DCT-basierte Dateiformate

Mathematisches Seminar

Christian Fischer

1. Grundlagen (1/2)

- Kompression = Irrelevante, Redundante Bestandteile entfernen, Symbole zusammenfassen bzw. neue Codewörter definieren und schließlich Codierung.
  - Kompressionsrate: $C_R = \frac{Datenmenge(Originalsignal)}{Datenmenge(codiertes\,Signal)}$
  - Angabe der Kompressionsleistung durch Bitrate: $R = \frac{(N_B \cdot 8)}{N_A}$
    - bpp = Bit per Pixel
    - N_B … Datenmenge des codierten Signals
    - N_A … Anzahl der Symbole
- Huffman-Codierung mit Hilfe des Codebaummodells:
  1. Symbole = Blätter des Baumes; trage Wahrscheinlichkeiten für jedes Symbol ein
  2. Fasse beide geringsten Wahrscheinlichkeiten zu einem Knoten zusammen; weise Ihre Summe dem Knoten zu
  3. Beschrifte die neuen Zweige mit 0 bzw. 1
  4. Wenn bei der Wurzel des Baumes die Wahrscheinlichkeit p = 1 eingetreten ist, beende die Konstruktion.
  5. gehe zu 2.
- Lauflängencodierung
  - A0DD0000000A000000B0000C würde zu A1DD7A6B4C werden
- Diskrete Transformation
  - allgemeine eindimensionale diskrete Signaltransformation:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot a[n,k], \quad k = 0,1,..., N-1$$

    x[n] … Zeitdiskretes Originalsignal
    a[n,k] … Transformationskern
    X[k] … transformiertes Signal
  - Rücktransformation:

$$x[n] = \sum_{n=0}^{N-1} X[k] \cdot b[k,n], \quad n = 0,1,..., N-1$$

    b[n,k] … Rücktransformationskern
  - Beziehungen:

$$X = A \cdot x, \quad x = B \cdot X \quad , \quad B = A^{-1}$$

- Diskrete Kosinus Transformation (DCT)
  - reellwertiger Transformationskern

$$a[n,k] = C_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left[(2 \cdot n + 1) \cdot \frac{k \cdot \pi}{2 \cdot N}\right], \; mit \; C_0 = \begin{cases} \frac{1}{\sqrt{2}}, & für \; k = 0 \\ 1 & , \; für \; k \neq 0 \end{cases}$$

  - Transformation:

$$X[k] = C_0 \cdot \sqrt{\frac{2}{N}} \cdot \sum_{n=0}^{N-1} x[n] \cdot \cos\left[(2 \cdot n + 1) \cdot \frac{k \cdot \pi}{2 \cdot N}\right]$$

  - Rücktransformation: IDCT – inverse DCT

$$x[n] = C_0 \cdot \sqrt{\frac{2}{N}} \cdot \sum_{n=0}^{N-1} X[k] \cdot \cos\left[(2 \cdot n + 1) \cdot \frac{k \cdot \pi}{2 \cdot N}\right]$$

1. Grundlagen (2/2)

- Farbsysteme
  - RGB = Mischung aus Rot,Grün,Blau
  - YUV
    - Y – Helligkeitskomponente (Luminanz)
    - U,V – Farbdifferenzwerte (Chrominanz)
    - Transformation vom RGB-Farbraum:

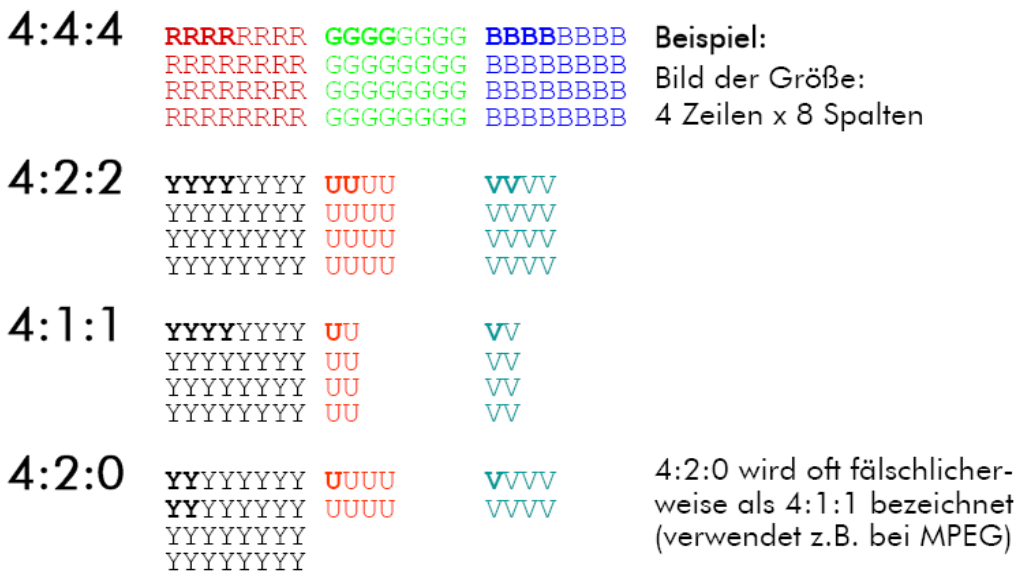
$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

