

Datenkompression und Datenreduktion - Reduzierung von Datenraten und -mengen

Das Bestreben Datenmengen zu verkleinern, zu komprimieren oder zu reduzieren, ist kein videotypisches Problem. Überall dort, wo Daten gespeichert oder übertragen werden, ist es letztlich die Beschränktheit der Speicher- bzw. Übertragungskapazität, die eine Verringerung der Datenflut wünschenswert erscheinen lässt. Das Problem ist aber in der Fernseh- und Video-Technik deshalb besonders akut, weil bei der digitalen Video-Signalverarbeitung besonders hohe Datenraten entstehen.

Bei der Video-Produktion kommen zunehmend mehr rechnergestützte, sogenannte „Workstations“ zum Einsatz. Deren standardisierte und nicht für die Video-Produktion speziell konzipierten technischen Rahmenbedingungen beispielsweise die Übertragungsrate vom Rechner auf die Festplatte und zurück via SCSI- oder PCI-Bus - bzw. die Rechenleistung, welche die Verarbeitung in Echtzeit gewährleisten muss, machen eine Verringerung der Datenrate erforderlich. Das gleiche gilt für die Speicherkapazität von Festplatten, durch welche die Dauer des zu speichernden Video-Materials begrenzt wird. Auch das digitale MAZ-Format Digital Betacam kann, unter Beibehaltung der technischen Parameter des analogen Betacam-Formates, nur einen Datenstrom von etwa 127 MBit/s aufzeichnen und arbeitet mit Datenreduktion.

Des Weiteren macht auch die Ausstrahlung digitaler Fernsehsignale, die die beschränkten Kapazitäten der bisher rein analog genutzten Ausstrahlungskanäle nicht sprengen soll, Datenkompression bzw. -reduktion notwendig.

Es gibt im wesentlichen zwei Stufen der Datenreduzierung, die Kompression und die Reduktion. Mit Kompression sei im Rahmen dieses Kapitels jener Prozess bezeichnet, der reversibel - d.h. ohne Verlust an Informationen erfolgt. In den Quelldaten stecken in der Regel überflüssige Daten, die mit Redundanz bezeichnet werden. Diese zu entfernen, ist die Aufgabe der Datenkompression. Je mehr redundante Daten man entfernt, desto mehr nähert man sich dem sogenannten „Entropie-Punkt“ es verbleibt die reine Information.

Eine Reduzierung über diesen Entropie-Punkt hinaus bezeichnet man als Reduktion. Sie ist mit einem echten Verlust an Information verbunden. Diesen Verlust so zu gestalten, dass wirklich nur solche Daten verloren gehen, die für den Empfänger - in diesem Fall das visuelle System - irrelevant sind, ist die Aufgabe der Irrelevanzreduktion. Das bedeutet im Fall der Video-Bilder, dass ausschließlich solche Bildinhalte weggelassen werden, deren Fehlen unter den jeweiligen Betrachtungsbedingungen für das Auge nicht bemerkbar sind.

Eine weitere Reduzierung über die Irrelevanz hinaus stellt dann eine Relevanzreduktion dar. Diese wird da angewendet, wo zum einen besonders hohe Kompressionsraten (s. auch Begriffsdefinition) erforderlich sind und zum anderen die Aussage der verbleibenden Nachricht nur im Kern verständlich sein muss.

Für Video-Bilder bedeutet dies, dass die Bildqualität u.U. so weit herabgesetzt wird, daß bei formatfüllender Bildwiedergabe die Störungen deutlich, die Bildinhalte nur noch grob gerastert wahrnehmbar sind. Dies ist zum Beispiel beim Off-Line-Schnitt der Fall, wo in der

Regel viel Bildmaterial gespeichert und gesichtet werden muss -und es dabei nur in zweiter Linie auf die Bildqualität ankommt. Die Grenzen der Irrelevanzreduktion liegen auch bei der Off-Line-Anwendung in der Regel da, wo jene Bildinhalte verloren gehen, die für die Synchronität entscheidend sind. Dies sind z.B. Lippenbewegungen oder fallende Gegenstände.

Einige ausgewählte Grundlagen

Entropie-Codierung bei Textdateien

Geht man davon aus, dass eine Buchstabenschrift etwa dreißig Buchstaben zuzüglich Leer- und Satzzeichen beinhaltet, so wären diese Zeichen mit binären Codewörtern der Länge 6 Bit - entsprechend $2^6 = 64$ Zeichen - darstellbar. Ein Text von 1000 Zeichen benötigt demzufolge 6000 Bit zur binären Darstellung.

Nun weißt aber jede Sprache eine bestimmte statistische Häufigkeitsverteilung der verschiedenen Zeichen auf. Im Deutschen kommen generell alle Vokale - insbesondere <e> - sowie einige Konsonanten wie <l>, <r>, <s> und <t> sehr viel häufiger vor als z. B. die Konsonanten <x>, <y>, <z>. Entsprechend gering ist der „Überraschungseffekt“ wenn z. B. ein <e> erscheint - man sagt auch der Informationsgehalt eines Zeichens ist umgekehrt proportional zu seiner Häufigkeit bzw. der Wahrscheinlichkeit seines Auftretens. Obwohl also der Buchstabe <e> einen geringen Informationsgehalt aufweist, wird er mit genau so vielen Zeichen dargestellt wie z.B. der Buchstabe <z>, der eine geringere Auftretswahrscheinlichkeit und somit einen hohen Informationsgehalt aufweist. Innerhalb einer derartigen, gleichmäßigen Codierung enthält die Darstellung von Zeichen mit hoher Auftretswahrscheinlichkeit sehr viel „Redundanz“.

Will man Daten einsparen, so kann man, aus der statistischen Analyse heraus, solche häufigen Zeichen mit wenigen Bits und die seltenen Zeichen - ihrem hohen Informationsgehalt entsprechend - mit vielen Bits darstellen. Kurze Datenworte erscheinen häufig und lange Datenworte selten. Das Ergebnis ist eine Reduzierung der Datenmenge um 40 bis 50%, nachdem diese Redundanz weitgehend eliminiert wurde.

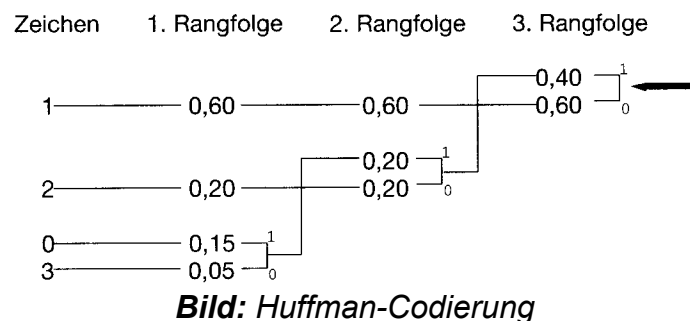
Huffman-Codierung

Genau diesen Weg verfolgen Codierungsansätze nach Huffman - hier der Entwurf eines Huffman-Codes für vier Zeichen, die mit unterschiedlicher relativer Häufigkeit auftreten.

Zeichen	binäre Darstellung	rel. Häufigkeit	Huffman Code	Länge	Länge x rel. Häufigkeit
0	00	0,15	111	3	0,45
1	01	0,60	0	1	0,60
2	10	0,20	10	2	0,40
3	11	0,05	110	3	0,15
Summe					1,60

Der Huffman-Code wird wie folgt ermittelt:

- Die Zeichen werden nach ihrer Häufigkeit geordnet.
- Die Häufigkeiten der unteren Zeichen werden addiert und mit der Summe wird eine neue Rangfolge gebildet. Dieser Vorgang wird wiederholt, bis es nicht mehr geht.
- An den jeweils oberen Summanden wird eine „1“ geschrieben, an den jeweils unteren eine „0“.
- Der Huffman-Code eines Zeichens ergibt sich, indem man das gezeigte Schema von rechts zu diesem Zeichen hin durchläuft und dabei alle „0-en“ und „1-en“ aufschreibt.



Der große Vorteil eines Huffman-Codes liegt - neben seiner komprimierenden Wirkung - darin begründet, dass man die einzelnen Codeworte ohne Trennzeichen aneinander reihen kann, weil kein kürzeres Wort identisch ist mit dem Anfang eines längeren. Im gezeigten Beispiel braucht der Huffman-Code im Mittel 1,60 Bit pro Zeichen, im Vergleich zu 2 Bit in der ursprünglichen Darstellung. Sind die Häufigkeiten noch extremer verteilt, und sind die Datenworte in Binärdarstellung länger als nur 2 Bit, so sind erheblich größere Codierungsgewinne zu erwarten.

Entropie- bzw. Huffman-Codierung ist verlustfrei.

„Umquantisierung“ digitaler Daten -Verkürzung von Datenworten

Bei einer linearen Quantisierung werden den analogen Samples in einer Vergleicherschaltung die entsprechenden Binärwerte mit festen Quantisierungsstufen und konstanter Wortlänge zugeordnet. Will man nachträglich die Wortlänge -also die Anzahl und somit die Größe der Stufen - verändern, so ist dies nur in vergrößernder Weise möglich.

Eine Vergrößerung kann leicht dadurch erreicht werden, dass die vorliegende Binärzahl durch einen Faktor den man gewissermaßen als Vergrößerungsfaktor bezeichnen kann - dividiert wird. Bei einer Quantisierung mit acht Bit ergeben sich 256 Stufen. Soll nun die Wortlänge auf sieben verkürzt werden, so entspricht diese Verkürzung einer Division

durch $(28-7 = 21 \Rightarrow) 2$. Jeder Stufenwert wird durch zwei geteilt, wobei nur die geradzahigen Stufen ganzzahlige Werte ergeben. Die ungeradzahigen Stufen, die jeweils genau in der Mitte zwischen zwei ganzzahligen Werten liegen, müssen auf- oder abgerundet werden. Es verbleiben $256 : 2 = 128$ Stufen.

Diese 128 Stufen beschreiben jetzt den Pegelbereich, der vorher mit 256 Stufen beschrieben wurde. Die Stufengröße wurde verdoppelt. Der Divisionsfaktor bestimmt den Grad der Vergrößerung. Will man z.B. eine 8-stellige Binärzahl in eine 3-stellige umrechnen, so hat der Divisor den Wert $(2^3 = 2^{\sim}) 32$. Von den 256 Stufen bleiben dann nur noch 8 Stufen übrig, die Stufengröße hat sich entsprechend um dem Faktor 32 erhöht.

Man kann innerhalb des Wertebereichs von 256 Stufen auch eine nichtlineare, ungleichmäßige Umrechnung durchführen, indem man kleine Pegel unbeeinflusst lässt und höhere Pegel mit zunehmend größeren Faktoren immer mehr vergrößert. Das hätte den Vorteil, dass bei kleineren Pegeln der relative Fehler geringer bleibt als bei einer linearen Umrechnung und trotzdem im höheren Pegelbereich Daten eingespart werden.

Transformation in den Frequenzbereich

Es gibt Verfahren, mit deren Hilfe man ein zeitlich fortlaufendes Signalgemisch aus verschiedenen Frequenzen in seine Grundschwingungen aufspalten kann. Man spricht dabei von einer Transformation aus dem Zeit- in den Frequenzbereich - der Amplitudenverlauf wird nicht mehr als kontinuierliche Funktion über der Zeit, sondern als diskrete Reihe von Frequenzen dargestellt. Dadurch kann man jede beliebige Schwingungsform durch Grundschwingungen annähern. Das bekannteste Verfahren ist sicherlich die Fourier-Transformation (FT) die im Audio-Bereich eine große Rolle spielt. Bei der Datenkompression von Video-Bildern spielt hingegen die Diskrete Cosinus Transformation (DCT) die entscheidende Rolle.

Fourier-Transformation (FT)

Transformation einer Schwingung in den Frequenzbereich, bedeutet, dass nicht mehr Amplitudenwerte in Abhängigkeit von der Zeit angegeben werden (wie dies in der Regel geschieht), sondern Amplitudenwerte in Abhängigkeit von der Frequenz. Die Fourier-Transformation gibt Auskunft auf die Frage: *Welche Grundschwingungen (Sinus bzw. Cosinus) sind mit welchen Pegeln teilen in der Gesamtschwingung enthalten?*

Das Bild (auf der nächsten Seite) zeigt die Teilschwingungen einer Rechteckschwingung der Frequenz „f“ und der Amplitude 100%. Bei fallenden Pegelanteilen kommen nur ungeradzahige Vielfache der Rechteckfrequenz „f“ vor. Das zweite Bild zeigt die Rekonstruktion - die Synthese - einer Rechteckschwingung aus den ersten drei Grundschwingungen.

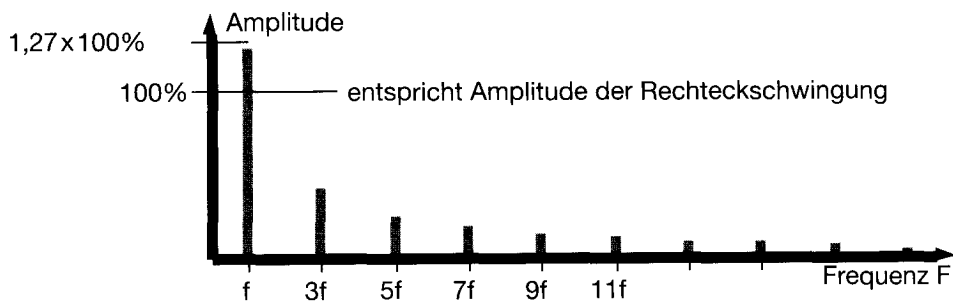


Bild: Fourier-Transformation: Rechteckschwingung der Frequenz f als Spektrum von Teilschwingungen

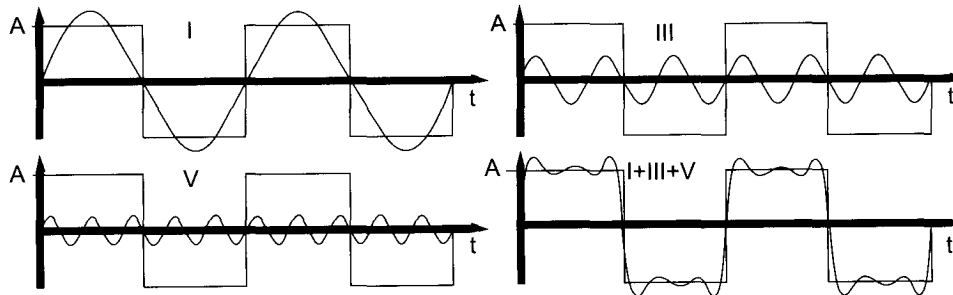


Bild: Fourier-Synthese mehrerer Grundschnwingungen zu einer Rechteckschwingung

Die Fourier-Transformation geht von unendlich vielen Teilschwingungen aus. Je mehr Teilschwingungen zur Verfügung stehen, desto exakter wird die Synthese.

Diskrete-Cosinus-Transformation (DCT)

Die DCT ist der Fourier-Transformation sehr ähnlich. Sie stellt ebenfalls einen vorgegebenen Signalverlauf durch eine Summe aus gewichteten Grundschnwingungen dar. Anders als die FT, geht die DCT aber nicht von unendlich vielen Teilschwingungen aus, sondern begnügt sich mit einem Fundus von wenigen - im Fall der Reduktion von Video-Daten acht - Grundschnwingungen ($G(U_0)$ bis $G(U_7)$), die durch Variation des Koeffizienten „ u “ gewonnen werden. Sie basieren auf der Cosinus-Funktion:

$$G(u) = \cos \{ (2x+1) (u \times 11,25^\circ) \}$$

Bei der FT laufen die Grundschnwingungen über der kontinuierlichen Variablen Zeit. Bei der DCT kann die Variable - der Laufkoeffizient „ x “ - nur ganzzahligen Werte annehmen, weil „ x “ in der hier beschriebenen Anwendung für nebeneinander liegende Bildpunkte steht und die (digitalisierten) diskreten Abtastwerte dieser Bildpunkte den Signalverlauf darstellen, der mit den Grundschnwingungen angenähert werden soll.

Das Argument der Cosinus-Funktion zeigt zwei Klammerausdrücke. Die erste Klammer enthält die Variable „ x “ und soll als Variablenklammer bezeichnet werden. Der Inhalt der zweiten Klammer ist der Form nach ein Winkel. Deshalb soll sie Winkelklammer genannt werden. Die Größe des Winkels kann durch den Faktor „ u “ bestimmt werden. Je größer

der Wert der Winkelklammer desto größer sind die Winkelschritte, die bei der schrittweisen Steigerung der Variablen „x“ entstehen, und desto schneller steigt der Winkel über „x“ an. Damit bestimmt „u“ die Frequenz der Grundschwingungen.

