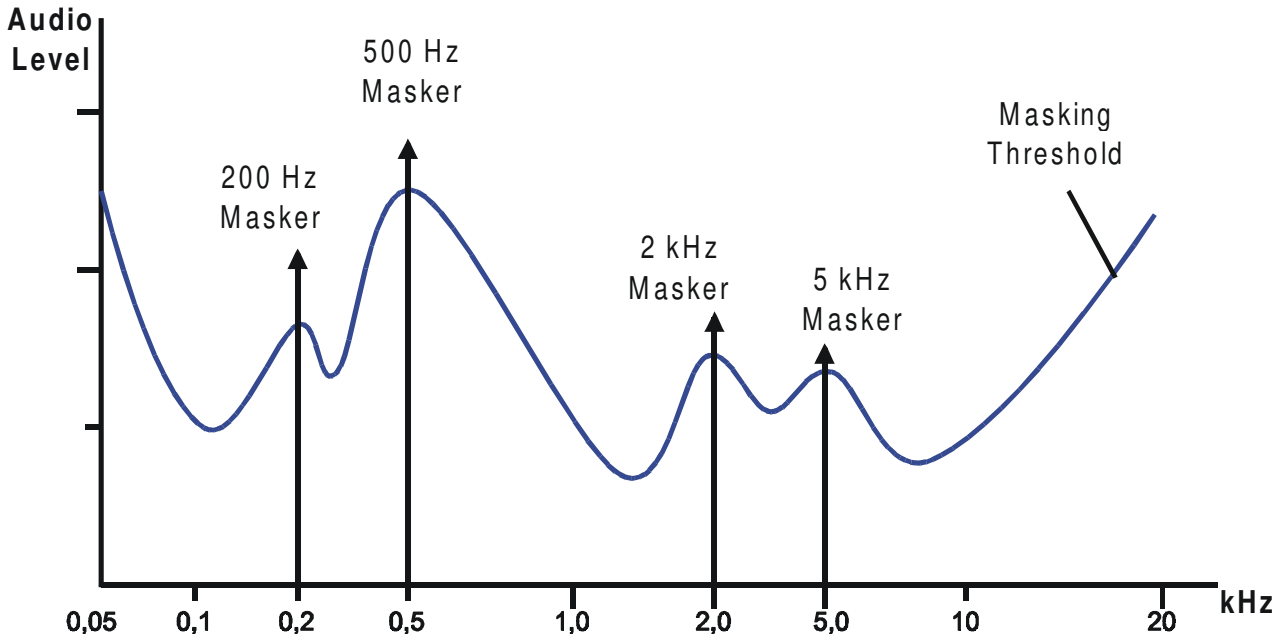
Verdeckungseffekt

- Mithörschwelle, frequenz- und pegelabhängig



Daten, die das Ohr sowieso nicht hören würde, werden schon an der Quelle herausgefiltert.

Psycho-akustische Modelle



Beispiel: MPEG Audio

Merkmale

Kompression auf 32, 64, 96, 128 oder 192 kbit/s

Audiokanäle

- Mono oder
- zwei unabhängige Kanäle oder
- "Joint Stereo"

Verfahren

- Abtastrate 32 kHz, 44,1 kHz oder 48 kHz
- 16 Bits pro Abtastwert
- Verzögerung durch Codieren und Decodieren höchstens 80 ms bei 128 kbit/s