  - YCbCr (YUV nach IEC 601)

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- Unterabtastformate

Format	Bits pro Bildpunkt	Beschreibung
4:4:4	8+8+8=24	keine Unterabtastung
4:2:2	8+4+4=16	horizontale Unterabtastung der Chrominanzen
4:1:1	8+2+2=12	horizontale Unterabtastung der Chrominanzen
4:2:0	8+2+2=12	horizontale und vertikale Unterabtastung



2. JPEG (Joint Pictures Experts Group - 1986) (1/2)

- Ziel: Entwicklung eines Progressives Kompressionsverfahren für Einsatz auf ISDN Kanälen (64Kbit/s)
- grundlegendes Verfahren wurde 1989 entwickelt: (baseline System)
  - verlustbehaftet
  - DCT-basiert
  - sequentiell mit Huffman-Code
  - 8 Bit Genauigkeit pro Bildpunkt
- Optional:
  - arithmetische Codierung (mit patenten geschützt)
  - progressive Codierung
  - hierarchische Codierung
- Zusätzlich gibt es eine Funktion zur verlustlosen Codierung.
- Verwendeter Farbraum: meistens YCbCr
  - begrenztes Farbsehen des Menschen wird ausgenutzt
    → Unterabtastung der Chrominanzen 4:2:0 o.ä.
- Wertebereich der Eingangsdaten wird zentriert:
  - 8Bit Daten: 0…255 → -128…127
- Bild wird in Blöcken à 8x8 Pixel eingeteilt
- 2D-FDCT (forward discrete cosine transform)

$$X[k,l]=\frac{C_k\cdot C_l}{4}\cdot\sum_{n=0}^{7}\sum_{m=0}^{7}x[n,m]\cdot\cos\left[\frac{(2n+1)\cdot k\cdot\pi}{16}\right]\cdot\cos\left[\frac{(2m+1)\cdot l\cdot\pi}{16}\right] \qquad C_k=\begin{cases}\frac{1}{\sqrt{2}}, & \text{für } k=0\\ 1, & \text{für } k\neq 0\end{cases}$$

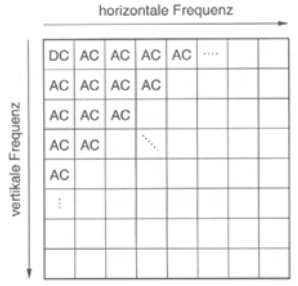
- 2D-IDCT (inverse discrete cosine transform)

$$x[n,m]=\sum_{k=0}^{7}\sum_{l=0}^{7}\frac{C_k\cdot C_l}{4}\cdot X[k,l]\cdot\cos\left[\frac{(2n+1)\cdot k\cdot\pi}{16}\right]\cdot\cos\left[\frac{(2m+1)\cdot l\cdot\pi}{16}\right] \qquad C_l=\begin{cases}\frac{1}{\sqrt{2}}, & \text{für } l=0\\ 1, & \text{für } l\neq 0\end{cases}$$