Für „u“=0 wird der Wert der Gesamtklammer =0° und der Wert der Cosinus-Funktion bleibt in einem solchen Fall konstant gleich 1 - das entspricht einem überlagerten Gleichanteil bzw. einer Schwingung der Frequenz Null. Bei „u“=4 schreitet die Cosinus-Funktion über „x“ in Schritten von jeweils $2 \times 45^\circ = 90^\circ$. Es ergibt sich ein vollständiger Schwingungszug (360°) über jeweils vier Bildpunkten „x“.

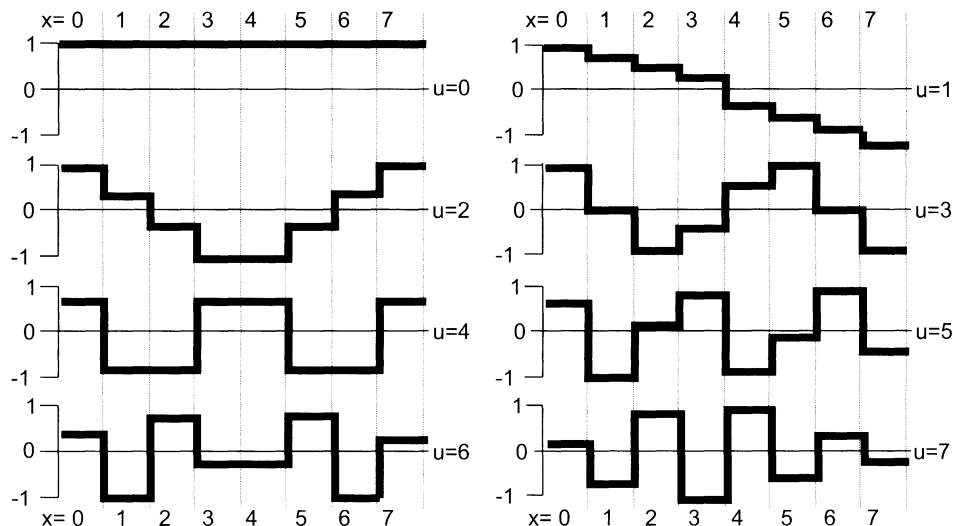
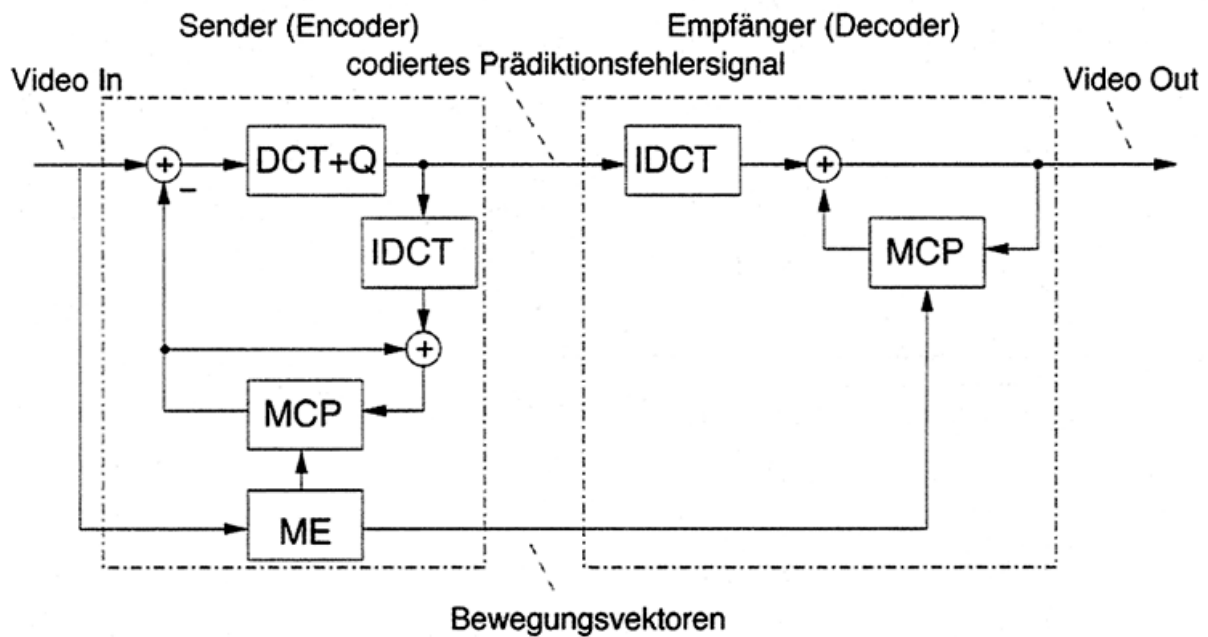


Bild: Grundfunktionen der DCT

Die DCT drückt den Signalverlauf über einer Reihe von Bildpunkten als Gewichtungsfaktoren der Grundschwingungen $G(u)$ aus. Wie bei der FT geben die Gewichtungsfaktoren den Anteil der Grundschwingungen an dem analysierten Signalverlauf an. Für jede Grundschwingung gibt es einen Gewichtungsfaktor. Die Gewichtungsfaktoren der einzelnen Grundschwingungen stehen für unterschiedlich hohe Frequenzanteile und damit, im Fall einer Anwendung der DCT auf einen Bildausschnitt, für unterschiedlich feine Details in dem betrachteten Bildausschnitt. Der Gewichtungsfaktor von „u“ = 0 gibt den mittleren Pegel in diesem Bildausschnitt an. Eine Vorlage mit großen Flächen, geringen Kontrasten und wenigen feinen Details ergibt nennenswerte Gewichtungskoeffizienten nur im Bereich der niederfrequenten Grundschwingungen, während die Koeffizienten der höherfrequenten Schwingungen vielfach nur sehr klein oder nahezu Null sind.

Die inverse DCT ist vergleichbar mit der Fourier-Synthese. Dabei wird durch Summierung der gewichteten Grundfunktionen der ursprüngliche Signalverlauf über den Bildpunkten rekonstruiert.



DCT+Q: Diskrete Cosinus-Transformation mit Quantisierung ; IDCT: Inverse DCT
MCP: Bewegungskompensierende Prädiktion (Motion Compensating Prediction),
beinhaltet Bildspeicher für bereits codierte und übertragene Bilder
ME: Bewegungsschätzung (Motion Estimation)

Laufängen-Codierung (Variable Length Codierung - VLC)

Bei der Datenkompression durch Differenz-Puls-Code-Modulation (DPCM) bzw. durch DCT (JPEG, MPEG) ergeben sich oftmals eine ganze Reihe von Datenwerten zu Null.

Um nun nicht jeden einzelnen Nullwert darstellen zu müssen, wird die sogenannte „Laufängen-Codierung“ verwendet. Sie gibt Auskunft über die Anzahl der Nullwerte und die Größe des nächsten Wertes der ungleich Null ist. Ein solches Verfahren zur Laufängen-Codierung, das „Zero-Pack-Unit“, stellt nur Werte ungleich „0“ dar und nutzt dabei z.B. eine Darstellung mit 2 x 4 Bit, wobei „R“ und „S“ als Variable für Binärwerte stehen:

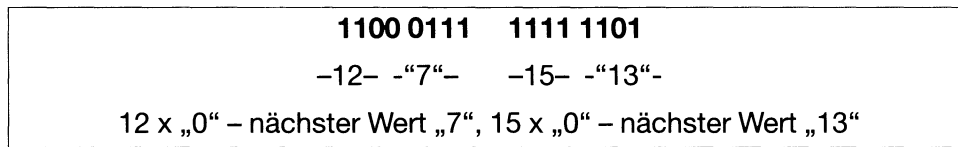
RRRR SSSS

„R“ hat die Bedeutung „Runlength“ für Lauflänge (Anzahl der Nullwerte), und „S“ hat die Bedeutung „Size“ für Größe des nächsten zählenden Wertes. Insofern sind zwei 4-Bit-Worte mit unterschiedlicher Bedeutung aneinander gereiht. Ergibt sich z.B. eine Wertefolge wie die nachstehende:

000000000000 7 0000000000000000 13

so müssten - bei einer 4-Bit Quantisierung - die 29 Ziffern von der ersten „0“ bis zum letzten Wert „13“ mit 29 x 4 Bit - entsprechend 116 Bit dargestellt werden.

Das „ZPU“ (**Zero Pack Unit**) stellt diese Zahlenfolge folgendermaßen dar, und benötigt dabei nur 16 Bit:



Kommen innerhalb einer definierten Verarbeitungseinheit (Zeile, Block ...) nur noch Nullwerte vor, so wird dies oftmals mit einem Kürzel - z.B. End Of Block (EOB) -angezeigt. Je nach mittlerer Lauflänge wird die Runlength - die Wortlänge (Anzahl der R-Stellen) angepasst. Lauflängen, die größer sind als die jeweils größte darstellbare Lauflänge, werden zerlegt, indem mehrere „R/S“-Worte mit zählendem Wert „0“ aneinandergehängt werden.

Differenzbilder

Die Ähnlichkeit zweier aufeinanderfolgender Bilder lässt sich darstellen, indem man die sogenannten „Differenzbilder“ erzeugt. Dabei werden die Zeilensignale eines Bildes von den entsprechenden Zeilensignalen des Vorbildes subtrahiert und der Betrag gebildet - entsprechende Geräte führen dies automatisch durch. Auf diese Weise entsteht Zeile für Zeile die Differenz bzw. die Veränderung zwischen zwei aufeinanderfolgenden Bildern.

Diese Differenzsignale können auf einem Bildschirm als Differenzbild dargestellt werden. Wenn sich die beiden Bilder exakt entsprechen (Standbilder), so wird das Differenzsignal an jeder Stelle gleich Null sein - das Differenzbild bleibt schwarz. Differenzpegel größer Null entstehen nur in jenen Bildpartien, in denen sich aufeinanderfolgende Bilder unterscheiden. Nur dort wird das Differenzbild helle Partien zeigen.

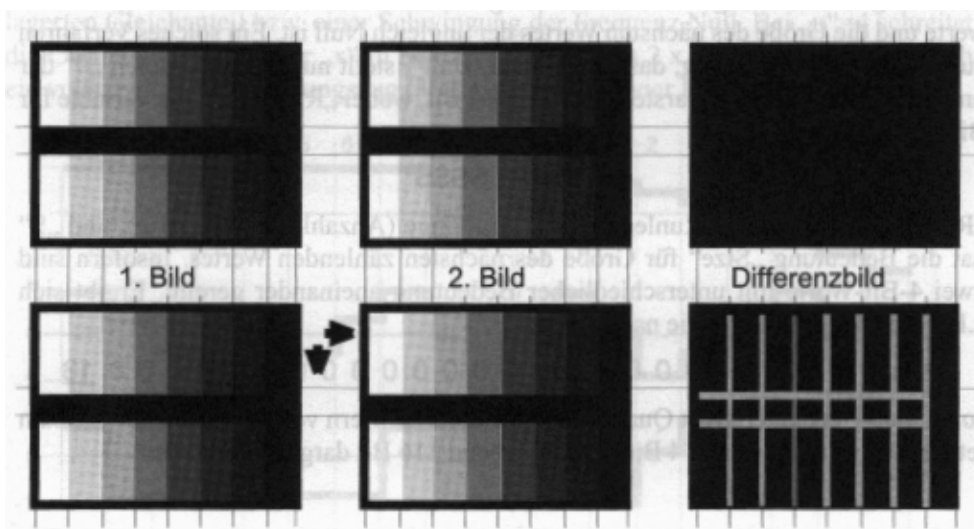


Bild: Differenzbilder bei identischstehendem bzw. identisch und vollständig bewegtem Folgebild

In der oberen Zeile entspricht das zweite Bild in Lage und Inhalt dem ersten. Das Differenzbild ist vollständig schwarz. In der unteren Zeile wurde das zweite Bild gegenüber

dem ersten entsprechend den angegebenen Pfeilen verschoben. Die Differenzen entstehen an den Kanten der horizontalen und vertikalen Balken und sind (vereinfacht) jeweils in gleicher Helligkeit dargestellt.

Auch normale Bildsequenzen weisen zwischen zwei aufeinanderfolgenden Bildern erhebliche Ähnlichkeiten auf, so dass Differenzbildsequenzen in aller Regel weniger Bildinformation als die entsprechenden Einzelbilder enthalten. Die oben dargestellte Bewegung, die das ganze Bild verschiebt, entspricht einer Bewegung der Kamera. Auch bei Bewegungen innerhalb des Bildes (s. Bild unten) zeigt das Differenzbild in der Regel viele schwarze Flächen.

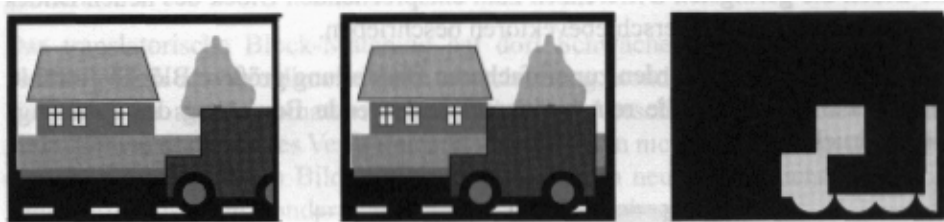


Bild: Differenzbild bei stehendem Hintergrund und bewegtem Bildinhalt im Vordergrund

Bewegungskompensation durch Block-Matching

Je mehr Bewegung bzw. Veränderung zwischen zwei Bildern vorkommt, desto weniger schwarze Flächen zeigt das Differenzbild und desto mehr Informationsgehalt steckt darin. Eine Reihe von Datenkompressionsverfahren basieren auf der Verarbeitung von Differenzbildern.

Das Ziel der Bewegungskompensation ist es, das jeweils vorangegangene Bild für die Differenzbildung so zu verändern, dass, trotz der Bewegung die im darauffolgenden Bild stattgefunden hat, das Differenzbild an möglichst vielen Stellen „Schwarz“ zeigt. Die Veränderungen werden als sogenannte „Bewegungsvektoren“ ausgedrückt und mitgespeichert bzw. übertragen. Es werden also zusätzliche Informationen gewonnen, mit deren Hilfe der Informationsgehalt und damit die Datenmenge der Differenzbilder reduziert werden kann.

Im Bild oben ist das gesamte Bild nach rechts und nach unten verschoben. Diese Bewegung lässt sich mit den beiden dargestellten Vektoren beschreiben und entspricht einer Bewegung der Kamera über einer Bildvorlage. Mit zwei Vektoren für das gesamte Bild können aber keine Veränderungen innerhalb des Bildes selbst beschrieben werden.

Je kleiner die bewegten Details sind, desto kleiner müssen auch die betrachteten Bildausschnitte sein, durch deren Verschiebung die Bewegung kompensiert wird. Im Grenzfall würde die Bewegung jedes einzelnen Bildpunktes ausgeglichen. Dies führte aber u.U. zu mehr zusätzlichen Informationen, als durch die Kompensation eingespart würde.

Die *Block-Matching-Technik* arbeitet in der Regel mit Blöcken von 8 x 8, 16 x 16 oder 32 x 32 Bildpunkten. Das Gesamtbild wird also in eine Anzahl kleinerer Bildausschnitte zerlegt. Für diese Bildausschnitte werden die Bewegungsvektoren ermittelt. Dabei wird jeweils ein Block innerhalb eines eingeschränkten Suchbereichs versuchsweise hin und her

verschoben. Für jede mögliche Position innerhalb des Suchbereichs werden Differenzblöcke zu den jeweiligen Blöcken des neuen Bildes gebildet. Innerhalb eines typischen Suchbereichs von beispielsweise ± 16 Bildpunkten horizontal und ± 8 Bildpunkten vertikal ergeben sich $(2 \times 16) \times (2 \times 8) = 512$ mögliche Differenzbildblöcke, die „abgearbeitet“, d.h. verschoben, subtrahiert und in ihrem Restwert verglichen werden müssen. Diejenige Position, in der der verschobene Block die geringsten Differenzen zum entsprechenden Block des neuen Bildes aufweist, wird durch die Verschiebevektoren beschrieben.

