

Datenkompression: Anwendungen

H. Fernau

email: `fernau@uni-trier.de`

WiSe 2008/09
Universität Trier

Anwendungskapitel

Erinnerung: JPEG im vorigen Abschnitt

Bewegtbildkompression (MPEG)

(etwas) Kompression von Audiodaten

Aspekte von Bewegtbildern

Bewegtbilder vs. *Standbilder*: **Zeit** als weitere Dimension

aufeinander folgende Bilder einander „ähnlich“ \rightsquigarrow
prädikative Ansätze in Zusammenhang mit Differentialcodierung

Weiterhin: Ähnlichkeiten innerhalb einzelner Bilder

Beachte:

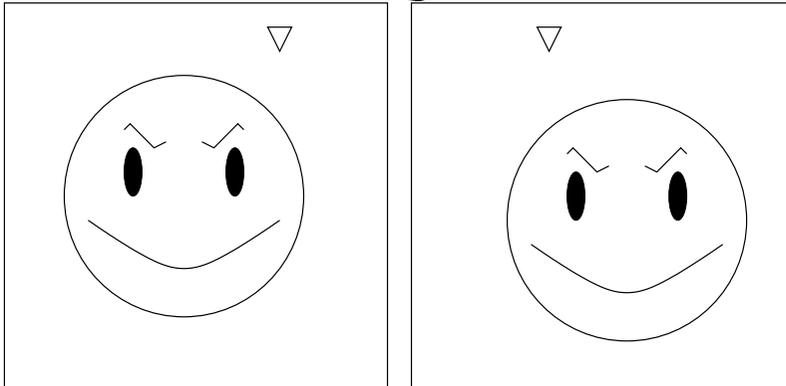
- (1) Störungen in „ruhigen Zonen“ erscheinen dem Menschen bei Bewegtbildern auffälliger als bei Standbildern
- (2) Kantenstörungen werden bei Standbildern stärker als bei Bewegtbildern wahrgenommen.

Unterschiedliche Anforderungen seitens der Anwendungen:

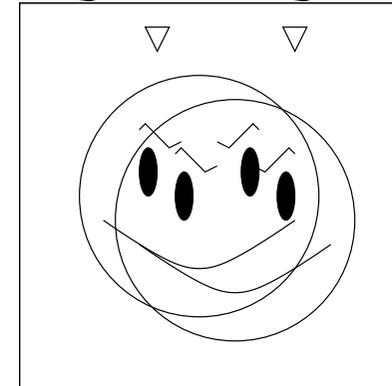
- (1) *Ein-Weg-Kommunikation* (verbreitetste Form der Bewegtbildnutzung): *asymmetrische Verfahren* denkbar;
- (2) *Zwei-Weg-Kommunikation* (Bildtelefon, Videokonferenz, . . .): schnelle *symmetrische Algorithmen* gefordert.

Bewegungskompensation

Zwei aufeinander folgende Bilder...

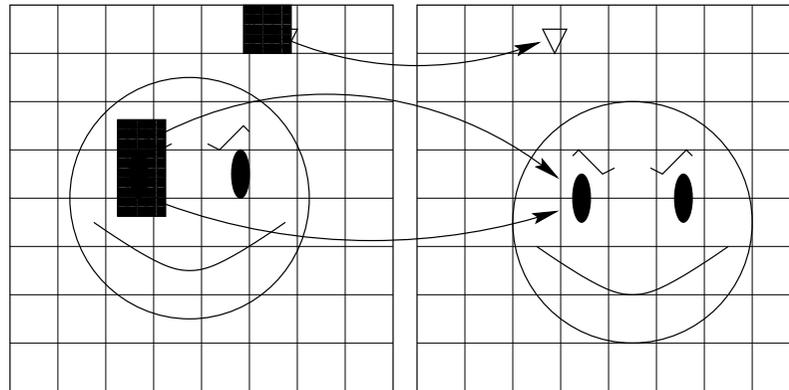


überlagert dargestellt



pixelweise Differenz von Bildern \leadsto **keine gute Dekorrelation:**
verschiedene Objekte (z.B.: Kopf / Dreieck) bewegen sich unterschiedlich
 \leadsto Bewegungs-*Erkennung*, *-Modellierung*, *-Kompensation*

Blockbasierte Bewegungskompensation



Unterteile das zu codierende Bild wird in $M \times M$ Pixelblöcke. Im „früheren“ Bild wird ein „möglichst ähnlicher Block“ (z.B. durch MSE gemessen) gesucht und eine *Fehlerkorrektur* nebst dem *Bewegungsvektor* übertragen.