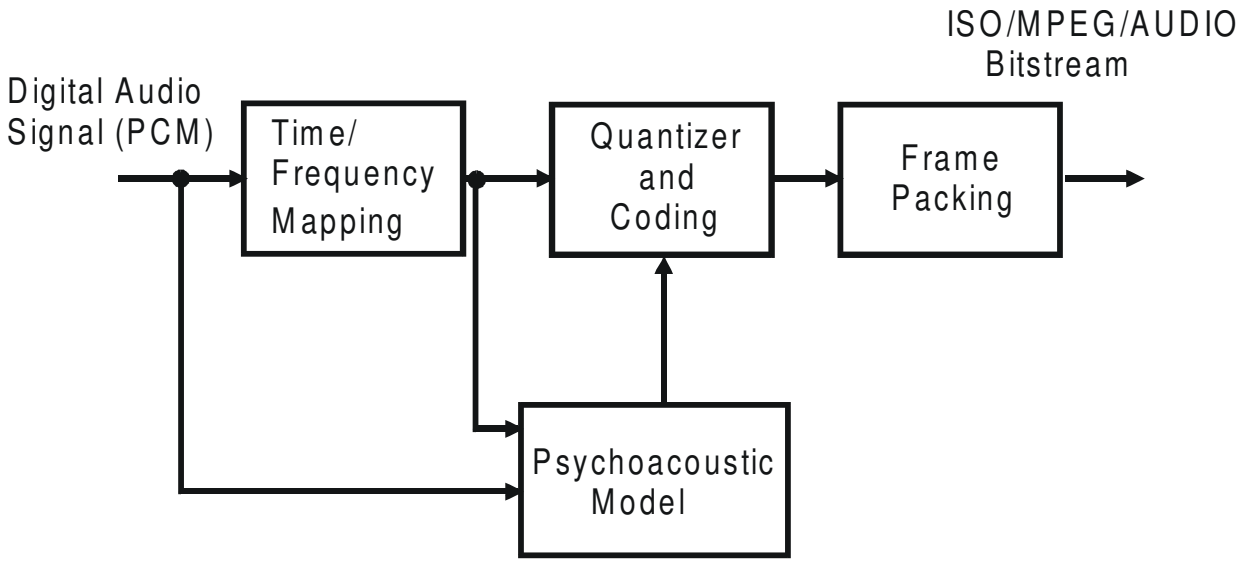
- **Ein psychoakustisches Modell steuert die Quantisierung**

Zwei Verfahren in MPEG-Audio

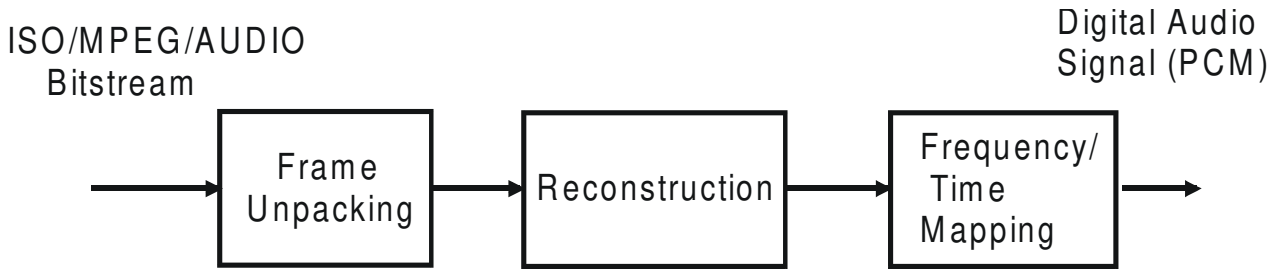
MUSICAM	ASPEC
Masking Pattern Universal Subband Integrated Coding And Multiplexing	Advanced Spectral Entropy Coding
Institut für Rundfunktechnik München	FhG Erlangen
Teilbandkodierung	dynamische Frequenzbänder (überlappend) Entropie-Kodierung (Huffman)
einfacher Aufbau	sehr gute Ergebnisse bei niedrigen Bitraten

MPEG Audio-Encoder und -Decoder

Encoder



Decoder



Drei Schichten (layers) in MPEG Audio

I. Teilbandkodierung mit 32 Bändern nach MUSICAM
   - hohe Datenrate
   - mono, stereo, 48 kHz, 44.1 kHz, 32 kHz

II. Teilbandkodierung nach MUSICAM, aufwendigeres psycho-akustisches Modell, bessere Bestimmung der Skalierungsfaktoren
   - mittlere Datenrate
   - bessere Klangqualität bei niedrigen Bitraten, Multikanal-Klang