- Quantisierung:

$$q[k,l]=\left\lfloor\frac{X[k,l]}{Q_{k,l}}+0.5\cdot\operatorname{sgn}(X[k,l])\right\rfloor$$

  q[k,l] … Quantisierungssymbol



- Rekonstruktion:

$$[X[k,l]]_Q=q[k,l]\cdot Q_{k,l}$$

- 8x8 Blöcke werden in 1 DC Anteil und 63 AC-Anteile aufgeteilt
  - Durch Quantisierung werden viele AC-Koeffizienten Null
  - DCT-Koeffizienten werden gleichmäßig quantisiert
  - Jeder Koeffizient X[k,l] wird durch seinen Quantisierungswert $Q_{k,l}$ dividiert.
    - → Es können Verluste auftreten !
  - Werte werden in Quantisierungstabelle abgelegt

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	10	10	77
24	35	55	65	81	10	11	92
49	64	78	87	10	12	12	10
72	72	95	98	11	10	10	99

Luminanz

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Chrominanzen

2. JPEG (Joint Pictures Experts Group - 1986) (2/2)

Codierung der DC-Koeffizenten

- DC-Koeff. werden unabhängig von AC-Koeff. mit prädiktiven Verfahren codiert
- → Annahme: benachbarte Blöcke haben ähnlichen Gleichanteil (Mittelwert)
- → DC-Wert kann aus Vorgänger vorausgesagt werden
- Sei i Blocknummer → Prädiktionsfehler DIFF:

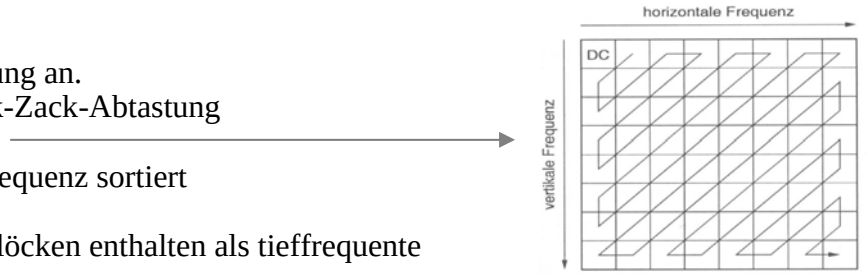
$$DIFF = DC_i - PRED \quad mit \quad PRED = \begin{cases} DC_{i-1} & i > 0 \\ 0 & i = 0 \end{cases}$$

- theoretischer Wertebereich von DIFF ist sehr groß
- → Einteilung in 12 Kategorien; jede Kategorie bekommt Codewort zugewiesen
- Differenzwerte innerhalb einer Kat. ungefähr gleichverteilt
- vorgeschlagene Huffman-Codes:

Kategorie	*DIFF*-Wert	Code (Lum)	Code (Chrom)
0	0	00	00
1	-1, 1	010	01
2	-3, -2, 2, 3	011	10
3	-7,. . . ,-4, 4,. . . , 7	100	110
4	-15,. . . ,-8, 8,. . . , 15	101	1110
5	-31,. . . ,-16, 16,. . . , 31	110	11110
6	-63,. . . ,-32, 32,. . . , 63	1110	111110
7	-127,. . . ,-64, 64,. . . , 127	11110	1111110
8	-255,. . . ,-128, 128. . . , 255	111110	11111110
9	-511,. . . ,-256, 256, . . . , 511	1111110	111111110
10	-1023,. . . ,-512, 512,. . . , 1023	11111110	1111111110
11	-2047,. . . ,-1024, 1024,. . . , 2047	111111110	11111111110

Codierung der AC-Koeffizenten

- Quantisierung → viele AC-Koeff. = 0
- → Es bietet sich eine Lauflängencodierung an.
- Die 8 x 8 Koeff.-Matrix wird durch Zick-Zack-Abtastung in eindimensionalen Vektor überführt



- Koeff. werden entsprechend ihrer Ortsfrequenz sortiert
- Annahme:
- weniger hochfrequente Anteile in den Blöcken enthalten als tieffrequente
- → ‚Nullen' am Ende des Vektors

- AC-Koeff. != 0 werden wie DC-Koeff. in Kategorien unterteilt:

Kategorie	AC-Wert
1	-1, 1
2	-3, -2, 2, 3
3	-7,. . . ,-4, 4,. . . , 7
4	-15,. . . ,-8, 8,. . . , 15
5	-31,. . . ,-16, 16,. . . , 31
6	-63,. . . ,-32, 32,. . . , 63
7	-127,. . . ,-64, 64,. . . , 127
8	-255,. . . ,-128, 128. . . , 255
9	-511,. . . ,-256, 256, . . . , 511
10	-1023,. . . ,-512, 512,. . . , 1023