Überschreiten eines *Schwellwertes* \leadsto codiere Block direkt.

Suchaufwand zum Auffinden „ähnlicher Blöcke“ erheblich, wenn auch kleiner als bei fraktaler Kompression.

Beachte, dass nicht nur die Blöcke im „früheren Bild“ untersucht und verglichen werden, sondern auch feinere Verschiebungen.

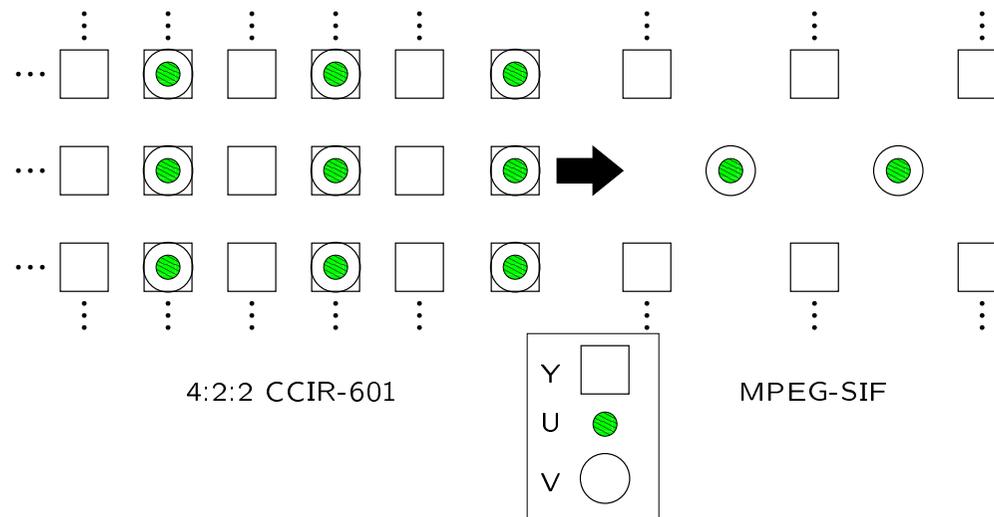
Zwei digitale Bildformate

Farbformate: *RGB*, *YIQ* (USA), *YUV-Signal* in Europa (PAL),
 dabei *U* Rot-Cyan Balance, *V* Gelb-Blau Balance;
 Digitalfernsehnorm (CCIR-601) *YU'V'*, Umrechnung gemäß der Formeln:

$$Y = 0,229R + 0,587G + 0,114B$$

$$U' = B - Y$$

$$V' = R - Y$$



Digitaler Bildformatstandard CCIR 601

... eigentlich eine ganze Schar von Formaten.

Grundfrequenz stets 3,375 MHz.

Beliebtestes Format: 4:2:2, d. h.,

das Luminanzsignal wird mit $4 * 3,375 = 13,5$ MHz,

die Chrominanzsignale mit je $2 * 3,375 = 6,75$ MHz übertragen.

Ziel: Übertragung 30 Bilder / Sekunde zu 720×480 Bildpunkten.

Für jeden Bildpunkt werden ein Luminanzsignal und für jeden zweiten noch zwei Chrominanzsignale erzeugt.

Diese werden (normalisiert und quantisiert) mit je einem Byte übertragen, so dass $60 \times 720 \times 480$ Bytes, also ca. 2 MB, je Sekunde zu übertragen sind.

Um weichere Bildübergänge zu erzielen, wird jedes Bild in zwei Durchgängen übertragen: zuerst die geradzahlig und dann die ungeradzahlig nummerierten Zeilen.