Für das untere Beispiel wurden zur einfacheren Darstellung größere Blöcke gewählt. In den Vektoren kommt die rein horizontal verlaufende Bewegung des Fahrzeugs zum Ausdruck.

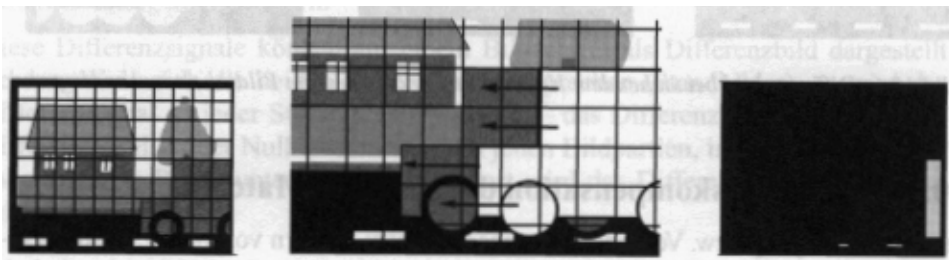


Bild: Differenzbild (s. Bild oben) nach Bewegungskompensation

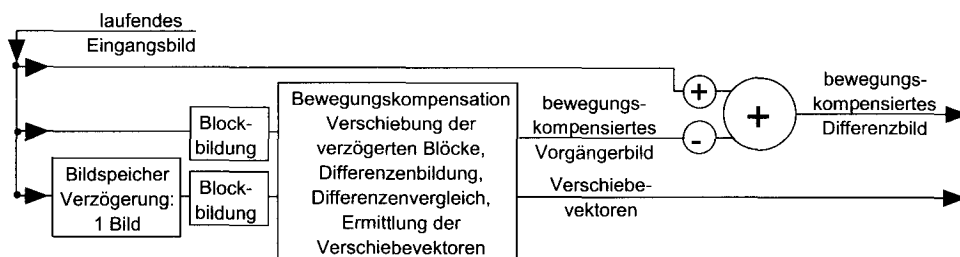


Bild: Blockschaltsbild Bewegungskompensation durch Block-Matching

Dieses Verfahren ist nicht geeignet, um bewegte Gegenstände als zusammengehörige Einheiten zu erkennen und tatsächlich deren Bewegung nachzuvollziehen. Wenn Bewegungen solche Strukturen betreffen, die über Blockgrenzen hinweg verlaufen, dann wird es in vielen Fällen nicht die *eine* richtige Lösung geben, sondern u.U. mehrere „ähnlich dunkle“ Blöcke. Die Bewegungskompensation ist im Prinzip jedoch schon dann erfolgreich, wenn das Differenzbild weniger helle Bereiche aufweist, gleichgültig ob das kompensierte - besser adaptierte - Vorgängerbild wirklich dem neuen Bild entspricht.

Das vorgestellte Verfahren verschiebt ausschließlich in horizontaler und vertikaler Richtung und wird deshalb auch „translatorisches Block-Matching“ bezeichnet. Der Rechenaufwand für solche Operationen ist im wahrsten Sinn des Wortes gigantisch. Ein CCIR 601-Luminanz-Bild von 720×576 Bildpunkten beinhaltet etwa 6500 Blöcke zu je 8×8 Bildpunkten. Rechnet man mit etwa 500 Suchpositionen je Suchbereich, sowie mit 25 Bildern je Sekunde, so ergeben sich mehr als 80 Millionen zu vergleichende Blöcke pro Sekunde. Derzeitige Rechnersysteme sind kaum in der Lage diese Berechnungen in Echtzeit vollständig durchzuführen. Aus diesem Grund wird das Suchraster zu den Grenzen des Suchbereichs hin oftmals vergrößert.

Das translatorische Block-Matching hat dort Schwächen, wo Bewegungen nicht ausschließlich als zweidimensionale Verschiebung, sondern z.B. als Rotation von Körpern oder als Kombination von Rotation und Verschiebung auftreten. Des weiteren können durch dieses Verfahren z.B. Zoomfahrten nicht richtig ausgewertet werden, da die betrachteten Bildausschnitte (Blöcke) im neuen Bild nicht mehr in der alten Form existieren, sondern ihre Größe verändert haben.

Rekonstruiert werden die Bilder durch Addition des alten Bildes mit dem Differenzbild. Durch die Bewegungskompensation stellt das Differenzbild aber nicht den Unterschied zwischen altem und neuem Bild, sondern den Unterschied zwischen dem *veränderten alten* und neuem Bild dar. Deshalb muss jedes rekonstruierte Bild für die Addition mit dem Differenzbild ebenfalls wieder - gemäß den übertragenen Vektoren verändert werden. Um den Prozess starten zu können, und um etwaige Übertragungsfehler nicht endlos fortzuschreiben, wird von Zeit zu Zeit ein vollständiges Bild übertragen.

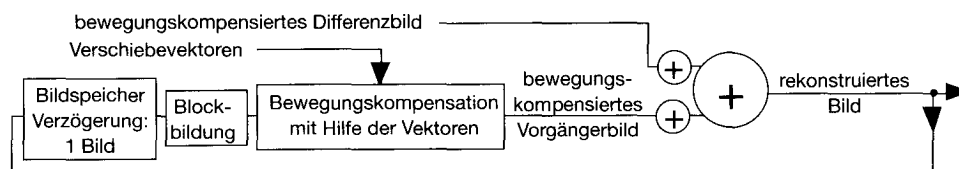


Bild: Rekonstruktion durch Addition von Differenzbild und bewegungskompensiertem Vorgängerbild

Differential Puls Code Modulation (DPCM)

Die beiden Verfahren, die heute am häufigsten eingesetzt werden, um eine günstige Verteilung der Auftrittswahrscheinlichkeiten der Bilddaten zu erreichen, sind die Differential Puls Code Modulation (DPCM) und die Transformationscodierung, hier besonders die Discrete Cosinus Transformation (DCT).

Bei der Differenzencodierung mit DPCM werden die Regelmäßigkeiten ausgenutzt, die sich über ausgedehnte Teile des Bildes erstrecken, indem nur die Differenzen zwischen Nachbarpunkten oder aufeinander folgenden Bildern übertragen werden. Gewöhnliche Bilder enthalten oft große Flächen, so dass die Differenz benachbarter Pixel nur gering ist. Nur die Kanten erzeugen große Differenzen und führen zu den wenigen hohen Werten. Die Differenzbildung kann so betrachtet werden, dass aus den übertragenen Bildpunkten eine Vorhersage (Prädikation) für den nächsten gebildet wird. Vom Signalwert wird die Prädikation subtrahiert, schließlich wird nur die Abweichung vom wirklichen Wert, der Prädikationsfehler, übertragen. Der Vorgang entspricht einer digitalen Filterung. Schon eine einfache Verzögerung eignet sich als Prädikationsfilter, und kann auch beim Integrierglied verwendet werden, das die Originalwerte wieder rekonstruiert.

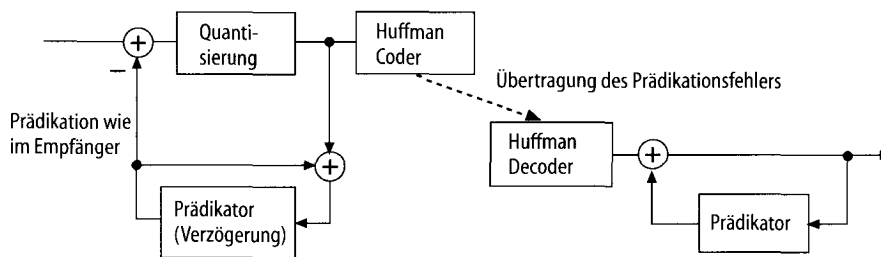


Bild: DPCM ohne Fehlerverschleppung

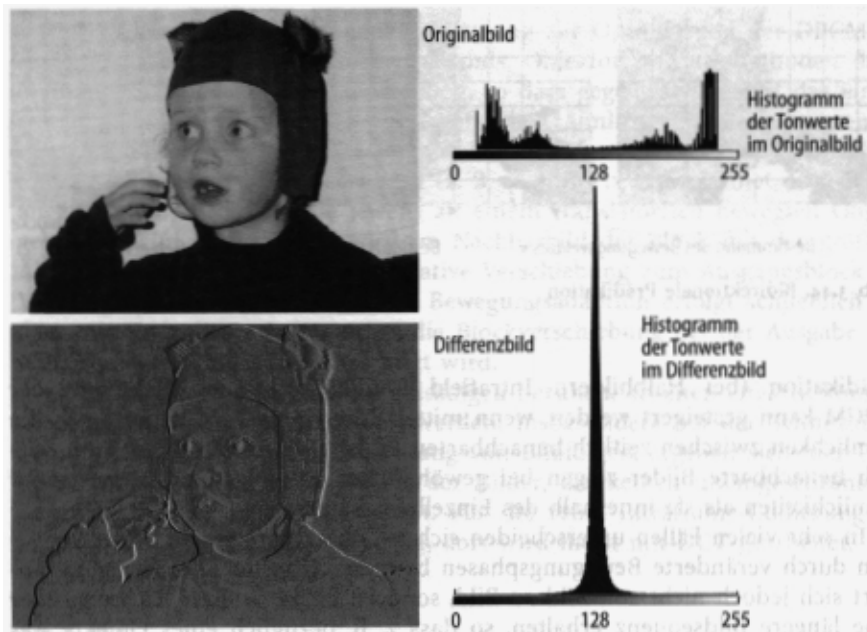


Bild: Original- und Differenzbild mit Grauwertverteilungen

Bei ungestörter Übertragung ist eine fehlerfreie Rekonstruktion möglich, es ergibt sich aber keine hohe Codiereffizienz. Um den Datenstrom weiter zu reduzieren, werden die Daten häufig quantisiert. Die dabei unvermeidlichen Quantisierungsfehler werden nun im Decoder aufaddiert und diese Fehlerverschleppung macht das Signal innerhalb kurzer Zeit untauglich. Die Fehlerfortpflanzung lässt sich aber vermeiden. Dazu wird bereits im Sender eine Decodierung vorgenommen, die genau der Decodierung im Empfänger entspricht. Das decodierte Signal wird wieder der Additionsstufe zugeführt, dass die Eingangsbedingungen am Decoder vorweggenommen sind

Als Beispiel für eine Intraframe-DPCM zeigt das Bild oben ein Graustufenbild und das zugehörige Differenzbild, sowie im Histogramm jeweils die zugehörigen Häufigkeitsverteilungen der Grauwerte. Zur Gewinnung des Differenzbildes wurde das Bild um ein Pixel horizontal verschoben und die Differenz zwischen verschobenem und unverschobenem Bild gebildet. Der Wert 128 erscheint als mittlerer Grauwert, positive Differenzen heller und negative dunkler. Mittlere Grauwerte überwiegen deutlich, auch im Histogramm wird wieder die Konzentration um den Wert 128 sichtbar. Die Statistik des Differenzbildes hat sich gegenüber dem Ausgangsbild verändert, der mittlere Grauwert tritt viel häufiger auf als die anderen Differenzwerte. Die Häufigkeitsverteilung ist damit so umgestaltet, dass die variable Längenzuordnung effektiv arbeiten kann.

Die bisher dargestellte Form der DPCM bezog sich allein auf Ähnlichkeiten zwischen den Nachbarpixeln in einem Bild, man spricht von Intraframe

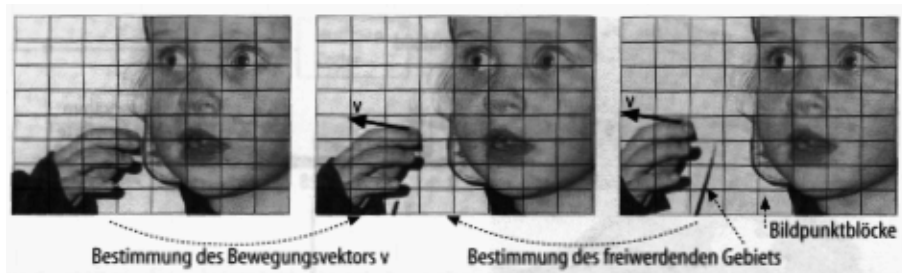


Bild: Bidirektionale Prädikation

Prädikation (bei Halbbildern Intrafield Prädikation). Die Effektivität der DPCM kann gesteigert werden, wenn mittels Interframe Prädikation auch die Ähnlichkeit zwischen zeitlich benachbarten Bildern ausgenutzt wird. Zwei zeitlich benachbarte Bilder zeigen bei gewöhnlichen Bildfolgen noch weit mehr Ähnlichkeiten als sie innerhalb des Einzelbildes auftreten.

In sehr vielen Fällen unterscheiden sich aufeinander folgende Bilder vor allem durch veränderte Bewegungsphasen bewegter Objekte. Die Bewegung ändert sich jedoch nicht von Bild zu Bild, sondern bleibt in ihrer Richtung über eine längere Bildsequenz erhalten, so dass z. B. bezüglich eines Objekts, das sich in einer Sekunde über zwei Drittel der Bildbreite bewegt hat, mit sehr großer Wahrscheinlichkeit abgeschätzt werden kann, an welcher Position es sich im nächsten Bild befinden wird. Mit dieser Abschätzung verschobener Bildteile aus dem Vergleich von aktuellem und vorhergehendem Bild wird dann eine Bewegungskompensation vorgenommen. Das aktuelle Bild wird der Bewegungsvorhersage entsprechend verändert und ist damit dem folgenden Bild wesentlich ähnlicher als ohne Bewegungskompensation. Die Abbildung oben zeigt ein Beispiel für ein Motiv mit bewegter Hand. Bei der Verfolgung der Bewegung wird deutlich, dass das Objekt zwar im, dem aktuellen Bild folgenden, Bild weitgehend richtig lokalisiert wird, aber bei der Bewegung an der Stelle wo aus es sich fortbewegt den Hintergrund verdeckt hat und damit in diesem Bereich die Prädikation versagt. Der Hintergrund wird erst im folgenden Bild freigegeben. Wenn diese Information für die Vorhersage des aktuellen Bildes benutzt werden könnte, wäre die Prädikation noch wesentlich verbessert. Eine optimale Prädikation ist also bidirektional, sie stützt sich auf das dem aktuellen vorhergehende und das nachfolgende Bild. Obwohl das folgende Bild in der Zukunft liegt und eigentlich gar nicht bekannt sein kann, gibt es die Möglichkeit zur bidirektionalen Prädikation durch zeitliche Verzögerung auf der Coderseite, d. h. das nächste Bild wird vor der Übertragung berücksichtigt. Anschließend kann die Bildfolge vor der Übertragung geändert werden, so dass zunächst beide Nachbarbilder des gewünschten Bildes am Decoder vorliegen, wo die Bilder durch Integration aus den Differenzen zurückgewonnen und anschließend wieder in die richtige Reihenfolge gebracht werden. Bilder die auf einfacher, unidirektionaler Prädikation beruhen, werden mit P gekennzeichnet, die aus bidirektionaler Prädikation mit B. Intraframe codierte Bilder ohne Prädikation werden als I-Frames bezeichnet.

Das Problem bei der Bewegungsschätzung zur Optimierung der DPCM ist die Erkennung des bewegten Gegenstands. Objektorientierte Methoden dazu befinden sich noch in der Entwicklung, so dass gegenwärtig meist das einfache Blockmatching-Verfahren zum Aufsuchen ähnlicher Bildteile verwendet wird. Dabei werden aus Gründen des Rechenaufwands nicht jeder einzelner Bildpunkt, sondern Bildpunktblöcke (z. B. 16 x 16 Pixel) betrachtet, von denen angenommen wird, dass sie jeweils zu einem translatorisch bewegten Objekt gehören. Beim Bildvergleich wird im Nachbarbild der Block mit der größten Ähnlichkeit gesucht und dessen relative Verschiebung zum Ausgangsblock als Vektor gespeichert. Die Bewegungsadaptation erfolgt schließlich so, dass mit Hilfe dieser

Information die Blockverschiebung bei der Ausgabe des vorhergesagten Bildes berücksichtigt wird.