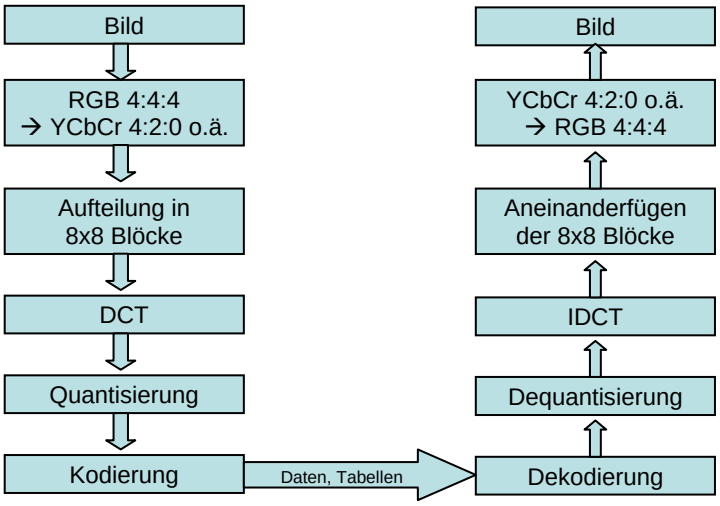
- Es wird für jeden Koeff. der Abstand zum Vorgänger ungleich 0 ermittelt.

- Lauflänge: 0…15
  → Neue Datensymbole aus Kombination von Lauflänge und Kategorie des AC-Koeff., da Anzahl benachbarter Nullen und Koeff. ähnlich sind.
  - ZRL := Lauflänge 15 Nullen mit Folgewert Null
  - EOB := End Of Block = Ende des Zick-Zack-Scans
  - → EOB folgt letzten Koeff. ungleich Null
    → nachfolgende Nullen müssen nicht codiert werden

Lauflänge/ Kategorie	Codewort
0/0 (EOB)	1010
0/1	00
0/2	01
0/3	100
0/4	1011
0/5	11010
0/6	1111000
0/7	11111000
0/8	1111110110
0/9	1111111110000010
0/10	1111111110000011
1/1	1100
1/2	11011
1/3	1111001
⋮	⋮

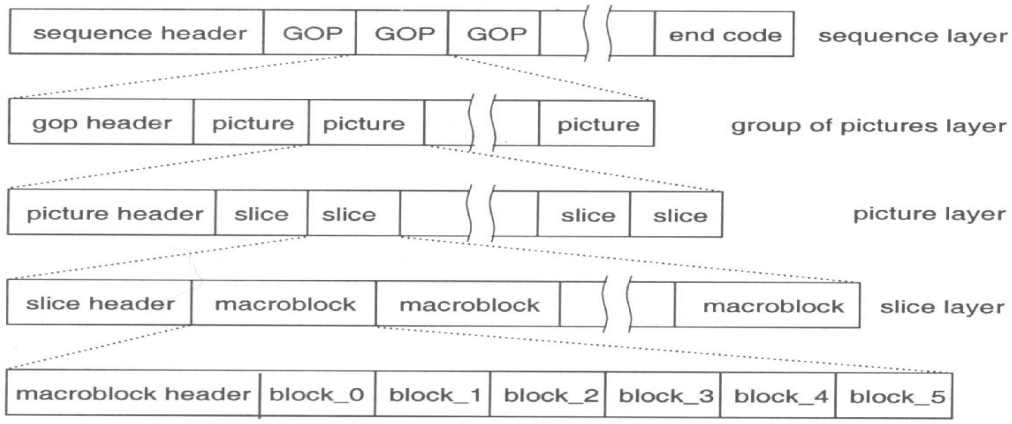
Lauflänge/ Kategorie	Codewort
⋮	⋮
2/1	11100
2/2	11111001
2/3	1111110111
.	.
3/1	111010
3/2	111110111
3/3	111111110101
.	.
11/1	1111111001
11/2	1111111111010000
.	.
15/0 (ZRL)	11111111001
.	.
15/10	1111111111111110

Allgemeines JPEG-Schema:



3. MPEG (Moving Pictures Experts Group - 1988) (1/2)

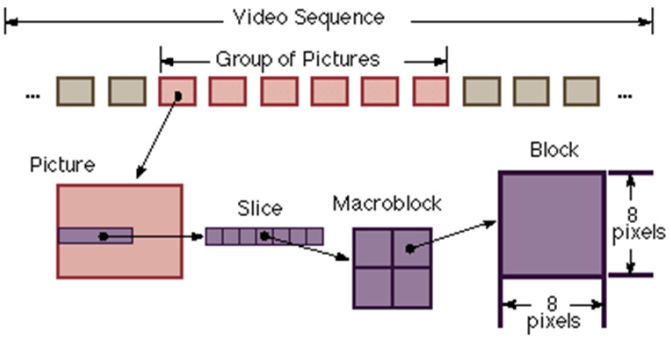
- Ziel: Wiedergabe von digitalen Video in Echtzeit auf CD-ROM-Applikationen etc.
- Datenrate ca. 1,5 Mbit/s
- → auch Kompression von Audiodaten
- 1993 MPEG 1 Standard
- 1996 MPEG 2 Standard
- CCIR-601 (heute ITU-R 601): Format 4:2:2 um Unterschiede zw. PAL und NTSC auszugleichen
- Struktur:



- Group of Pictures
  - erstes Bild ist immer ein intra-codiertes Bild (I-Bild)
    - wird wie ein Einzelbild ohne Infos von anderen Bildern verarbeitet
    - → sind Einsprungspunkte
  - P-Bilder (prädiktiv-codierte Bilder)
    - brauchen vorangegangene I- oder P- Bilder derselben Gruppe als Referenzbild
  - B-Bilder werden intercodiert
    - bidirektionale Prädiktion: vorangegangene & nachfolgende I- & P- Bilder werden einbezogen
- Durch Prädiktion von Bildinhalten kann Bitrate stark reduziert werden!
- Kompressionsrate für B-Bilder i.A. am höchsten
  - aber: Aufwand zur Bewegungsschätzung durch Vor- und Rückprädiktion ist ca. doppelt so hoch
- Anzahl der Bilder pro Gruppe kann frei gewählt werden
  - Es sollten allerdings nicht so viele Bilder sein, da sonst Bewegungsschätzung für P- Bilder schlechter wird.

Bilder

- Setzen sich aus Scheiben zusammen (slices), die Wiederum aus Makroblöcken bestehen.
- Slices dienen zur Resynchronisation und zum Auffrischen von Prädiktionswerten o.ä.



- Makroblock bei MPEG 1
  - vier 8x8-Luminanzblöcke
  - zwei 8x8-Chrominanzblöcke
- Makroblock bei MPEG 2 dito
  - jedoch 2,4 oder 8 Chrominanzblöcke möglich
    - → hängt von der Farbunterabtastung zusammen
      - MPEG 1: 4:2:0
      - MPEG 2: 4:2:0 oder 4:2:2 oder 4:4:4

3. MPEG (Moving Pictures Experts Group - 1988) (2/2)

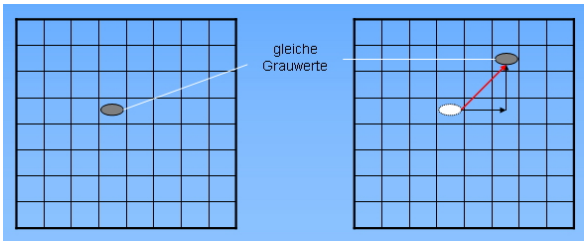
Codierung von I-Bilder

- folgt dem gleichen Prinzip wie die JPEG-Codierung
  - Jeder 8x8-Block wird DCT- transformiert und anschließend quantisiert
  - DC-Koeff. werden differentiell verarbeitet
  - AC-Koeff. Lauflängen-Huffman-Codierung
  - Quantisierungswert
    - MPEG 1: 8
    - MPEG 2: 4,2,1
- Quantisierung: $$q[k,l] = \frac{16 \cdot X[k,l] + \mathrm{sgn}(X[k,l]) \cdot q_{scale} \cdot Q_{k,l}}{2 \cdot q_{scale} \cdot Q_{k,l}}$$
- Rekonstruktion: $$X[[k,l]]_Q = \frac{q[k,l] \cdot q_{scale} \cdot Q_{k,l}}{8}$$
  - Qk,l – frei wählbare Quantisierungswerte sind in einer Matrix abgelegt
  - qscale (1…31) beeinflusst Quantisierungsstärke zusätzlich → Variation der Kompression
- Prädiktionsreihenfolge der DC-Koeff. ist durch Makroblockstruktur vorgegeben
- Codierung der DC-Koeff.: siehe JPEG
- Codierung der AC-Koeff.:
  - werden nicht in Kategorien eingeteilt
  - → kombinierte Datensymbole direkt aus Lauflänge & Koeffizientenwert
  - sonst siehe JPEG (Zick-Zack-Scan, Lauflängencodierung, Huffman-Codierung, EOB)