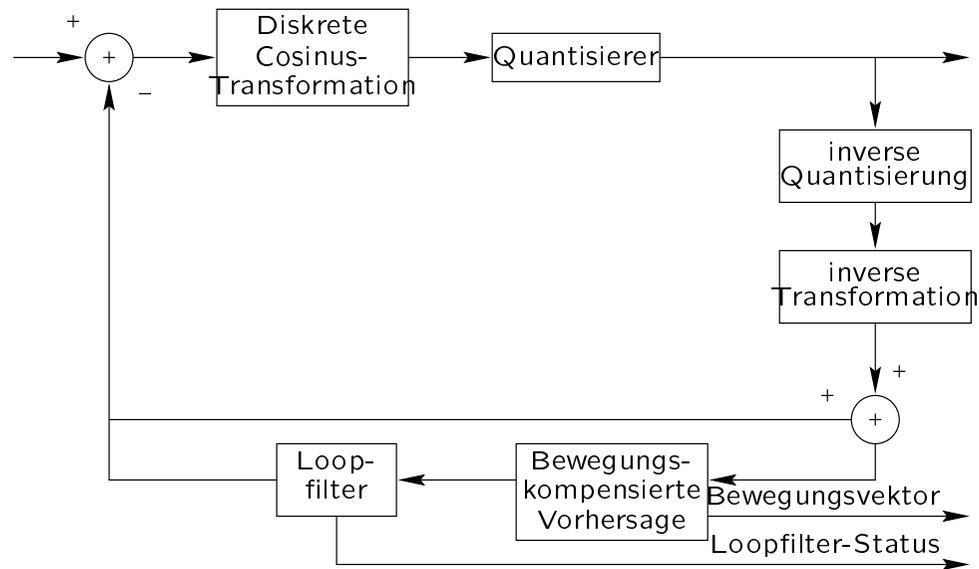
Das in obiger Abb. dargestellte **MPEG-SIF-Format** (für MPEG-1 Standard) viertelt die zu übertragende Datenmenge in der angedeuteten Weise.

Für Videokonferenzen haben sich noch gröbere Auflösungen (ähnlich aufgebaut wie MPEG-SIF) als Standards durchgesetzt (wieder 30 Bilder je Sekunde):

| | | |
|------|---------------------------|-----------------------------|
| CIF | Luminanz 288×352 | Chrominanz 144×176 |
| QCIF | Luminanz 144×176 | Chrominanz 72×88 |

Videokonferenz

Der ITU-T H.261 Codierer — Grundschemata



Bewegungskompensation beim ITU-T H.261 Standard

Der Luminanz- bzw. die Chrominanzanteile eines Bildes werden jeweils in 8×8 große Blöcke unterteilt.

Ein *Macroblock* besteht aus 4 *Luminanzblöcken* und je einem (entsprechenden) *Chrominanzblock*.

Zu jedem Macroblock wird der günstigste „Partner“ im früheren Bild gesucht, wobei nur die Luminanzblöcke zur Differenzbildung herangezogen werden.

Der Bewegungsvektor der entsprechenden Chrominanzblöcke wird durch Halbierung des für die Luminanz berechneten gewonnen.

Der Suchumfang wird auf ± 15 Pixel in vertikaler und horizontaler Richtung für jeden Macroblock eingeschränkt.

Ein 4×4 Beispielblock...

| | | | |
|-----|-----|-----|-----|
| 110 | 218 | 116 | 112 |
| 108 | 210 | 110 | 114 |
| 110 | 218 | 210 | 112 |
| 112 | 108 | 110 | 116 |

... wird zuerst horizontal...

| | | | |
|-----|-----|-----|-----|
| 110 | 165 | 140 | 112 |
| 108 | 159 | 135 | 114 |
| 110 | 188 | 187 | 112 |
| 112 | 109 | 111 | 116 |

... und dann vertikal gefiltert.

| | | | |
|-----|-----|-----|-----|
| 110 | 165 | 140 | 112 |
| 108 | 167 | 148 | 113 |
| 110 | 161 | 154 | 113 |
| 112 | 109 | 111 | 116 |

Der **Loopfilter** ist ein Tiefpassfilter, der für „innere Blockpunkte“ (im Beispiel sind dies bei horizontaler Anwendung die beiden mittleren Spalten und bei vertikaler die beiden mittleren Zeilen) den „neuen“ Pixelwert durch die Formel

$$x_{neu} = x/2 + x'/4 + x''/4$$

berechnet (x' bzw. x'' : Pixelwerte der beiden Nachbarn).