Codierungsmethoden, die auf Bildfolgen beruhen, arbeiten effektiv, können allerdings nicht immer eingesetzt werden. Insbesondere bei der Schnittbearbeitung (Editing), also der Umstellung von Bildfolgen, ergeben sich hier Probleme aufgrund der Abhängigkeit der Bilder, die keinen uneingeschränkten Zugriff auf jedes Einzelbild erlaubt. Für die reine Intraframe-Codierung ist aber die DPCM nicht effektiv genug, dort wird meist mit DCT gearbeitet.

Descrete Cosinus Transformation (DCT)

Die Diskrete Cosinustransformation als Sonderform der allgemeinen Transformationscodierung beruht auf dem Umstand, dass sich, ähnlich wie bei einem fouriertransformierten Analogsignal, aus dem Frequenzspektrum andere Inhalte und Bearbeitungsmöglichkeiten erschließen als bei der Zeitdarstellung.

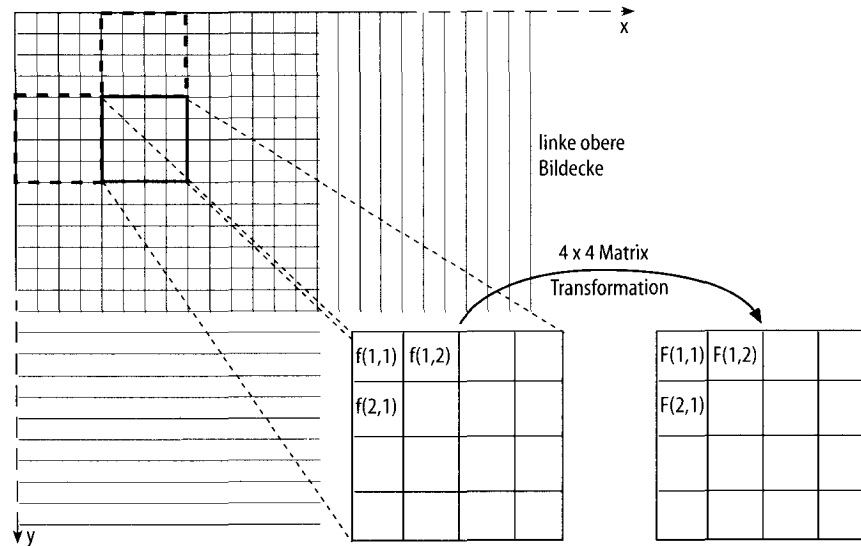


Bild: DCT-Blockbildung

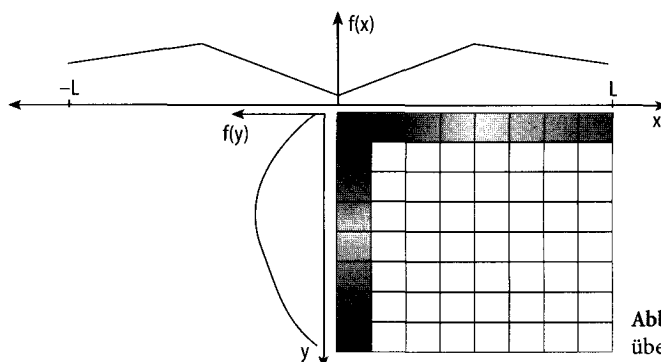


Abb. 3.46. Funktionsverlauf über den Bilddimensionen

Das Frequenzspektrum eines Videosignals kann dreidimensional dargestellt werden. Neben der Frequenz bezüglich der Zeit ergeben sich die Ortsfrequenzen bezüglich der Koordinaten der Bildfläche. Tiefe Ortsfrequenzen repräsentieren große Bildstrukturen und

langsame Helligkeitsübergänge, hohe Frequenzen werden dagegen durch kleine Strukturen und abrupte Übergänge verursacht. Da in natürlichen Bildern grobe Strukturen weit häufiger vorkommen als feine, findet eine Konzentration auf niederfrequente Komponenten statt. Falls hochfrequente Komponenten zur Datenreduktion ungenauer dargestellt oder ganz weggelassen werden, ist der Fehler oft irrelevant.

Zur DCT wird das Bild in Blöcke aufgeteilt, die meist quadratisch angeordnet sind, ein Block besteht dann aus $N \times N$ benachbarten Pixeln (Bild). Die Transformation wird jeweils auf alle Pixel eines Blockes angewendet. Bei zu kleinen Blöcken werden die Ähnlichkeiten benachbarter Pixel nicht gut ausgenutzt, bei zu großen Blöcken können die Inhalte verschiedener Blöcke ineinander übersprechen. Häufig wird eine Blockgröße mit $N = 8$ gewählt.

Die Transformation eines Blockes von Bildpunkten führt zu einem gleich großen Block von Transformationskoeffizienten. Die Transformation ist prinzipiell reversibel, ein $N \times N$ Block von Bildpunktswerten wird in $N \times N$ Koeffizienten überführt, die statt der Werte selbst übertragen werden. Die diskreten Pixelpositionen im Bild können mit x und y bezeichnet werden, die Positionen der Transformationswerte mit u und v . Die Werte der Bildpunkte sind dann durch $f(x,y)$ beschrieben, die Transformationswerte durch $F(u,v)$. Mit dieser Festlegung entsteht die DC-Transformation folgendermaßen aus der Fourier-Transformation:

Eine eindimensionale Funktion $f(x)$, die stetig über dem Intervall $-L$ bis $+L$ definiert ist, lässt sich als Fourierreihe mit den Summen über die mit den Fourierkoeffizienten a und b gewichteten \cos - und \sin -Funktionen darstellen, die als Argumente die Vielfachen der Grundfrequenz enthalten. Betrachtet man nun ein Intervall von 0 bis $+L$, wie es z. B. dem Funktionsverlauf über einer Videozeile entspricht, so lässt sich die Funktion über diesen Bereich auf das Intervall $-L$ bis $+L$ erweitern und es entsteht eine gerade Funktion, bei der alle zur \sin -Funktion gehörigen Fourierkoeffizienten zu Null werden und nur noch die \cos -Anteile übrigbleiben (Bild voriger Seite). Da die erweiterte Funktion über dem halben Intervall aber wieder der Ursprungsfunktion

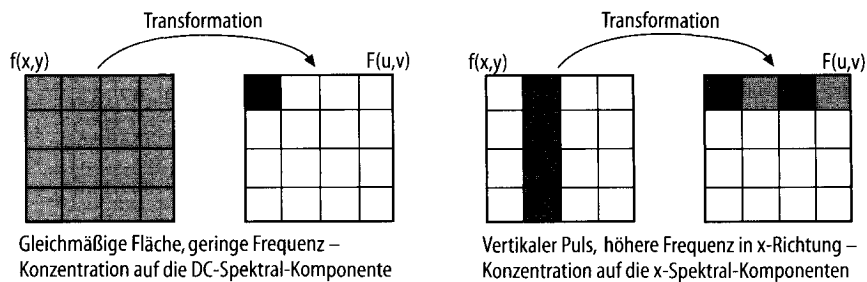


Bild: DCT-Beispiele

entspricht, bleibt es beim Wegfall der \sin -Anteile und es gilt:

$$f(x) = \sum_u a_u \cos (u\pi x/L).$$

Für die Transformation aller Werte, die auf einer zweidimensionalen Fläche vorliegen (z. B. beim Bildwandler) kann zunächst von der Multiplikation zweier Funktion ausgegangen werden, die von x und y abhängen :

$$f(x,y) = f(x) f(y).$$

Damit folgt die zweifache Reihenentwicklung

$$f(x,y) = \sum_u \sum_v c_{uv} \cos(u\pi x/L_1) \cos(v\pi y/L_2).$$

Mit dem Übergang von einer kontinuierlichen auf eine diskrete Intensitätsverteilung, entsprechend der Werte einzelner Bildpunkte innerhalb eines Blockes der Größe $N \times N$ folgen schließlich die Beziehungen für die DCT:

$$f(x,y) = 2/N \sum_u \sum_v C(u) C(v) F(u,v) \cos((2x+1)u\pi/2N) \cos((2y+1)v\pi/2N)$$

Für die Rücktransformation gilt:

$$F(u,v) = 2 C(u) C(v)/N \sum_x \sum_y f(x,y) \cos((2x+1)u\pi/2N) \cos((2y+1)v\pi/2N)$$

mit der Summe von 0 bis $N-1$ und $C(w) = 1/\sqrt{2}$ für $w = 0$ und $C(w) = 1$ sonst. Es wird deutlich, dass in die Berechnung eines einzelnen Transformationswertes alle Werte des zu transformierenden Blockes einfließen. Die DC-Komponente für den mittleren Grauwert des Blocks ist von großer Bedeutung, sie befindet sich in der linken oberen Ecke der DCT-Matrix. Wenn für jeden Block nur dieser Wert übertragen würde, ergäbe sich bereits ein Bild, das jedoch eine horizontale und vertikale Auflösungsverminderung um den Faktor N aufweisen würde. Die Koeffizienten für die niederfrequenten AC-Spektralanteile befinden sich in direkter Nachbarschaft darunter bzw. rechts vom DC-Koeffizienten. Die hochfrequenten AC-Anteile sind unbedeutender. Sie liegen bei der unteren rechten Ecke der Matrix und weisen i. d. R. wesentlich kleinere Werte auf als die Koeffizienten in der Nähe der DC-Komponente, zudem sind zugehörigen hohen Ortsfrequenzen visuell weniger gut wahrnehmbar.

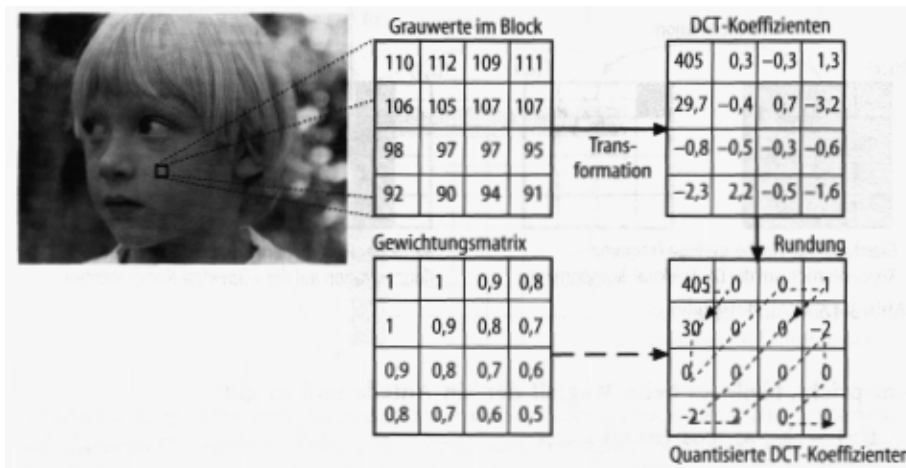


Bild: DCT mit Gewichtung und Rundung

Mit der Transformation wird eine Konzentration der Signalleistung auf wenige Teilbänder des Frequenzspektrums vorgenommen. Das Bild ganz oben zeigt zwei Beispiele für einfache Transformationen. Bei einer gleichmäßig hellen Fläche ergibt sich nach der Transformation nur der DC-Koeffizient, während ein vertikaler Impuls einer Sprungfunktion in der Horizontalen entspricht, woraus entsprechende Koeffizienten folgen. Das Bild oben zeigt die Blockbildung bei einem Graustufenbild, die Grauwerte des ausgewählten 4×4 -

Blocks sind als Zahlen dargestellt. Nach einer DCT ergibt sich der in der Abbildung (vorherige Seite) dargestellte Block von Koeffizienten. Es wird deutlich, dass der Gleichanteil den größten Wert aufweist und sich hier die Zahlen erheblich mehr voneinander unterscheiden als im Bildblock. Damit sind die Voraussetzungen für eine effektive variable Längencodierung geschaffen.

Quantisierung

Die bisher beschriebene Form der DCT ist hinsichtlich der Datenreduktion nicht sehr effektiv, dafür findet eine verlustlose Reduktion statt, die DCT ist vollständig reversibel. Zunächst sind für die Darstellung der DCT-Koeffizienten sogar mehr als 8 Bit erforderlich. Bereits der DC-Wert 405 aus dem einfachen Beispiel im Bild oben ist nicht mehr mit 8 Bit darstellbar. Ausgehend von einem Block mit 8 x 8 Bildpunkten die mit je 8 Bit codiert sind, ergibt sich durch die Summierung über die 64 Werte, die zur Gewinnung jedes Koeffizienten erforderlich ist, eine Erhöhung um 6 Bit. Aufgrund der Normierung mit dem Divisor 8 bzw. 4 in der DCT-Bestimmungsformel reduzieren sich diese wiederum, so dass alle Koeffizienten mit 11 Bit dargestellt werden könnten. Darüber hinaus weisen die Mehrzahl der AC-Koeffizienten kleine Werte auf und der Zahlbereich kann weiter eingeschränkt werden.

Um die Daten weiter zu reduzieren, werden die Koeffizienten quantisiert und dabei unterschiedlich bewertet. Die hochfrequenten Signalanteile werden mit geringerer Auflösung quantisiert als die niederfrequenten, oder sie werden ganz weggelassen, denn das menschliche Auge nimmt die zu den hochfrequenten Anteilen gehörigen feinen Bildstrukturen weniger gut wahr als die groben. Das Bild unten zeigt zwei anhand eines psycho-physiologischen Experiments optimierte Quantisierungstabellen für Luminanz und Chrominanz in denen für jeden Koeffizienten eines 8 x 8-DCT-Blocks ein Divisor aufgeführt ist. Es wird deutlich, dass die zu hochfrequenten Bildinhalten gehörigen Koeffizienten ein geringeres Gewicht bekommen. Der DC-Divisor der Tabelle ist 16, so dass der DC-Koeffizient, der gewöhnlich den größten Wert aufweist, bei dieser Bewertung mit 7 Bit statt 11Bit dargestellt werden kann. Die Teilung der höchsten Koeffizientenwerte durch ca. 100 reduziert die erforderliche Bitzahl auf 5.

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	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Quantisierungstabelle für die 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

Quantisierungstabelle für die Chrominanz

Bild: Quantisierungstabellen

Nachdem die Werte auf ganze Zahlen gerundet wurden sind nur noch sehr wenige Koeffizienten von Null verschieden. Dieser Umstand wird besonders deutlich, wenn die Original-Abtastwerte nur sehr wenige Grauwertabstufungen umfassen. Fein strukturierte Bildblöcke können eine größere Zahl von Null verschiedener Koeffizienten enthalten. Daran wird deutlich, dass die Datenrate bei der Übertragung in Abhängigkeit vom Bildinhalt schwankt. Wenn die DCT in Systemen eingesetzt wird, die diese Schwankung

nicht verarbeiten kann, wie z. B. Bandaufzeichnungsverfahren, so muss der Datenstrom geglättet werden.

Im Bild unten ist die Rücktransformation der Koeffizientenmatrix aus dem einfachen Beispiel oben dargestellt. Zu Beginn wird invers gewichtet, aus dem quantisierten Koeffizientenfeld ergibt sich danach durch Rücktransformation das decodierte Bildsegment. Schließlich wird deutlich, dass sich nur wenige der zurückgewonnenen Bildpunktwerte von den Ausgangsbildpunkten unterscheiden.