III. Transformationskodierung mit max. 512 dynamischen Fenstern und Entropie-Kodierung nach ASPEC (MP3)
   - niedrigste Datenrate
   - Stereo Audio in CD-Qualität: 128 kbit/s (Verhältnis 12:1)
   - Mono Audio in Telefonqualität: 8 kbit/s

MP3

<wird noch ergänzt>

2.4.3 Sprachkodierung

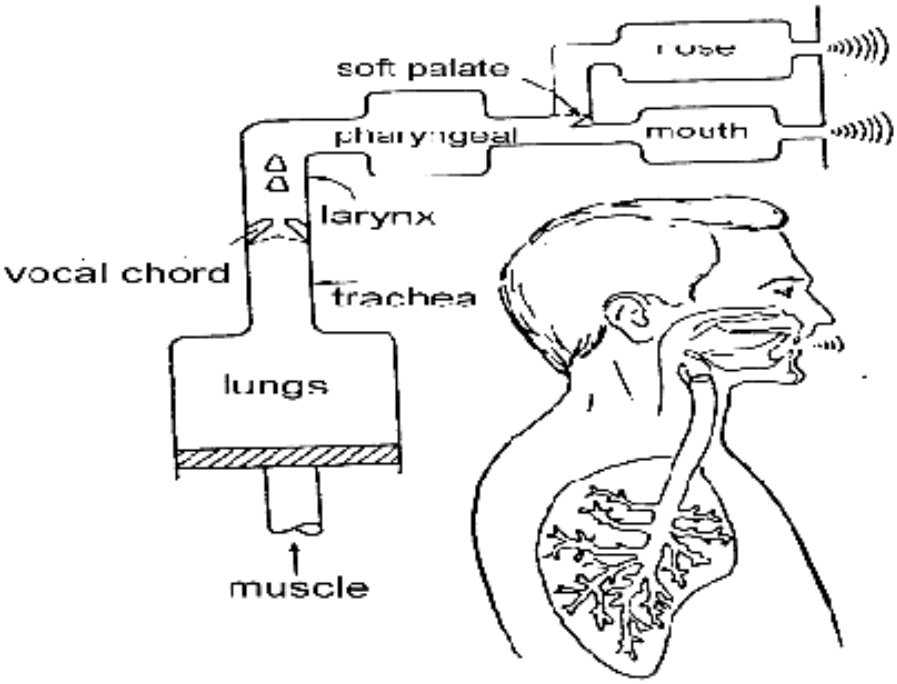
Idee

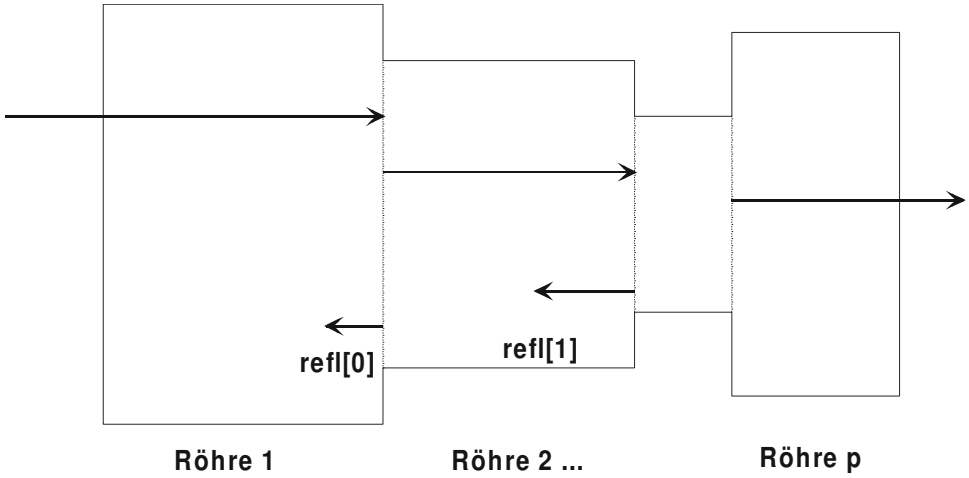
Spezielle, für die menschliche Sprache optimierte Codecs (normaler Stimmenbereich ca. 300 Hz bis 3,4 kHz) können eine hohe Sprachqualität bei sehr niedrigen Bitraten erreichen.