So ergibt sich der Wert in der ersten Zeile, zweite Spalte in der zweiten Tab. durch $165 = 110/4 + 218/2 + 116/4$ (ganzzahlige Division!), und der Wert in der zweiten Zeile, erste Spalte in der dritten Tab. durch $108 = 110/4 + 108/2 + 110/4$ (beachte, dass bei „normaler Division“ sich hier 109 ergäbe).

Wie im Schema zu sehen, ist solch ein Loopfilter in die Macroblocksuchschleife eingebaut; genauer gesagt: passt ein solchermaßen tiefpassgefilterter Luminanzanteil des Macrovergleichsblocks „am besten“ bei der „Partnersuche“, so wird dies dem Decodierer mit einem Extra-Bit (*Loopfilter-Status*) mitgeteilt.

Weitere Bestandteile im ITU-T H.261 Standard

DCT, angewendet jeweils auf 8×8 -Blöcke (wie bei JPEG)

ABER: arbeitet nun teilweise auf Pixeln [sog. „intra mode“] und auf Pixeldifferenzen [sog. „inter mode“] (Bewegungskompensation!) \rightsquigarrow Koeffizienten haben sehr unterschiedliche Charakteristika, was die einheitliche Quantisierung erschwert.

\rightsquigarrow H.261 kann zwischen zweiunddreißig verschiedenen Gleichquantisierern für die AC-Koeffizienten von Macroblock zu Macroblock hin- und hergeschalten.

Da die Übertragung der quantisierten DCT-Koeffizienten wie bei JPEG nach Zickzackabtastung geschieht, ist bei den AC-Koeffizienten Null stets auch ein Repräsentant im Quantisierer.

Übertragungsrate

Beachte, dass **große Schrittweiten** zwar evtl. eine schlechte Güte, andererseits aber einen ziemlich großen „Schwanz“ von Nullen mit sich bringen, die Quantisiererwahl also auch die Übertragungsrate beeinflusst.

Um der Notwendigkeit zu begegnen, für jeden Macroblock einen eigenen Quantisierschlüssel zu übertragen, werden 3×11 Macroblöcke zu einer **Block-gruppe** (GOB $\hat{=}$ engl.: group of blocks) organisiert, und für jede GOB wird ein Default-Quantisierer im Kopf übertragen.

Bei Videokonferenzen steht den Teilnehmern nur eine beschränkte Übertragungsrate zur Verfügung.

Andererseits ist —im wesentlichen bedingt durch unterschiedlich „günstige“ zu codierende Lauflängen und Nullschwänze nach der Quantisierung der zick-zackabgetasteten DCT-Koeffizienten— je Block eine unterschiedliche Anzahl Bits zu übertragen.

↪ Die Daten werden vor dem Versenden zunächst kurz zwischengepuffert (nicht im Schema dargestellt).

unerwünschte Übertragungsverzögerungen ↪ Puffer recht klein.

Droht **Überlauf** ↪ eine entsprechende Nachricht geht an den „Quantisierer“, die Rate zu senken, sei es durch Wahl eines Quantisierers mit größerer Schrittweite oder (im schlimmsten Fall) durch Auslassen eines Blocks bei der Übertragung.