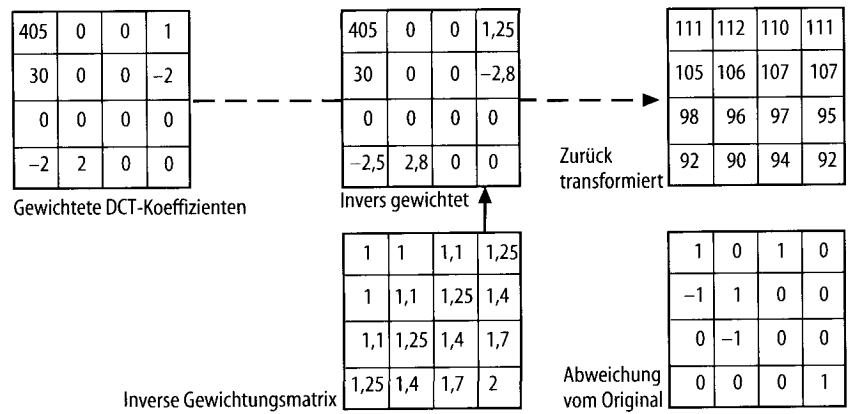


Bild: DCT-Rücktransformation

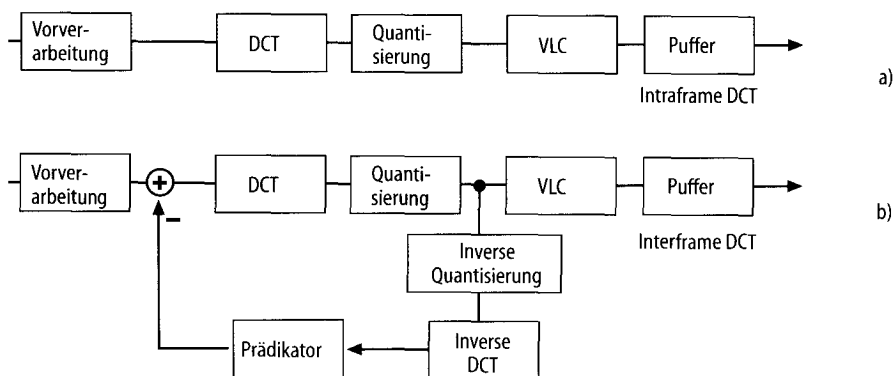


Bild: Vergleich von Intraframe-DCT (a) und Hybrid-DCT (b)

Aber auch wenn die Fehler klein sind, so wird das Signal doch durch den Quantisierungsprozess verfälscht. Ein typischer Fehler, der bei der DCT auftritt, ist der Blocking-Effekt, d. h. dass die Blockgrenzen sichtbar werden. Es hängt von der Wahl der Bewertungs-Matrizen ab, in welchem Maße die Fehler in Erscheinung treten. Die Wahl der Bewertungs-Matrizen ist der wesentliche Unterschied zwischen verschiedenen Herstellern, die DCT in ihren Geräten implementieren. Obwohl alle Hersteller ein Verfahren mit der gleichen Bezeichnung einsetzen, können Datenreduktionsfehler bei einem Gerät deutlicher in Erscheinung treten als bei anderen. Die Bewertungsmatrizen werden meistens nicht veröffentlicht.

Hybride DCT

Eine weitere Erhöhung der Effektivität der Datenreduktion ergibt sich, wenn die Nachbarschaftsbeziehungen zwischen den Pixelwerten nicht nur innerhalb eines Bildes (Intraframe), sondern auch zwischen verschiedenen Bildern einer Bildfolge (Interframe) ausgenutzt werden. Die Erweiterung der DCT unter Einbeziehung der zeitlichen Dimension ist sehr rechen- und speicheraufwendig, deswegen wird in der Praxis meist eine Kombination aus DPCM und DCT, die sog. hybride DCT, angewendet (Bild oben), bei der die DCT die Ähnlichkeitsbeziehungen innerhalb eines Bildes und die DPCM die Ähnlichkeiten zwischen benachbarten Bildern ausnutzt.

Bei der hybriden DCT wird zunächst die Differenz zweier aufeinander folgender Bilder ermittelt und anschließend wird das Differenzbild transformationscodiert. Die hybride DCT bewirkt eine sehr effektive Datenreduktion. Bilder in PAL-Qualität können mit einer Datenrate von ca. 5 Mbit/s übertragen werden. Bei akzeptabler Qualität sind Kompressionsfaktoren bis zu 100, bei geringen Qualitätsansprüchen auch Faktoren über 500 erreichbar, Videobilder geringerer Auflösung können dann mit 64 kbit/s übertragen werden. Die hybride DCT ist die Basis der Video-Quellencodierung bei MPEG und damit auch die Basis für das digitale Fernsehen, Digital Video Broadcasting (DVB).

Tabelle: Datenreduktionsverfahren mit DCT

Standard	Bits/Pixel	Blockbildung	Bewegungskompensat.	DPCM	Ratecontrol
CMTT-2	8	8 x 8 Intrafield	Halbbild (vorwärts)	ja	ja
Digital Betacam	10	8 x 4 Intrafield	nein	nein	Vorwärtsschätzung
DV	8	8 x 8 oder 8 x 4	nein	nein	Vorwärtsschätzung
M-JPEG	8	8 x 8	nein	nein	nicht festgelegt
MPEG-1	8	8 x 8 intraframe	ja (bidirektional)	ja	ja
MPEG-2	8	8 x 8 oder 8 x 4	ja (bidirektional)	ja	ja

Dort, wo das Videosignal bearbeitet wird, muss der Zugriff auf jedes einzelne Bild gewährleistet sein, die Interframe-Codierung ist hier nicht geeignet. Falls die Bilder aber nur noch übertragen und nicht mehr bearbeitet werden sollen, ist die hybride Codierung gut einsetzbar. Beim MPEG-Standard werden als Kompromiss die meisten Bilder hybrid codiert, in festen Abständen werden aber einige unabhängige, Intraframe-codierte Bilder in den Datenstrom eingefügt. Damit wird statt des Zugriffs auf Einzelbildern der Zugriff auf Bildgruppen möglich.

Die DCT ist heute die Bildcodierungsvariante mit der größten Bedeutung, sie wird bei MPEG und JPEG (s. u.) ebenso eingesetzt wie bei den speziellen Datenreduktionsverfahren für digitale Videorecorder (z. B. Digital Betacam oder DV). Die Tabelle oben zeigt eine Übersicht über verschiedene Datenreduktionsstandards die mit DCT arbeiten, die einzelnen Parameter werden in den folgenden Kapiteln erläutert. Neben DCT und DPCM sind weitere Codierungsmethoden verfügbar, z. B. die der DCT verwandte Discrete Wavelet Transformation (DWT). Bei der DWT wird das Bild mehrstufig räumlich herunterskaliert, und der dabei entstehende Fehler wird weiterverarbeitet. Der Vorteil ist hier, dass keine Blockbildung wie bei DCT vorgenommen werden muss, und ein mit DWT komprimiertes Bild dementsprechend keine Blocking-Artefakte aufweist. Der Nachteil ist eine gegenüber DCT schlechtere Wiedergabe feiner Strukturen. Ein weiteres Beispiel ist die theoretisch der DCT überlegene Vektorquantisierung, bei der ermittelt wird, welche Grundmuster von Pixelanordnungen im Mittel wie häufig auftreten.

Für die Grundmuster wird ein aus Codevektoren bestehendes sog. Codebuch festgelegt und übertragen. Die neueste Generation von Codern versucht eine Bildsequenz als Abbildung von bewegten und unbewegten Objekten zu verstehen. Wenn die Objekte einmal identifiziert sind, können sie mit sehr wenigen Informationsbits beschrieben werden.

JPEG (Joint Photographics Expert Group)

Die Abkürzungen JPEG und MPEG stehen für eine anwendungsunabhängige Standardisierungen bezüglich der Datenreduktion. Zur Definition wurden Expertengruppen gebildet, die Joint Photographics Expert Group (JPEG) und die Moving Pictures Expert Group (MPEG), die den Standards die Namen gaben.

JPEG beschreibt einen universellen Algorithmus zur Einzelbildcodierung und wurde ohne besondere Rücksicht auf Videosignale entwickelt. Das Verfahren beruht auf der Intraframe-DCT. Zunächst wird das Bild in Blöcke von 8 x 8 Bildpunkten zerlegt, die dann mit DCT in den Frequenzbereich transformiert werden. Als nächstes folgt die Gewichtung der DCT-Koeffizienten nach psycho-visuellen Gesichtspunkten. Als Besonderheit bei JPEG werden die DC-Koeffizienten einer DPCM unterzogen, d. h. ihre Werte werden verringert indem nur die Differenz zu den DC-Werten der Nachbarblöcke codiert wird. Über die Quantisierung und Rundung wird der Reduktionsfaktor festgelegt, wobei viele der hohen Ortsfrequenzen anschließend den Wert Null aufweisen. Die endgültige Datenreduktion wird mit der sich anschließenden VLC und RLC erreicht (Bild nächste Seite). Damit diese effektiv arbeiten, werden die Werte der zweidimensionalen Matrix in eine eindimensionale Folge gebracht. Dies geschieht mit Hilfe des sog. Zick-Zack-Scanning, das die Abtastung bei dem großen DC-Koeffizienten beginnt und die AC-Koeffizienten von tiefen zu hohen Ortsfrequenzen mit ihren immer kleineren Werten hin anordnet. Auf diese Weise wird erreicht, dass zum Ende der Kette sehr viele Nullen auftauchen, die mit der RLC sehr effektiv zusammengefasst werden können (Bild unten). An dem Punkt, ab dem bis zum Blockende nur noch Nullen auftreten, wird das Zeichen End of Block (EOB) übertragen. Der Decoder kann die restlichen Zeichen durch Nullkoeffizienten auffüllen, da die Blockgröße ja festliegt .

Schließlich wird die Huffman-Codierung vorgenommen, um einen optimalen, den Auftrittswahrscheinlichkeiten der Koeffizienten angepassten Code zu finden. Statt den Code immer neu zu berechnen, können auch Standard-Codes zum Einsatz kommen, die z. B. für Videobilder optimiert sind.

JPEG steht für eine Vorschrift zur Einzelbildcodierung, der Rechenprozess kann als Software auf Standardrechnern implementiert werden und braucht im Prinzip nicht in Echtzeit abzulaufen. Wenn die Datenreduktion nach JPEG aber für Videobilder eingesetzt werden soll, was vor allem bei nichtlinearen Schnittsystemen der Fall ist, so müssen die Bilder nacheinander in Echtzeit komprimiert werden, was als Motion JPEG (M-JPEG) bezeichnet wird.

Das mit JPEG definierte Dateiformat sieht vor, dass die Unterabtastung der

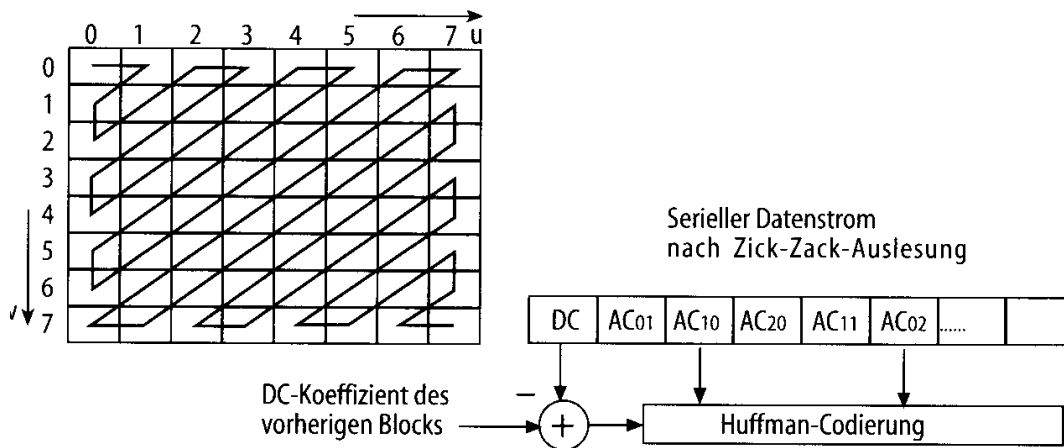


Bild: Serialisierung der Koeffizienten durch Zick-Zack-Auslesung

Farbdifferenzkomponenten gegenüber den Luminanz-Werten horizontal und vertikal gleich ist und die Chrominanz-Abtastwerte mittig zwischen den Luminanzwerten liegen. Dieses Format widerspricht den Empfehlungen in ITU-R 601, was zur Folge hat, dass bei Videoanwendungen meist den 601-Empfehlungen beachtet und das JPEG-Dateiformat verletzt wird.

Der DV-Algorithmus

Dieser Algorithmus ist als Datenreduktionsverfahren für das DV-Magnetbandaufzeichnungsverfahren entwickelt worden. Da sich das DV-Format im Laufe der Zeit im Produktionsbereich etabliert hat, hat der DV-Algorithmus inzwischen als wichtiges Datenreduktionsverfahren neben MPEG übergeordnete Bedeutung erhalten. Wie JPEG beruht DV auf einer Intraframe-Codierung ohne sich auf Nachbarbilder zu beziehen. Die Grundlage ist bei beiden Verfahren die DCT, mit Blöcken von 8 x 8 Bildpunkten. Im Gegensatz zu JPEG ist DV aber für Videoanwendungen optimiert und berücksichtigt Halbbilder ebenso wie die Abtaststruktur von ITU-R 601.

DV arbeitet mit einer 8-Bit-Auflösung. Das Verfahren ist im Gegensatz zu MPEG symmetrisch, d. h. Encoder und Decoder haben etwa den gleichen Funktionsumfang. Aus einem Eingangssignal nach ITU-R 601 wird zunächst mit Hilfe eines Vertical Chroma Filter ein nach 4:2:0 (PAL) oder 4:1:1 (NTSC) unterabtastetes Signal gewonnen. Für DV in Europa erzeugt die Y-Abtastung 720 X 576 und die C-Abtastung 360 X 288 Bildpunkte. Die Aufteilung in Blöcke mit 8 x 8 Pixeln erzeugt 90 x 72 Luminanz- und 45 x 36 Chrominanzblöcke.

Zur effektiven Verarbeitung der Halbbilder wird der Voll- und der Halbbildmodus unterschieden. Der Vollbildmodus verwendet die ineinander geschobenen Halbbilder bei der Blockbildung, während im Halbbild-Modus der 8 x 8 Block in zwei 4 x 8-Blöcke aufgeteilt werden, die jeweils nur die Information aus einem Halbbild enthalten. Da sich bei ruhenden Bildbereichen die Halbbilder wenig unterscheiden, ist die Korrelation örtlich benachbarter Zeilen groß und es wird der Vollbildmodus verwendet, während bei Blöcken mit schnell bewegten Inhalten auf den Halbbildmodus umgeschaltet wird, da sich die größten Ähnlichkeiten innerhalb des Halbbildes ergeben (Bild oben nächste Seite). Die Umschaltung wird von einem Bewegungsdetektor gesteuert und für die einzelnen Bildbereiche separat vorgenommen, so dass auch Bildvorlagen optimiert werden, die

ruhende und bewegte Teile enthalten.

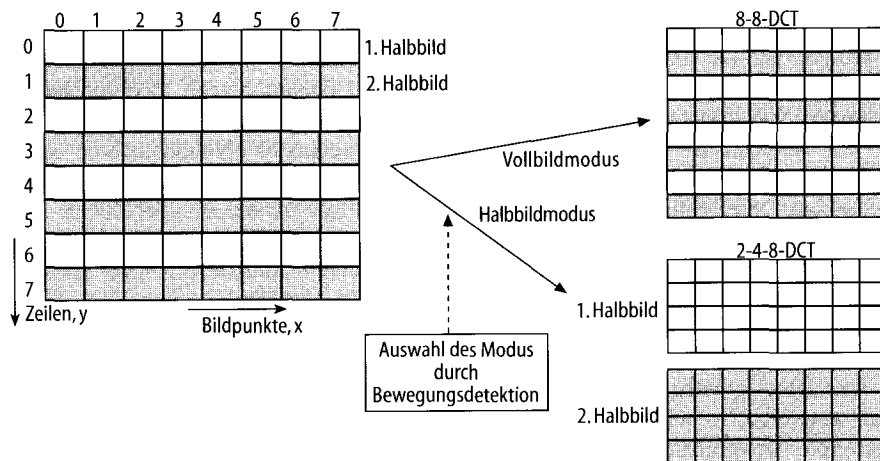


Bild: Halb- und Vollbildverarbeitung bei DV

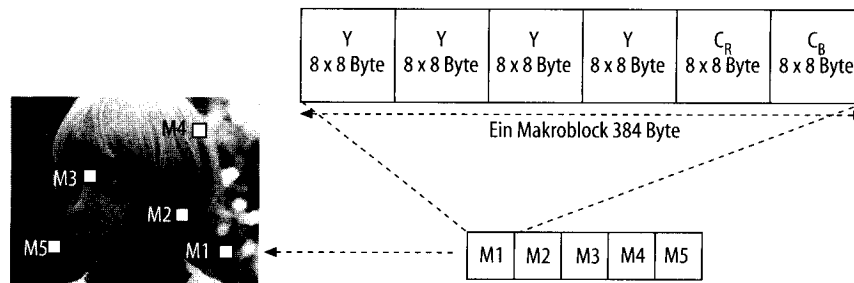


Bild: Intraframe Makroblock-Shuffling bei DV

Um die Y- und C-Verarbeitung zu vereinfachen, werden je 4 Luminanzblöcke und je ein CR- und GB-Block zu einem Makroblock zusammengefasst. Die 45 x 36 Makroblöcke werden schließlich 27 Superblöcken zugeordnet, von denen ein Bild 5 x 12 enthält. Im sog. Intra-Frame-Shuffling-Prozess werden dann die Superblöcke systematisch verwürfelt, so dass die benachbarten anschließend aus weit entfernten Bildbereichen stammen (Bild oben). Damit wird schon vor der Datenreduktion eine Angleichung der durchschnittlichen Redundanz erzielt und eine effektive Weiterverarbeitung gewährleistet.