Codierung von P- und B- Bilder

- Encoder entscheidet auf Makroblock-Ebene, ob Block
  - inter-codiert (bei erfolgreicher Prädiktion durch Bewegungskompensation)
  - intra-codiert (bei nicht voraussagbaren Bildinhalten)
  - Bewegungsvektor ungleich Null übertragen werden muss
  - zusätzlich bei B-Bilder:
  - Entscheidung ob Vor-, Rück- oder kombinierte Prädiktion zum Einsatz kommt
- Prinzip der Bitmarkierung:
  - diejenigen 8x8 Blöcke (Luminanz & Chrominanzen) eines Makroblocks werden gekennzeichnet, deren Quantisierungssymbole alle gleich Null sind $q[k,l] = 0 \ \forall \ k,l$
  - solche Blöcke sind bei Codierung auszulassen
  - Decoder füllt betreffenden Bereich mit Referenzdaten
- Codierung analog I-Bilder
- Modifikation der Quantisierung bei Inter-Codierung: $$q[k,l] = \frac{X[k,l]}{q_{scale} \cdot Q_{k,l}}$$
- Rekonstruktion: $$X[[k,l]]_Q = \frac{(2 \cdot q[k,l] + \mathrm{sgn}(q[k,l])) \cdot q_{scale} \cdot Q_{k,l}}{16}$$
- Alle Frequenzanteile gleich wichtig → $Q_{k,l} = 16$

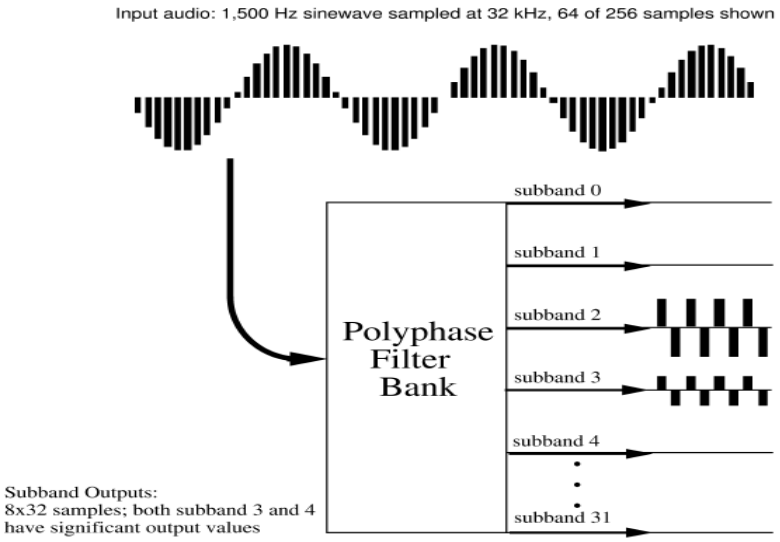
Bewegungskompensation



- basiert auf Block-Matching-Verfahren
- → Verfahren mit Prädiktion von Grauwerten
- Art und Weise der Suchstrategie ist nicht vorgeschrieben
- pro Makroblock wird Bewegungsvektor ermittelt und übertragen
  → Einsparung von Daten