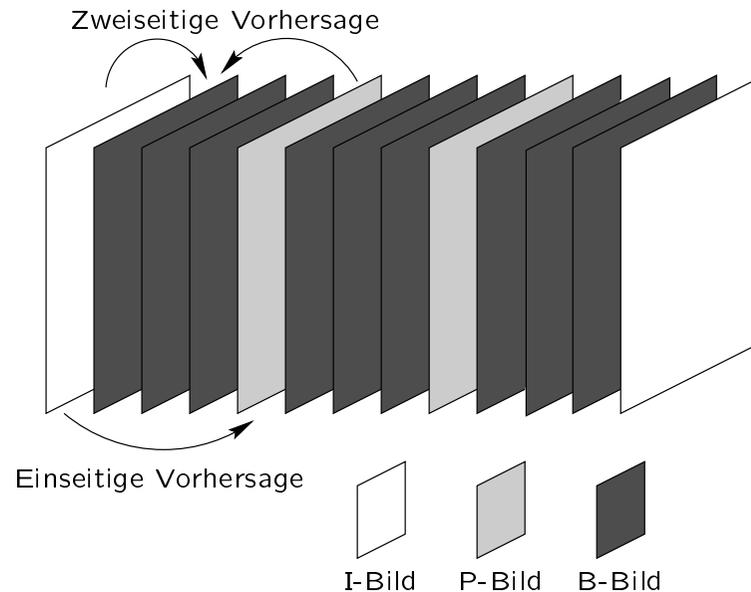
MPEG Standards

„Moving Pictures Expert Group“: eine Reihe von Vorschlägen zur Bewegtbildcodierung

Unterschiede in ihrer Funktionalität: MPEG-1 sollte für Übertragungsraten um 1,5 Mbps gute Ergebnisse liefern, MPEG-2 für 10 Mbps und MPEG-3 für 40 Mbps.

Wir werden im Folgenden MPEG-1, MPEG-2 und (kurz) MPEG-4 diskutieren.

MPEG-1



Das Grundschemata ist auch hier gültig.

Wesentlicher Unterschied zu H.261 ist die primäre Ausrichtung der MPEG-Standards auf Multimedia-Anwendungen \rightsquigarrow

„wahlfreier Zugriff“ auf die Bildfolge muss gewährleistet werden.

Schon beim unidirektionalen (digitalen) Fernsehen will der Zuschauer ja evtl. sich später in einen laufenden Film einschalten. . .

Aufsetzen und Zeitprädikation bei MPEG-1

Nicht zeitprädikativ codierte *I-Bilder* werden periodisch übertragen.

P-Bilder sind dagegen „vorhersagecodiert“, (prediction coded); dies entspricht im Wesentlichen dem H.261-Standard, nur dass —aus gleich verständlich werdenden Gründen— P-Bilder (die mit den I-Bildern zusammen auch *Ankerbilder* genannt werden) nicht unbedingt auf das letzte Bild der zu sendenden Bildfolge, sondern beispielsweise auf das fünftetzte zurückgreifen; dies ist natürlich auch bei der Bewegungskompensation zu berücksichtigen (Es wird empfohlen, den Suchraum für Macroblöcke in Abhängigkeit vom „Abstand“ gesendeter Ankerbilder anzupassen).

Um eine gute Komprimierung zu gewährleisten, gibt es in MPEG zusätzliche *B-Bilder* (engl.: „bidirectionally prediction coded“), die Korrelationen sowohl zum in der zu sendenden Bildfolge vorausgehenden Ankerbild als auch zum in der zu sendenden Bildfolge später kommenden Ankerbildes ausnutzen, was bei Szenenwechseln sinnvoll ist.

Eine *Bildgruppe* (*GOP* $\hat{=}$ engl.: „group of pictures“) enthält mindestens ein I-Bild und ist die kleinste Bildfolge, in die sich ein Benutzer zuschalten kann.

Die GOP-Struktur von MPEG-1 ist auch vorteilhaft gegen eine mögliche Fehlerfortpflanzung; aus ähnlichen Gründen wirkt sich die rückgekoppelte Ratensteuerung nur auf die Übertragung der „unwichtigen“ B-Bilder aus.

Anzeigeordnung einer GOP

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|
| I | B | B | P | B | B | P | B | B | P | B | B | I |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Übertragungsordnung der selben GOP

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|
| I | P | B | B | P | B | B | P | B | B | I | B | B |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

Wegen der Abhängigkeiten zwischen den verschiedenen Bildern unterscheidet man zwischen der *Anzeigeordnung*, also der Reihenfolge, in der die Bilder anzuzeigen sind, und der *Übertragungsordnung*. Wie im Beispiel zu erkennen ist, werden zunächst die Ankerbilder übertragen und dann die sich unmittelbar darauf beziehenden B-Bilder.