DV erzielt die eigentliche Datenreduktion wie die meisten Verfahren mit DCT und anschließender VLC und RLC, wobei die Datenrate vom Bildinhalt abhängig ist. Da der DV-Algorithmus für die Anwendung bei der Bandaufzeichnung konzipiert ist, wurde besonderer Wert darauf gelegt, dass die dafür erforderliche konstante Datenrate möglichst effektiv erzielt wird. Während MPEG die konstante Datenrate quasi rückwärtsgerichtet erreicht, indem nach Ablauf der Datenreduktion anhand eines zu voll laufenden Pufferspeichers die folgenden Daten stärker reduziert werden, wird bei DV zu diesem Zweck mit der sog. Feed-Forward-Steuerung gearbeitet. Bei diesem Verfahren wird vor der eigentlichen Datenreduktion mit Hilfe einer Vorausberechnung anhand vieler verschiedener Quantisierungstabellen bestimmt, welche von ihnen die optimale ist, d. h. eine Datenrate erzeugt, die am dichtesten an dem durch das Aufzeichnungsverfahren gesetzten Limit liegt.

Der DV-Algorithmus erzielt auf diese Weise sehr gute Ergebnisse, bezieht sich aber ebenso wie JPEG nur das Bild und ist damit nicht wie MPEG für komplette

Programmbeiträge optimiert, wo neben der Video- auch die Audiocodierung und die Synchronisation erfasst ist.

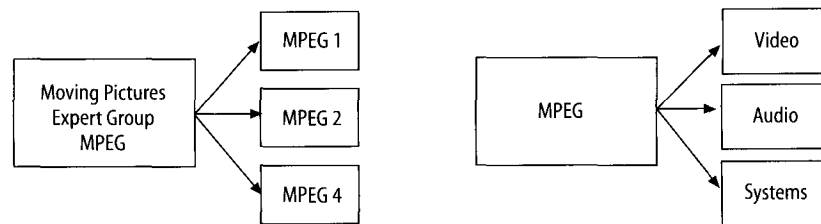


Bild: MPEG-Struktur

MPEG (Moving Picture Experts Group)

MPEG steht für einen universellen Standard zum Austausch von Bewegtbildern unter Einbeziehung von Audio- und Zusatzdaten. Die Entwicklung zielte auf eine möglichst umfassende Anwendbarkeit zur Datenübertragung, Speicherung und zum Einsatz in Multimediasystemen. MPEG sorgt vor allem dafür, dass das Format von allen Empfängern verstanden wird, die MPEG unterstützen. Definitionen zu diesem Punkt befinden sich im MPEG-Teil »Systems«. Zwei weitere Teile, nämlich »Video« und »Audio« beschreiben im wesentlichen mögliche Datenreduktionsarten (Bild oben). MPEG definiert die Codiervorgänge nicht bis in alle Einzelheiten, sondern stellt einen Rahmen dar, den verschiedene Entwickler unterschiedlich füllen können. Die Entwickler können beispielsweise jeweils ihre eigene Art der Quantisierung festlegen, sie müssen dann allerdings dafür sorgen, dass die Quantisierungskennlinie im Datenstrom mit übertragen wird. Die MPEG-Standards sind in mehreren Stufen festgelegt worden. MPEG-1 wurde zunächst im Hinblick auf kleine Datenraten bis 1,5 Mbit/s entwickelt, unter Inkaufnahme einer relativ schlechten Bildauflösung und -qualität. Die Definition von MPEG-2 berücksichtigt auch die Eigenschaften von Videosignalen und lässt Datenraten bei Standardauflösung bis zu 15 Mbit/s zu, womit eine sehr gute Bildqualität erreicht wird. Mit MPEG-3 sollten anschließend die Definitionen für hochauflösendes Fernsehen HDTV festgelegt werden. Diese gingen im Laufe der Entwicklung aber bereits in MPEG-2 auf, so dass MPEG-3 nicht existiert. Wenn im Zuge der Audio-Datenreduktion von einer MP3-Codierung gesprochen wird, so ist hier der Layer 3 der MPEG-2-Audiocodierung gemeint. Gegenwärtig befindet sich MPEG-4 in der Entwicklung, das mit Hilfe der Beschreibung von Objekten eine Datenrate im Bereich kbit/s für die Bewegtbildübertragung erreichen soll. Alle MPEG-Definitionen sind abwärtskompatibel ausgelegt, ein zukünftiger MPEG-4-Decoder soll also in der Lage sein, auch einen MPEG-2-Datenstrom zu decodieren.

MPEG-Videocodierung

Die MPEG-Videocodierung beruht auf der hybriden DCT, bei der im Vergleich zu JPEG und DV eine höhere Datenreduktionseffizienz erreicht wird, da sowohl die örtliche Korrelation (Wechselbeziehung) zwischen den Bildpunkten durch die DCT ausgenutzt wird, als auch die zeitliche Ähnlichkeit der Bilder in der Zeitdimension durch die DPCM.

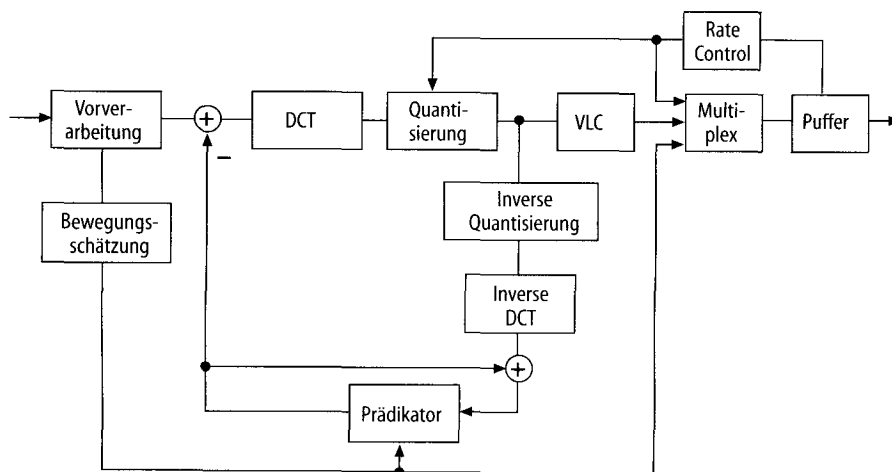


Bild: Blockschaltbild des MPEG-Video-Encoders

Das Bild zeigt das Blockschaltbild des MPEG-Video-Encoders. Wie bei JPEG ist im Kern eine DCT mit folgender Quantisierungsstufe enthalten, an die sich die Redundanzreduktionsstufe mit VLC und RLC anschließt. Damit die zur Quantisierung benutzten Tabellen nicht immer im Datenstrom zum Decoder übertragen werden müssen, sind für MPEG die im Bild auf der nächste Seite dargestellten Standardtabellen festgelegt, es besteht aber kein Zwang sie zu benutzen.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Quantisierungstabelle für die Luminanz

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

Quantisierungstabelle für die Chrominanz

Bild: Standard-Quantisierungstabellen bei MPEG

Nach Durchlaufen der Quantisierungsstufe wird das Signal verzweigt und mit reziproker Gewichtung und inverser DCT werden die Bildpunktwerte zurückgewonnen. Im Encoder wird aus dieser Information ein Prädikationsbild (Vorhersagebild) ermittelt woraus, zusammen mit dem nächsten Bild am Codereingang, die DPCM-Differenz gebildet wird. Wie bereits im Abschnitt über DPCM erläutert, wird die Prädikation optimiert, wenn eine Bewegungskompensation berücksichtigt wird. MPEG unterstützt diesbezüglich auch die dort beschriebene bidirektionale Prädikation, die auf den Nachbarbildern vor und hinter dem aktuellen Bild beruhen. Hierdurch steigt die Datenreduktionseffizienz weiter, da ein bidirektional prädizierter Block mit etwa der Hälfte eines unidirektional prädizierten codiert werden kann.

Die Informationen zur Bewegungskompensation stammen aus einer Verarbeitungsstufe zur Bewegungsschätzung, die meist mit dem Blockmatching-Verfahren arbeitet. Das Verfahren selbst ist aber bei MPEG nicht festgelegt. Die Informationen gelangen in Form eines Bewegungsvektors nicht nur zum Prädiktor (Bildverzögerer), sondern auch an die Daten-Multiplexstufe am Ausgang, da sie auch für den Decoder zur Verfügung stehen müssen und daher mit im Datenstrom übertragen werden. Um nicht für Luminanz- und Chrominanzblöcke separate Bewegungsvektoren ermitteln zu müssen, wird beim 4:2:0-Format aus vier Luminanzblöcken und je einem für CR und CB ein Makroblock gebildet, der die Basis für die Bewegungsschätzung bildet. Für das Format 4:2:2, wird die Anzahl

der Chrominanzblöcke verdoppelt (Bild unten).

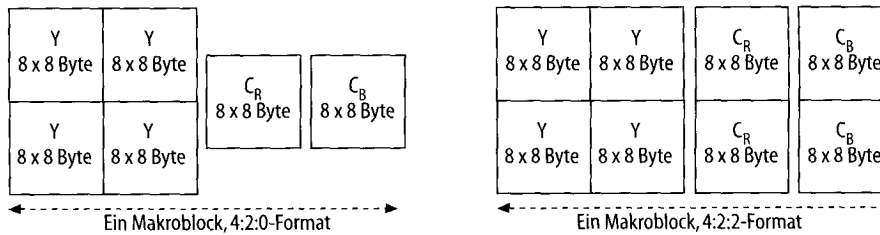


Bild: Makroblock-Strukturen bei MPEG

Neben den Ausgangsdaten und den Bewegungsvektoren wird eine Information über die Steuerung des Quantisierungsfaktors über die Multiplexstufe am Ausgang in den Datenstrom integriert. Das Multiplexverfahren bestimmt die Strukturierung der Daten. Es ist in der Lage nicht nur mit festen sondern auch mit flexiblen Datenraten zu arbeiten. Die Steuerdaten werden aus einer Pufferstufe am Encoder-Ausgang gewonnen, die die Aufgabe hat, den bildinhaltsabhängig variierenden Datenstrom zu glätten. Falls nun eine lange Folge komplexer Bilder codiert werden muss, wird der Datenstrom so hoch, dass der Pufferspeicher überzulaufen droht. Um diesen Zustand zu verhindern wird die Quantisierungsstufe vom Pufferfüllstand gesteuert. Dabei werden für jeden Block alle Einträge in der Quantisierungstabelle mit Ausnahme des Wertes für die DC-Komponente bei intraframe-codierten Bildern mit einem gemeinsamen Faktor variiert. Zur Senkung der Datenrate werden also die Werte in der Tabelle erhöht, so dass die Division der Koeffizienten durch diese Werte eine vergrößerte Quantisierung bewirkt. Falls auch die Teilung durch den Wert 31 nicht ausreicht, werden die Makroblöcke übersprungen.

MPEG-Decoder

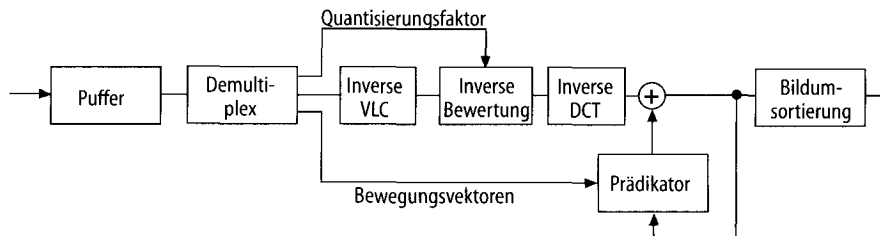


Bild: Blockschaltbild des MPEG-Video-Decoders

Das Bild oben zeigt das Blockschaltbild des MPEG-Videodecoders. Auch hier werden die Daten zunächst in einen Pufferspeicher übernommen. Im Demultiplexer werden dann die Bewegungsvektoren und die Quantisierungsinformationen abgetrennt und dem Prädiktor, bzw. der Verarbeitungsstufe zur inversen Quantisierung zugeführt. Nachdem die Koeffizienten der DCT zurücktransformiert sind und die Vorhersagewerte zur Invertierung der DPCM aufaddiert wurden, steht die Bildinformation wieder zur Verfügung. Der Prozess der Decodierung ist damit erheblich einfacher als der der Encodierung, vor allem weil der Rechenaufwand zur Bewegungsschätzung entfällt. MPEG stellt damit ein unsymmetrisches Codierungsverfahren dar, mit dem Vorteil, dass die Decoder in den vielen Endgeräten erheblich weniger aufwendig und teuer sind als die Encoder auf der Sendeseite.

Group of Pictures (GOP)

Am Ausgang des Decoders wird, genau wie vor der Encodierung, eine Stufe zur Bildumsortierung durchlaufen. Diese ist erforderlich, weil nicht alle Bilder gleich codiert werden. Wie bereits beschrieben ist die effizienteste Form die, die auf bidirektionaler Prädikation beruht. Da die B-Frame-Codierung aber von den Nachbarbildern abhängt, kann sie nicht für alle Bilder durchgängig beibehalten werden, sondern im Datenstrom müssen in regelmäßigen Abständen auch unidirektional prädiziert P-Frames und auch I-Frames auftauchen, die nicht von den Nachbarbildern abhängen, da ein Decoder sonst keinen Einstiegspunkt finden kann. Bei MPEG werden die I-, B- und P-Frames periodisch wiederholt. Auf diese Weise entstehen zyklisch wiederkehrende Bildgruppen, die als Group of Pictures (GOP) bezeichnet werden. Die GOP-Struktur wird bei der Encodierung festgelegt, je mehr B- und P-Frames auftauchen, desto effektiver ist die Datenreduktion. In der Praxis wird oft mit einer Folge von 12 Bildern mit einem I-, drei P- und viermal zwei B-Frames gearbeitet.

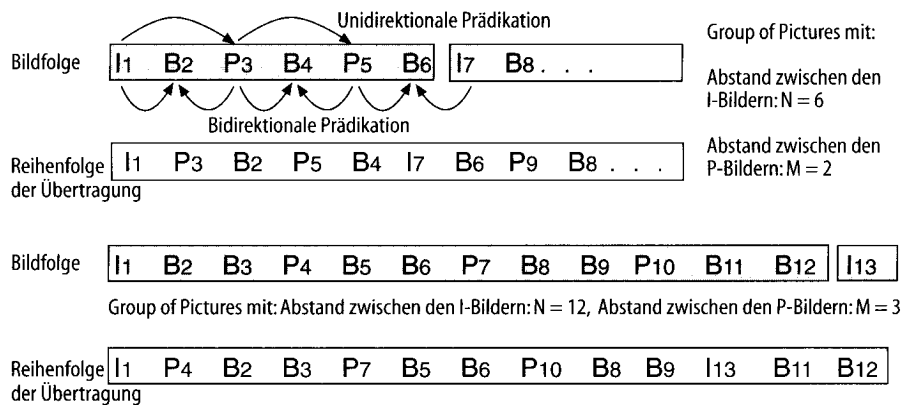
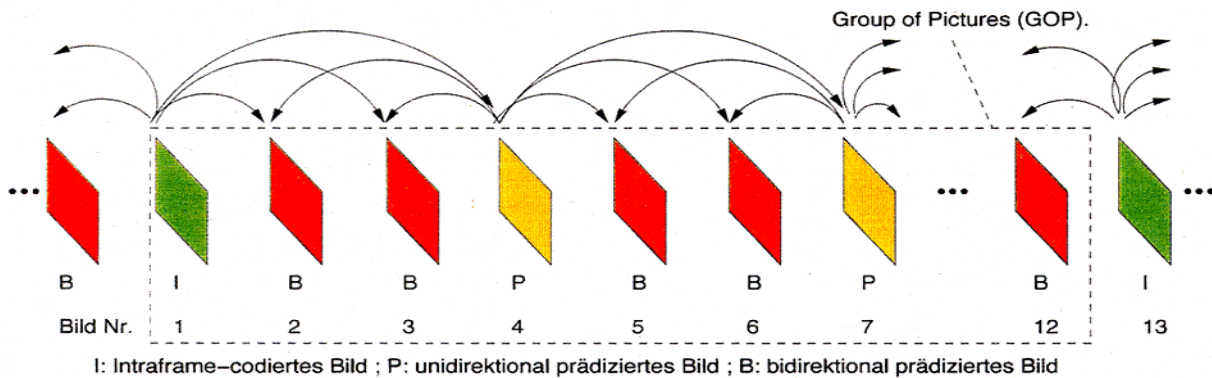


Bild: Verschiedene Formen von Groups of Pictures

Die Bildreihenfolge wird nun für die Übertragung mit der erwähnten Bildumsortierungsstufe verändert, denn dem Decoder müssen zuerst die Nachbarbilder eines B-Frames bekannt sein, damit dieses decodiert werden kann. Für die Verarbeitung mehrerer benachbarter B-Frames müssen viele Bilder gespeichert werden, das geht mit einer entsprechend großen Verzögerungszeit einher, die die Verwendung sehr langer Folgen von B-Frames einschränkt.