Solche Sprachcodierer basieren in der Regel auf **Linear Predictive Coding (LPC).**

Linear Predictive Coding (1)

LPC modelliert den menschlichen Sprachkanal als ein System von Röhren (Zylindern) mit unterschiedlichen Durchmessern.





Linear Predictive Coding (2)

Akustische Wellen werden von den Stimmbändern stimmhaft oder stimmlos generiert, durchlaufen ein System von Röhren, werden an Übergängen von Röhren mit unterschiedlichem Durchmesser teilweise reflektiert und interferieren mit nachfolgenden Wellen.

Die Reflektionsrate wird durch die Reflektions-koeffizienten refl[0], ..., refl[p-1] repräsentiert. Mit einer relativ kleinen Zahl von Parametern lässt sich so die (sprecherabhängige) Erzeugung des Sprachklangs beschreiben.

LPC-Encoder

- Das Audio-Signal wird in kleine Rahmen fester Länge zerlegt (20 - 30 ms). Für jeden Rahmen s[i] werden p Gewichte lpc[0], .. , lpc[p-1] berechnet, so dass s[i] möglichst gut durch

  lpc[0] * s[i-1] + lpc[1] * s[i-2] + ... + lpc[p-1] * s[i-p]

  angenähert wird. Übliche Werte für p sind 8 oder 14.
- Ein synthetisch generiertes Quellsignal dient als Eingabe des Modells und ergibt synthetische Sprache für die Dauer des Rahmens. Die Generierung kann zwischen "stimmhaft" (Sinus-Schwingung) und "stimmlos" (Rauschen) umgeschaltet werden.
- Die Unterschiede zwischen dem generierten Sprachsignal und dem tatsächlichen Sprachsignal werden ermittelt und führen zur Neuberechnung der Prädiktionskoeffizienten lpc[i]. Kodiert und übertragen werden für jeden Rahmen die Art der Anregung (stimmhaft oder stimmlos) sowie die aktuellen Paremeterwerte lpc[i].

LPC-Varianten

- **CELP** (Code Excited Linear Prediction): Es werden nicht nur "stimmhaft" und "stimmlos" unterschieden, sondern wesentlich mehr Arten der Anregung. Diese werden vom Systementwerfer in Form eines Codebooks hinterlegt. Für jeden Rahmen werden jetzt der Codebook-Index und die lpc-Parameter codiert und übertragen.
- **ACELP**: wie CELP, aber mit adaptivem Codebook
- **GSM 06.10**: Regular Pulse Excitation – Long Term Prediction (RPE-LTP)

Ein gutes Kurz-Tutorial über LPC mit Audio-Beispielen ist unter

http://asylum.sf.ca.us/pub/u/howitt/lpc.tutorial.html

zu finden.

LPC-Beispiele

G.723.1

Adaptiver CELP-Codierer (Code Excited Linear Predictor).