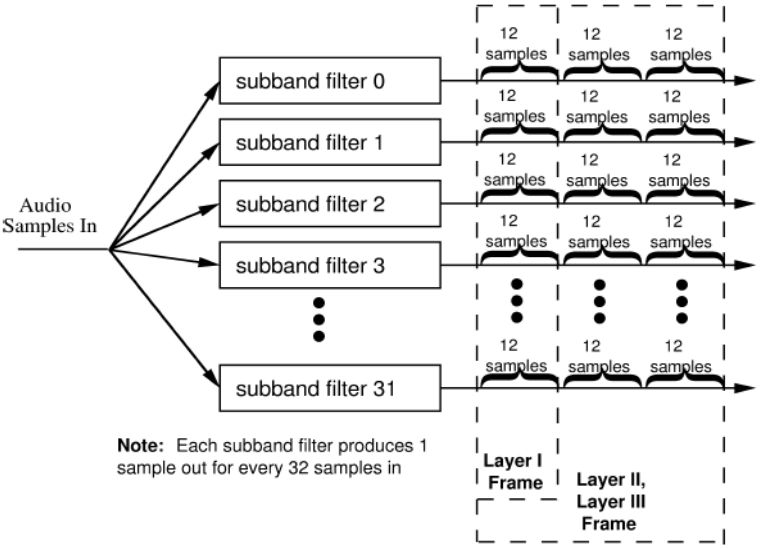
4. MPEG - Audio (1/2)

- MPEG-Audio – Motion Pictures Experts Group MPEG 1 – Audio Layer I-III
- bekannteste Form: MP3 (MPEG1-Layer3)
- nutzt das begrenzte Hörvermögen des Menschen aus (20Hz-20KHz)
  → Verlustbehaftet
- Das menschliche Gehör besitzt die Eigenschaft, dass ein lauter Ton bei einer Frequenz die Empfindlichkeit bei Frequenzen in der Nähe des lauten Tones verringert.
- Wie?
  - Signal wird durch eine Filterbank in verschiedene Bänder unterteilt.
  - Die Daten in diesen Bändern werden mit geringster Bitzahl quantisiert, bei der das Quantisierungsrauschen für das menschl. Ohr nicht hörbar ist.
  - = Maskierung
- → Kompression



Layer I:
- 12 Samples werden zu einer Gruppe zusammengefasst.
- → 32 Gruppen mit je 12 Samples werden zu einem Rahmen zusammengefasst.
- → Rahmen bekommt Header & CRC Checksumme & evtl. zusätzliche Daten
- 1 Frame=384 Samples

Layer II/III:
- 3x12 Samples pro Band
- 1Frame=1152 Samples



Layer III

Die Auflösung im Frequenzbereich wird erhöht.
- → Ausgänge der Filterbank werden mit einer MDCT weiterverarbeitet.
- Erhöhung der Auflösung im Frequenzbereich bewirkt eine Verringerung der Auflösung im Zeitbereich!
  - → verschiedene Blocklängen:
    - 36 oder 12 Samples
    - MDCT überlappt sich zu 50%
      - → mehrere Fenster nötig
- 3 Verschiedene „Betriebsarten“:
  1. alle Bänder mit 36 Samples (lange MDCT)
  2. alle Bänder mit 12 Samples (kurze MDCT)
  3. gemischt:
     - 2 niedrigsten Frequenzbänder mit langer MDCT
     - übrige Bänder mit kurzer MDCT

4. MPEG - Audio (2/2)

Modifizierte Diskrete Kosinus Transformation

- MDCT: $X_t(m) = \sum_{k=0}^{n-1} f(k) \cdot x_t(k) \cdot \cos\left[\frac{\pi}{2n} \cdot \left(2k + 1 + \frac{n}{2}\right) \cdot (2m+1)\right]$, $m = 0 \ldots \frac{n}{2} - 1$

- IMDCT: $Y_t(p) = f(p) \cdot \frac{n}{4} \cdot \sum_{m=0}^{\frac{n}{2}-1} X_t(m) \cdot x_t(k) \cdot \cos\left[\frac{\pi}{2n} \cdot \left(2p + 1 + \frac{n}{2}\right) \cdot (2m+1)\right]$, $p = 0 \ldots n - 1$

  - x(k) …Samples im Zeitbereich
  - $x_t(k)$, k=0…n-1 …Samples zur Berechnung der Werte $X_t(k)$, k=0…n/2-1 des t-ten Blocks im Frequenzbereich
  - f(k) …Werte des jeweiligen Fensters, das verwendet wird

5. Quellen:

Internet:

- www.jpegclub.org
- www.mpeg.org
- www.google.de → Suchbegriffe wie „JPEG“, „Chroma Subsampling“, etc.

Bücher:

- Data Compression – The Complete Reference, 3rd Edition (David Salomon)
- Bilddatenkompression (Tilo Strutz)