MPEG-1

Die MPEG-Codierung wurde zunächst als MPEG-1 im Hinblick auf kleine Datenraten von bis zu 1,5 Mbit/s ausgelegt. Eine typische Anwendung ist z. B. das CD I-Format (Compact Disc Interactive). MPEG-1 arbeitet mit der Hybrid-DCT mit Blöcken aus 8 x 8 Bildpunkten, die Datenrate wird zusätzlich dadurch vermindert, dass in einer Vorverarbeitungsstufe die Abtastwerte horizontal und vertikal verringert werden. MPEG-1 bezieht sich auf progressiv abgetastete Bilder mit 352 X 288 Bildpunkten bei 50 Hz Bildwechselfrequenz, bzw. 352 X 240 Punkten bei 60 Hz. Das Bild unten zeigt wie diese, als Source Input Format (SIF) bezeichnete Struktur durch Unterabtastung aus dem Digitalsignal nach ITU-R 601 gewonnen wird. Die Abtrennung der Pixel am Bildrand ist erforderlich, um für die Makroblöcke eine durch 16 bzw. 8 teilbare Zahl zu erhalten. Auf der Wiedergabeseite werden die 720 X 576 Bildpunkte durch Interpolation aus den übertragenen Werten zurückgewonnen.

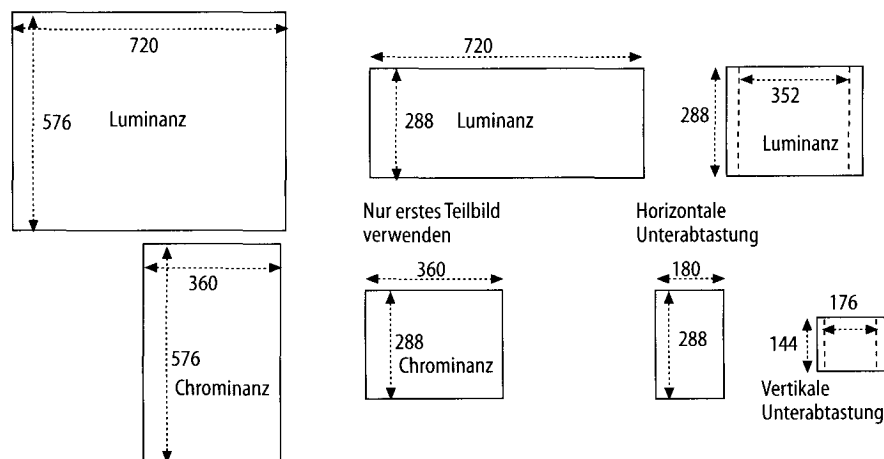


Bild: Vorverarbeitung durch Unterabtastung bei MPEG-1

MPEG-2

Auch die Codierung nach MPEG-2 beruht auf der in diesem Abschnitt allgemein dargestellten MPEG-Videocodierung auf Basis der hybriden DCT mit Blöcken aus 8 x 8 Bildpunkten. Im Gegensatz zu MPEG-1 sind aber mehrere Bildauflösungen zwischen dem SIF von MPEG-1 bis hin zu HDTV zugelassen. Neben der am häufigsten angewendeten 4:2:0-Abtastung sind auch die Formate 4:2:2 oder 4:4:4 erlaubt.

Darüber hinaus wird die Verarbeitung von Halbbildern unterstützt. Wie beim DV-Algorithmus kann bei der Encodierung zwischen dem Voll- und dem Teilbildmodus gewählt werden. Der Vollbildmodus nutzt die Ähnlichkeiten der Nachbarzeilen aus zwei Halbbildern, die bei ruhender Bildvorlage auftreten und beruht daher auf 8 x 8-Bildblöcken die aus beiden Halbbildern stammen. Bei starken Bewegung im Blockbereich sind die Halbbilder meist so stark unterschieden, dass es günstiger ist, auf die Teilbildverarbeitung überzugehen und die Zeilen entsprechend umzuordnen (Bild auf der nächsten Seite), so dass nur die Ähnlichkeiten innerhalb des Teilbildes ausgenutzt werden.

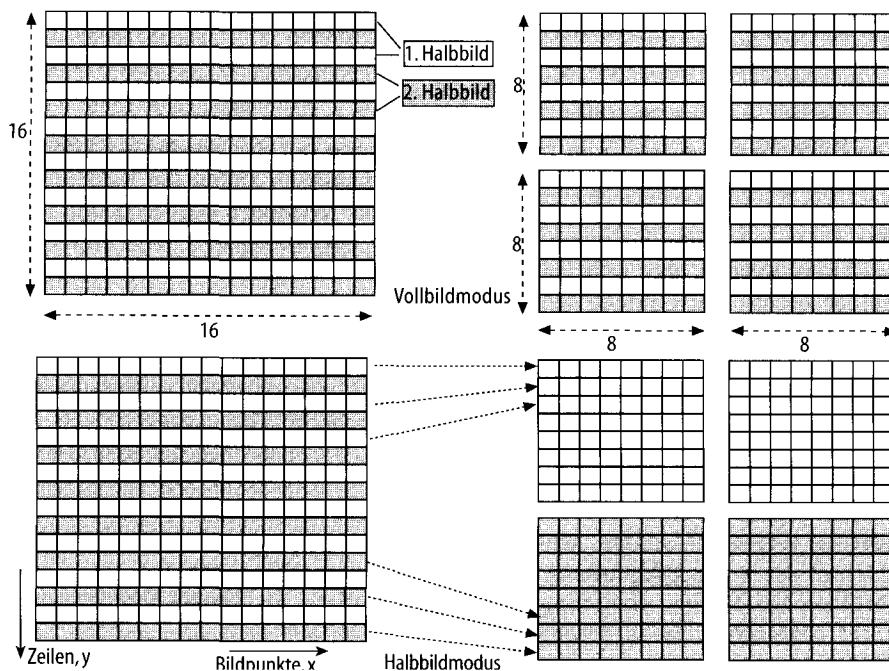


Bild: Makroblockaufteilung in 8x8-DCT-Blöcke zur Halbbildverarbeitung

Als universeller Standard erlaubt MPEG-2 die Codierung von Videobildern mit Auflösungen auf dem Niveau MPEG-1, mit Standard-Auflösung (720 x 576 Bildpunkte) bis hin zu HDTV-Auflösung (1920 x 1152 Bildpunkten), die Daten können halb- oder vollbildorientiert verarbeitet werden. Auch bezüglich weiterer Funktionen stellt MPEG-2 eine so reiche Auswahl an Möglichkeiten zur Codierung zur Verfügung, dass dafür gesorgt wurde, diese auf einige wenige, häufig verwendete Funktionalitäten (Profile) bei verschiedenen Auflösungsstufen (Levels) zu beschränken. Damit wird es möglich, den unterschiedlichen Aufgaben auf einfache Weise gerecht zu werden und klare Anforderungen an die Leistungsfähigkeit der Decoder zu formulieren. Fünf verschiedene Profile werden zusammen mit vier Stufen (Level) der Bildauflösung für Low bis High-Definition-TV in einer Matrix angeordnet.

Das Bild auf der nächsten Seite zeigt, welche Kombinationen dabei zulässig sind. Aus der Abbildung ist auch die jeweils maximal erlaubte Datenrate, sowie die Anzahl der verwendeten Bildpunkte ersichtlich. Weiterhin wird deutlich, wie die Farbdifferenzsignale relativ zum Luminanzanteil abgetastet werden (meist 4:2:0). Der Low Level entspricht einer durch MPEG-1 festgelegten Bildauflösung, es ist aber eine höhere Datenrate bis zu 4 Mbit/s erlaubt. Der Main Level zielt auf die Mehrzahl der Videoanwendungen ab, die Auflösung entspricht der heutigen Standardauflösung. Die beiden High Levels mit den hohen Auflösungen sind für HDTV-Anwendungen konzipiert. Das Simple Profile richtet sich auf einfachste Lösungen ohne die MPEG-2-spezifische bidirektionale Prädikation. Das Main Profile at Main Level (MP@ML) mit einer Auflösung von 720 x 576 Pixeln hat die größte Bedeutung, es bietet sich bei mäßigem technischen Aufwand für eine Vielzahl von Anwendungen an. Ein mit Standard-PAL-Qualität vergleichbares Signal erfordert hier eine Datenrate von ca. 5 Mbit/s, die maximal erlaubte Datenrate beträgt 15 Mbit/s. Das High Profile zielt auf hochqualitative Anwendungen, hier kann insbesondere mit erhöhter vertikaler Farbauflösung (4:2:2) gearbeitet werden. Zwischen Main und High Profile sind die skalierbaren Profile angeordnet. Skalierbare Profile werden unterschieden in eine Skalierung bezüglich des Signal-Rauschspannungsabstandes (SNR) und des Raumes (Spatially). Die skalierbare SNR-Codierung ist für die terrestrische Ausstrahlung von Videobildern von Bedeutung. Sie soll das typische »Alles oder Nichts« der Digitaltechnik

verhindern, d. h., dass das System bei schwankender Übertragungsqualität entweder gut funktioniert oder gar nicht.

Level	Low Level	Main Level	High Level 1440	High Level 1920
max. Bildwechsel	30/s	30/s	60/s	60/s
Profiles				
Simple Profile 4:2:0 (keine B-Frames)		720 x 576 Pixel < 15 Mbit/s		
Main Profile 4:2:0	352 x 288 < 4 Mbit/s	720 x 576 < 15 Mbit/s	1440 x 1152 < 60 Mbit/s	1920 x 1152 Pixel < 80 Mbit/s
SNR Scalable Profile 4:2:0	352 x 288 < 4 (3) Mbit/s	720 x 576 < 15 (10) Mbit/s		
Spatially Scalable Profile 4:2:0			1440 x 1152 (720 x 576) < 60 (40) Mbit/s	
High Profile 4:2:0 oder 4:2:2		720 x 576 (352 x 288) < 20 (15) Mbit/s	1440 x 1152 (720 x 576) < 80 (60) Mbit/s	1920 x 1152 Pixel (960 x 576) < 100 (80) Mbit/s

Werte in Klammern geben die Alternativen bezgl. der Skalierung an

Bild: Profiles und Levels bei MPEG-2

Stattdessen soll sich ein Verhalten wie bei analogen Systemen einstellen, die Bildqualität soll also bei verschlechtertem Signal-Rausch-Verhältnis allmählich und nicht abrupt abnehmen (Graceful Degradation, Bild unten).

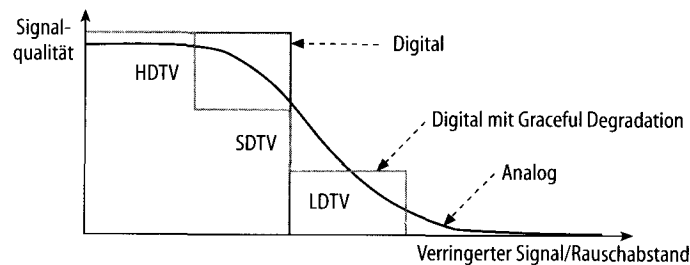


Bild: Graceful Degradation

Die SNR-Skalierbarkeit wird erreicht, indem zwei Datenströme mit Bildern gleicher Auflösung aber unterschiedlicher Störsicherheit erzeugt werden. Zunächst wird eine DCT mit grober Quantisierung gewählt und damit ein Datenstrom für Bilder mit geringer Qualität erzeugt, für den eine sehr störsichere Übertragungsart gewählt wird. Daneben wird der Quantisierungsfehler berechnet und codiert in einem zweiten Datenstrom übertragen. Die Signalqualität am Empfänger ist davon abhängig, ob der Signal-Rauschabstand ausreicht, beide Datenströme auszuwerten oder nicht.

Die räumliche (Spatially) Skalierung bezieht sich auf unterschiedliche Bildauflösungen. Damit soll eine hierarchische Codierung ermöglicht werden, die aus nur einem Datenstrom in Abhängigkeit vom Empfänger mehrere Qualitätsstufen zwischen LDTV und HDTV ableiten kann. Bezüglich eines terrestrischen Übertragungskanal mit 8 MHz Bandbreite sind vier Qualitätsstufen vorgesehen: HDTV (High Definition), EDTV (Enhanced Definition), SDTV (Standard Definition) und LDTV (Low Definition).

Videodatenübertragung mit MPEG-2 ermöglicht eine hohe Bildqualität bei geringer

Datenrate. Eine Qualität, die mit der Wiedergabe von im VHS-System aufgezeichneten Bildern vergleichbar ist, ist mit 2 ... 4 Mbit/s erreichbar. Für eine mit dem Standard-PAL-Signal vergleichbare Qualität rechnet man mit 5...6 Mbit/s. MPEG-codierte HDTV-Signale erfordern eine Datenrate von ca. 30 Mbit/s. Beim Einsatz von Datenreduktion nach MPEG muss bedacht werden, dass bei Verwendung langer GOP die Editierfähigkeit eingeschränkt ist, da kein Zugriff auf jedes Einzelbild möglich ist.

Vor dem Hintergrund einer erfolgreichen Einführung von MPEG-2 und des darauf beruhenden DVB-Verfahrens (Digital Video Broadcasting) zur digitalen Signalverteilung an den Konsumenten entstand der Wunsch, den MPEG-Standard auch für Studioanwendungen nutzen zu können. Dafür ist eine Erweiterung vorgesehen, die mit 4:2:2- oder Professional Profile bezeichnet wird und in der üblichen Darstellung der Profiles und Levels nicht enthalten ist. Die wichtigsten Aspekte sind dabei die Unterstützung einer 4:2:2-Abtastung und die Nutzung der Zeilen in der vertikalen Austastlücke (720H x 608V). Um geringe Qualitätsverluste und den Zugriff auf jedes Einzelbild zu ermöglichen werden die »I-Frame-only«-Codierung und die »I-B-I-B«-Codierung mit Datenraten von bis zu 50 Mbit/s genutzt.

MPEG-4

Mit MPEG-4 soll erstens eine gegenüber MPEG-2 erhöhte Codiereffizienz erreicht werden, zweitens wird durch die Einführung von Objekten und Objektebenen ein interaktiver Umgang ermöglicht. Darüber hinaus wird ein universeller Zugriff auf die Daten über verschiedenste Speichermedien und Netzwerke unterstützt. Ein wichtiger Aspekt dabei ist die Synthetic Natural Hybrid Codierung (SNHC), d. h. die gemeinsame Codierung herkömmlicher audio-visueller Daten und künstlich erzeugter Graphiken und Sounds .

Aus Gründen der Abwärtskompatibilität wird die Basiscodierung nach MPEG-1/2 beibehalten, d. h. für die Videocodierung ist zunächst die übliche 8-Bit-Quantisierung von Bildern mit Luminanz- und Farbdifferenzkomponenten vorgesehen. Als Neuerung kommt ein generischer Coder hinzu, der beliebig geformte Videoobjekte definiert, die sog. Video Objekt Planes (VOP) zugeordnet werden können. Dabei kommt die Verarbeitung von Transparenz- und Konturinformationen durch grauwertige Alpha-Plane-Bilder zum Einsatz. Eine Szene kann damit in inhaltsbezogene Bestandteile zerlegt und auf verschiedene VOP verteilt werden (Bild unten).