Bitrate für G.723.1: 5,3 kbit/s - 6.3 kbit/s

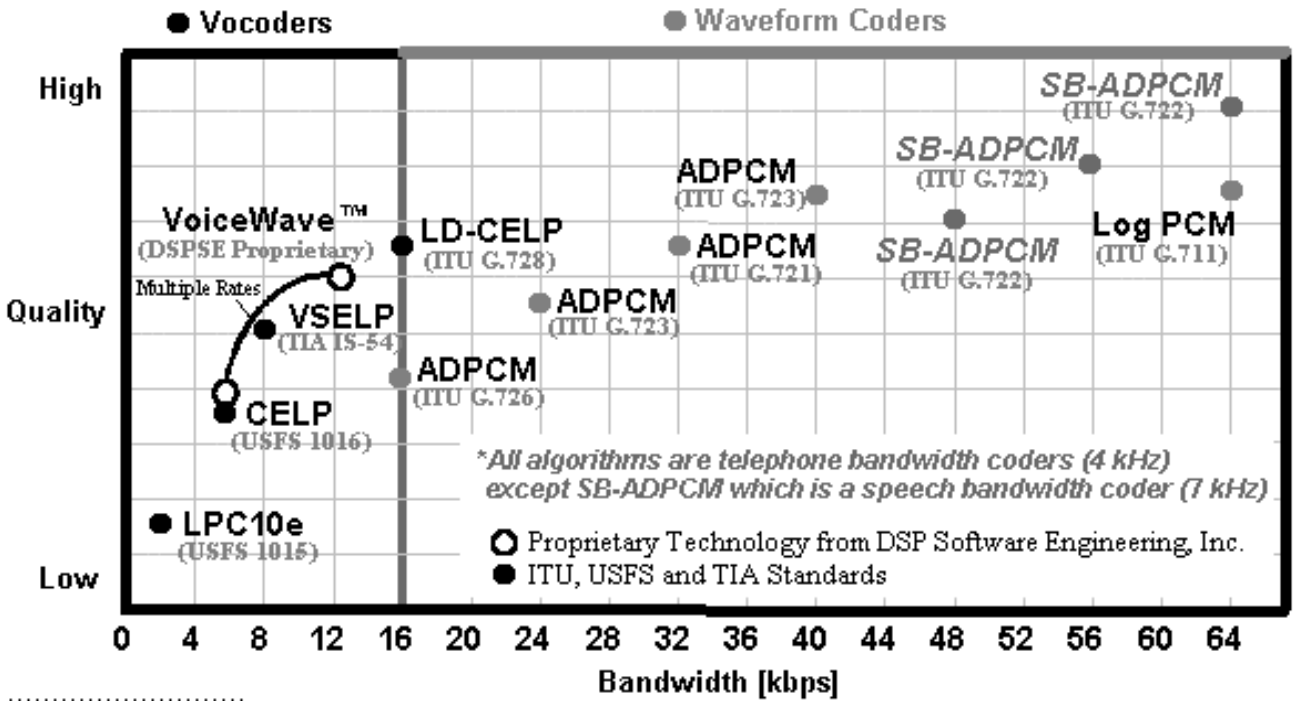
GSM 06.10

Regelmäßige Pulsstimulierung - langfristige Vorhersage (RPE-LTP)

- LPC-Kodierung
- Synthetisch generiertes Signal basiert auf vorhergehenden Signalwerten und benutzt stimmlose Phoneme im Signal zur Pulsstimulierung

Bitrate für GSM 06.10: 13.2 kbit/s

Spezielle Sprachkodierung vs. PCM-Kodierung



ITU-T-Standards für die Sprachkodierung

Eine Auswahl aus den G.7xx-Standards

- **G.711**: 64kbit/s (GSTN-Telefonie, H.323 & H.320 Videoconferencing)
- **G.728** LD-CELP: 16 kbit/s (GSTN-Telefonie, H.320 Videoconferencing)
- **G.729** ACELP: 8 kbit/s (GSTN-Telefonie, H.324 Video-Telefonie)
- **G.723.1** MPE/ACELP 5.3 kbit/s bis 6.3 kbit/s (GSTN Video-Telefonie, H.323 Telefonie)