MPEG-1 liefert eine Qualität, die mit VHS-Videos vergleichbar ist, wenn nicht zuviel Bewegung darin enthalten ist.

MPEG-2 ist eigentlich eine Sammlung von Algorithmen (empfehlungen) bzw. Spezifikationen, jeweils angepasst auf definierte sog. *Profile* (Anforderungen) und *Levels*.

Es gibt fünf Profile; in der Reihenfolge aufsteigender Anforderungen: *einfach* (engl.: simple), *Haupt-* (engl.: main), *SNR-skalierbar*, *räumlich skalierbar* (engl.: spatially scalable) und *hoch*.

Höhere Profile können jeweils niedrigere decodieren.

Die vier Levels sind: *niedrig*, *Haupt-*, *hoch 1440* und *hoch*.

Sie definieren die Bildauflösung (352×240 , 720×480 , 1440×1152 und 1920×1080) und damit die zu übertragende Datenmenge.

Übrigens benutzt digitales HDTV MPEG-2 mit Hauptprofil und hohem Level. Die einfachen und Haupt-Profile entsprechen im Wesentlichen dem zu MPEG-1 Gesagten; das einfache Profil verwendet keine B-Bilder.

Die drei übrigen Profile benutzen mehr als einen Bitstrom zur Codierung.

Der Grundbitstrom ist eine niederratige Codierung der Bildfolge; die anderen Bitströme dienen zur Bildqualitätserhöhung.

Bei Netzwerkübertragung braucht bei Überlast nur der Grundbitstrom übertragen zu werden, um so die Bitrate zu senken. Daher werden diese Profile auch „skalierbar“ genannt.

Ideen ähnlich der von der fortschreitenden Bildübertragung kommen zur Anwendung, nur dass die Filter sowohl zeitlich als auch räumlich arbeiten, s. nächste Folie.

MPEG-2 gestattet —im Gegensatz zu MPEG-1— aufwändigere Bewegungskompensation, indem

- auch Informationen des schon übertragenen Teils des aktuellen Bildes benutzt werden und
- unterschiedliche Bewegungsrichtungen von 16×8 -Sub-Macroblöcken berücksichtigt werden (können).

MPEG-4 ist eine Antwort auf (im wesentlichen auf die Zukunft gerichtete) Schlagwörter wie „interaktives Fernsehen“, „multiperspektives Fernsehen“, „Video on demand“, etc.

Funktionalitäten bei MPEG-4

| Funktionalitäten | Kategorien |
|--|-----------------------------------|
| 1. Inhaltsbasierte Skalierbarkeit 2. Inhaltsbasierte Interaktivität und Editieren von Datenströmen 3. Werkzeuge für den inhaltsbasierten Zugriff auf Multimedia-Daten 4. Hybride Kodierung von Daten mit natürlichem und synthetischem Ursprung | I. Inhaltsbasierte Interaktivität |
| 5. Codierung mehrfacher, nebenläufiger, zeitlich verzahnter Datenströme 6. Verbesserte Kodierungsleitung | II. Kompression |
| 7. Robustheit in fehlerträchtigen Umgebungen 8. Verbessertes Zeitverhalten bei wahlfreiem Zugriff | III. Universeller Zugriff |

Es folgen kurze **Erläuterungen zu den Funktionalitäten**:

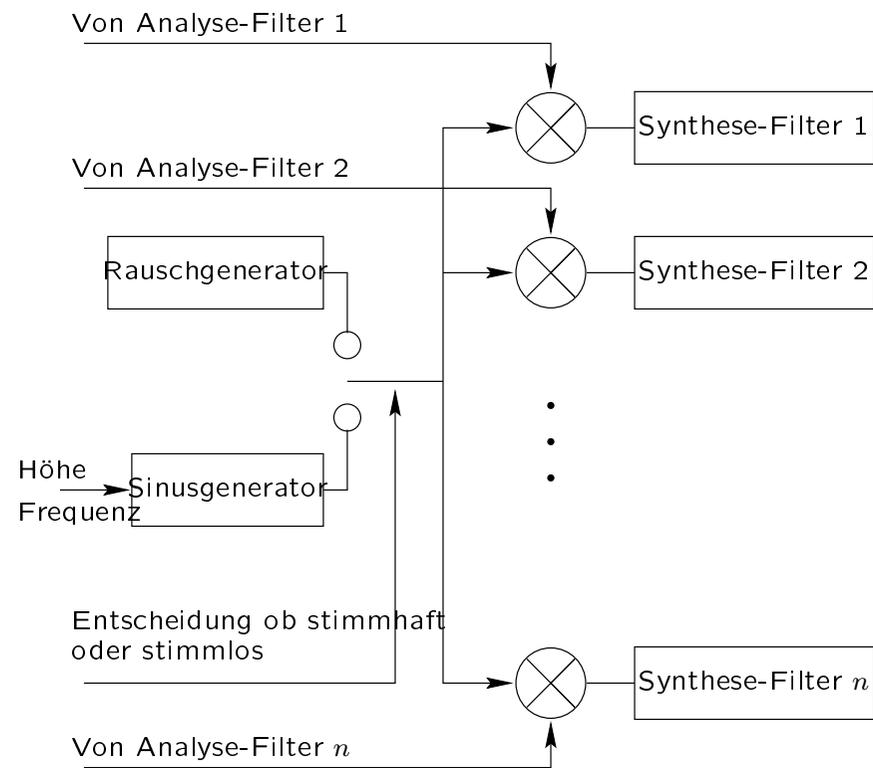
1. Ein Bild wird aus verschiedenen, priorisierten *audiovisuellen Objekten* (AVOs) zusammen gesetzt gedacht, die jeweils mit unterschiedlicher Genauigkeit übertragen werden können. Es ist schwierig, solche AVOs automatisch zu identifizieren; dem Hersteller von MPEG-4-Filmen ist es aber freigestellt, diese explizit zu definieren. Die Übertragungseigenschaften (Rate, SNR) verschiedener AVOs können unterschiedlich sein.
2. Diese Funktionalität (die z. B. zu einem Hobby „Editieren von Filmen“ führen könnte) soll dadurch hergestellt werden, dass Bewegungsinformationen von AVOs durch (C++) *Kompositionsskripte* übertragen werden.
3. Werkzeuge können sein: Indizes, Hyperlinks, ...
4. Hybride Codierung soll auch *Multiperspektivität* unterstützen. So könnte sich ein Zuschauer in der Zukunft eine Torraumszene eines Fußballspiels wahlweise aus einer Torwart- oder einer Tribünenperspektive betrachten.
5. Codierungen mehrfacher Datenströme können beispielsweise mehrere Tonspuren (Sprachen) der selben Szene oder mehrere Bildperspektiven (3D-Fernsehen) betreffen; die dabei erzeugte Redundanz sollte durch intelligente Komprimierung kompensiert werden.
8. Z. B. soll der „schnelle Suchlauf rückwärts“ für selektierte AVOs möglich sein.

Audio-Daten

Natürlich sind bei Filmen beispielsweise nicht nur Bild-, sondern auch Toninformationen zu übertragen. Für eine gute Kompression von solchen Audio-Daten (insbesondere Sprache) ist zu berücksichtigen:

1. eingeschränkter hörbarer Frequenzbereich;
2. besondere Empfindlichkeit für Tonhöhen;
3. Lautstärkewahrnehmung ist mit Tonhöhenwahrnehmung korreliert;
4. in europäischen Sprachen gibt es deutlich voneinander trennbare „stimmhafte“ (sinusartige) und „stimmlose“ (rauschartige) Laute; ein Decodierer für solche Sprachen kann daher wie auf der nächsten Folie dargestellt arbeiten.

Schema eines Sprachdecodierers



Kommentare

Wie kann der Codierer eine Entscheidung darüber treffen, ob ein Laut stimmhaft oder stimmlos ist? Eine Möglichkeit ist die Filteranalyse; denn Rauschsignale sind hochfrequenter als sinusartige.

Der Tonhöhenempfindlichkeit wird beispielsweise im auch bei MPEG verwendeten CELP („code excited linear prediction“ von Atal und Schroeder) durch eine spezielle adaptive Codebuchtechnik für die Erregungssignalformen Rechnung getragen.