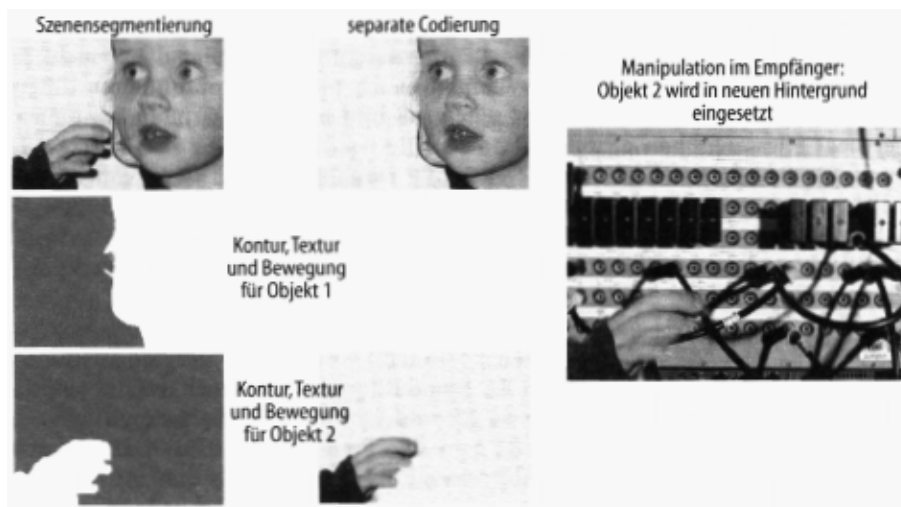


Bild: Video Objekt Planes bei MPEG-4

Dabei werden feste und variable Bildwiederholfrquenzen für die einzelnen Planes unterstützt und überlappende und nicht überlappende Bildbestandteile definiert. Die VOP werden getrennt codiert und übertragen. Erst beim Empfänger werden die Planes wieder zu einem Bild zusammengesetzt, wobei sich die Form und Position mit der Bewegung der Bildinhalte ändern kann. Auf diese Weise wird ein interaktiver Zugriff auf die Objekte möglich, die Bildentstehung kann im Empfänger manipuliert werden. Mit dem Binary Format for Scene Description (BIFS) ist dabei eine Szenenbeschreibungssprache definiert, die die Stellung der einzelnen Videoobjekte zueinander definiert. Die Interaktionsmöglichkeiten werden dadurch gesteigert, dass MPEG-4 die Kommunikation über einen Rückkanal unterstützt. Neben der Manipulierbarkeit ergeben sich auf diese Weise die neuen Möglichkeiten zur Datenreduktion. Befindet sich z. B. eine Person vor einem unbewegten Hintergrund, so kann dieser einer separaten VOP zugeordnet werden, die nur einmal pro Szene übertragen wird. Zusätzlich muss nur noch das Bild der Person in einer zweiten, überlappenden VOP übertragen werden, was nicht die Bildpunktanzahl des Gesamtbildes erfordert.

Wie bei MPEG generell, sind auch bei MPEG-4 wieder Audio- und Systembeschreibungen implementiert. Die Systemebene wird mit den neuen Funktionen sehr viel komplexer, daher kommt der dafür definierten MPEG-4 Systems and Description Language (MSDL) eine besonders große Bedeutung zu. Diese Sprache dient neben der Synchronisation und Multiplexbildung zur Beschreibung der codierten audio-visuellen Objekte und der Zusammensetzung der Objekte zu einem Gesamtbild inklusive der Unterstützung der Interaktivität. Als universelles Übertragungskonzept für den Transport multimedialer Daten ist in MPEG-4 das Delivery Multimedia Integration Framework (DMIF) definiert, welches ein standardisiertes Application Interface für alle Anwendungen bietet. Damit können die Inhalte unabhängig von den Übertragungswegen und spezifischen Protokollen übertragen werden.

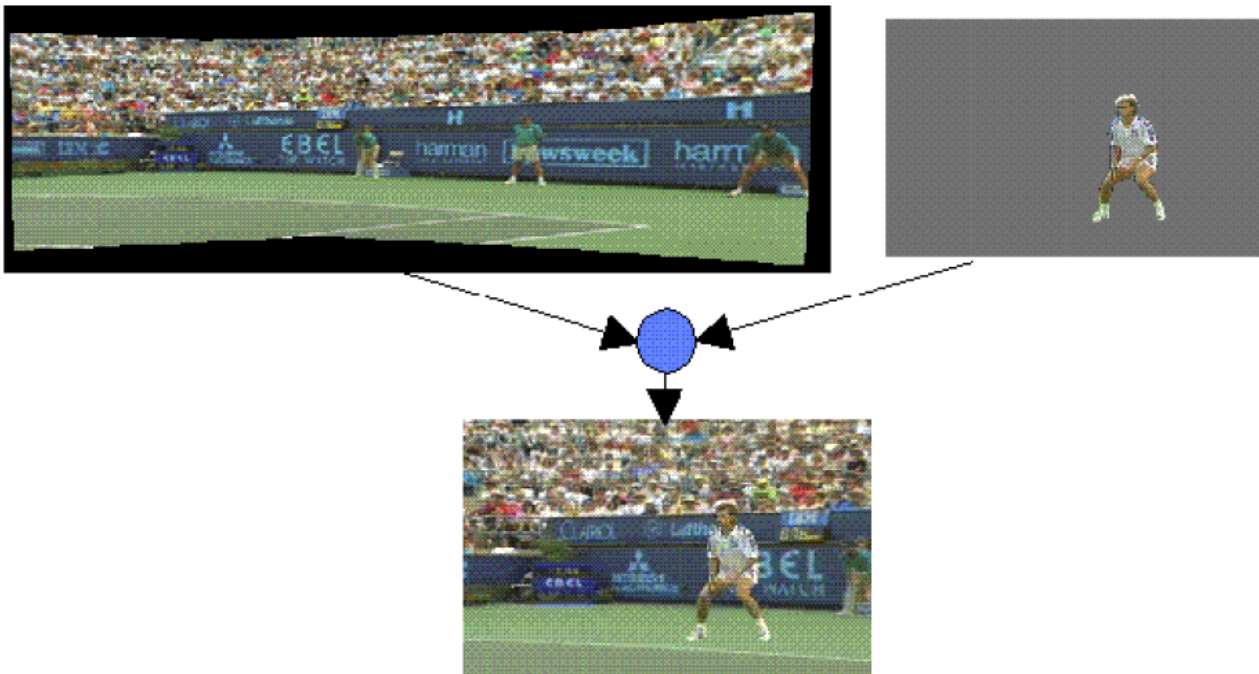


Bild: Die Videoobjekte im Vordergrund werden separat übertragen und der Decoder setzt das Sprite (Hintergrundbild) mit dem Videoobjekt wieder zusammen und zeigt den entsprechenden Bildausschnitt an.

MPEG-7

MPEG-5 und MPEG-6 Versionen wurden übersprungen. MPEG-7 versucht, Ordnung in den MPEG-Dschungel durch Metadaten-Standards zu bringen. Mit Hilfe von XML-Anwendungen soll ein **Multimedia Description Scheme (MDS)** entwickelt werden, mit dessen Hilfe sich die Audio- und Video-Inhalte indizieren, mit Schlüsselwörtern versehen und ordnen lassen. Die Entwicklung konzentriert sich im Moment also eher auf die Definition von Metainformationen und Schnittstellen, weniger auf die Entwicklung bestimmter Kompressionsverfahren oder Werkzeuge. Damit bleibt das Format offen für Experimente aber trotzdem kompatibel.

MPEG-21

Die nächste MPEG-Version soll MPEG-21 heißen, und befindet sich in der Planungsphase. In Bezug auf Formate, Codecs, Metadaten etc., zwischen kommerziellen Business-Systemen zielt MPEG-21 auf die Bereitstellung eines sogenannten „Multimedia Frameworks“ ab. Das Ziel ist die Schaffung einer heterogenen Landschaft von Geräten und Netzwerken, die eine globale Anwendbarkeit anstrebt.

MPEG-Audio

Beim MPEG-Standard ist eine datenreduzierte Audioübertragung vorgesehen. Zum Verständnis der Reduktionsverfahren müssen zunächst einige Phänomene der menschlichen Hörwahrnehmung bekannt sein.

Der wesentliche Vorgang beim Hören ist die Umsetzung von Schalldruckschwankungen in Nervenreize. Die Lautheitsempfindung steigt dabei nicht linear mit dem Schalldruck. Der Dynamikbereich, also der empfundenen Unterschied zwischen laut und leise, ist sehr groß und umfaßt 6 Größenordnungen, er liegt etwa zwischen 2×10^{-5} und 20 Pa. Die Grenzen werden als Hörschwelle und Schmerzgrenze bezeichnet. Der große Dynamikbereich wird vom Gehör nicht linear verarbeitet, daher ist es vorteilhaft, die Schallwechselgrößen in logarithmischer Form als Pegelwerte anzugeben, wobei der Bezugswert (Pegel 0 dB) etwa dem Schalldruck der Hörschwelle entspricht. Der Schalldruckpegel berechnet sich nach der Beziehung

$$L = 20 \log p/p_0.$$

Als Bezugswert gilt $P_0 = 2 \cdot 10^{-5}$ Pa. Ein mittlerer Schalldruck von $2 \cdot 10^{-3}$ Pa entspricht einem Schallpegel von 60 dB und wird als mittellaut empfunden.

Die Gehörempfindlichkeit hängt von der Frequenz ab. Nur bei mittleren Frequenzen (1-4 kHz) liegt die Hörschwelle in der Größenordnung $2 \cdot 10^{-5}$ Pa, bei höheren und tieferen Frequenzen nimmt die Empfindlichkeit des Gehörs ab und erreicht die Grenzen der Unhörbarkeit bei 16 Hz bzw. 20 kHz, wobei die obere Grenze mit zunehmendem Alters der Hörer sinkt. Der Verlauf der Frequenzabhängigkeit der Hörempfindung bei verschiedenen Pegeln ist im Bild (nächste Seite) als sog. Hörfläche dargestellt. Hier wird deutlich, welcher

Schallpegel bei jeder Frequenz für eine bestimmte Lautheitsempfindung erforderlich ist (Kurven gleicher Lautstärke mit der Einheit „phon“, entsprechend dB bei 1 kHz).

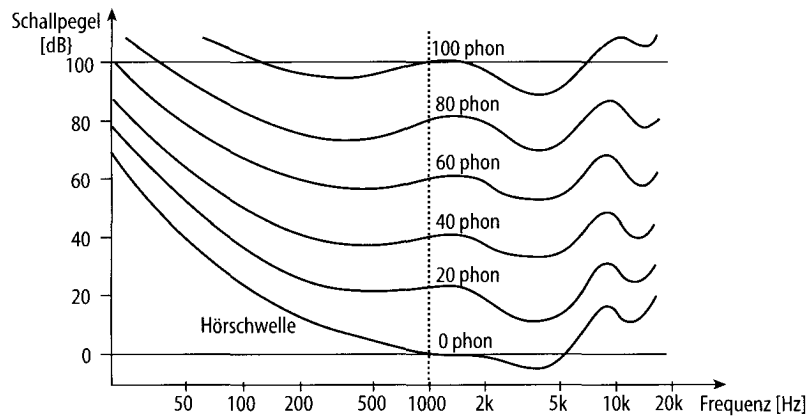


Bild: Die Hörfläche

Zur Signalverarbeitung wird der Schalldruck mit Mikrofonen möglichst linear in elektrische Spannungen umgesetzt und für die anschließende Digitalisierung wird die Spannung i. d. R. mit einer Rate von 48 kHz abgetastet. Um einen guten Signal-Rauschabstand zu erhalten, wird jeder Abtastwert mit 16 Bit dargestellt. Daraus folgt ein Dynamikbereich von 96 dB, der sich aber durch Head- und Feetroom (Über- und Untersteuerung) auf ca. 70 dB verringert. Dieser Wert ist völlig ausreichend, da einerseits die Schallsignale selbst weniger Dynamik aufweisen und andererseits auch bei der Wiedergabe nicht mehr umgesetzt werden kann. So liegt z. B. schon bei der Abhörsituation keine absolute Stille vor, so dass Rauschsignale, die um 70 dB unter dem Maximalpegel liegen, nicht mehr wahrgenommen, sondern verdeckt werden.

Dieser Effekt, bei dem Schallereignisse durch andere überdeckt werden können (Verdeckungseffekt) ist es, der bei den Audio-Datenreduktionsverfahren in geschickter Weise ausgenutzt wird. Dabei interessiert besonders der Umstand, dass nicht nur Fremdsignale das Nutzsignal verdecken können, sondern auch Spektralanteile im Signal selbst. Durch Experimenten mit definierten Tönen oder Rauschen, lässt sich feststellen, in welcher Form die Ruhehörschwelle, bzw. die Kurven gleicher Lautstärke in der Hörfläche durch diese sog. Maskierer verändert werden.

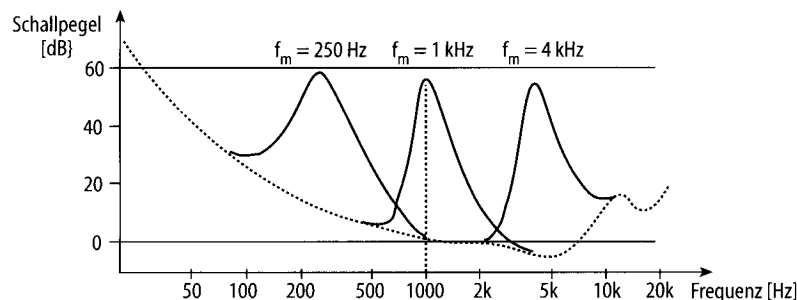


Bild: Mithörschwellen resultierend aus Schmalbandrauschen dreier Mittenfrequenzen

Das Bild oben zeigt als Beispiel das Verhalten des Gehörs bei Anregung durch Schmalbandrauschen mit drei verschiedenen Mittenfrequenzen bei einem Schallpegel von 60 dB. Der sich ergebende gesamte Kurvenverlauf wird als Mithörschwelle bezeichnet. Signale die unter dieser Schwelle liegen sind unhörbar, sie werden von den Maskierern verdeckt. Dies gilt auch für das Quantisierungsrauschen. Daraus resultiert die Grundidee

der Datenreduktion: jeder Abtastwert wird nur mit so wenig Bits quantisiert, dass das Quantisierungsrauschen gerade unter der Mithörschwelle liegt.

Bei MPEG wird diese Idee mit Hilfe des sog. Subbandcoding realisiert. Dafür sind als Abtastraten für das digitale Eingangssignal folgende Standardwerte zugelassen: 32 kHz, 44,1 kHz und 48 kHz, bei MPEG-2 auch die jeweils halbierten Werte. Das von 20 Hz bis 20 kHz reichende Audiospektrum wird dann in 32 gleich große Teilbänder (Subbands) aufgespalten, wobei die Abtastrate für die Teilbänder gegenüber dem ungeteilten Frequenzspektrum um den Faktor 32 reduziert wird. Anschließend werden die gewonnenen Werte quantisiert.

Um die Signalverarbeitung flexibel durchführen zu können, wurde der MPEG-Audio-Standard in drei Layer eingeteilt. Layer 1 bietet die geringste Effektivität der Datenreduktion, erfordert dafür aber auch den geringsten Implementierungsaufwand, während für den Layer 3 das Umgekehrte gilt. Der Rechenaufwand schlägt sich auch in der Zeitverzögerung nieder, die für den Codiervorgang gebraucht wird, die Minimaldelay-Werte betragen 19 ms, 35 ms und 59 ms für die Layer 1, 2, 3. Bei Internetverbindungen, wo es auf minimale Datenrate ankommt, wird meist Layer 3 verwendet, der oft auch als MP3 bezeichnet wird. Für DVB wird Layer 2 eingesetzt. Das Bild unten zeigt das Blockschaltbild des Coders nach Layer 2. Es wird ein Block von 36 Abtastwerten gebildet, die bei einer Samplerate von 48 kHz einer Dauer von 24 ms entsprechen. Für jeden Block wird ein Skalenfaktor bestimmt, durch den die zugehörigen Werte geteilt werden, bevor sie quantisiert werden. Zur Festlegung der Zahl der Quantisierungsstufen wird aus den Skalenfaktoren, unterstützt durch eine Fouriertransformation, das psychoakustische Modell gebildet, das Auskunft über die Maskierungsschwelle gibt.

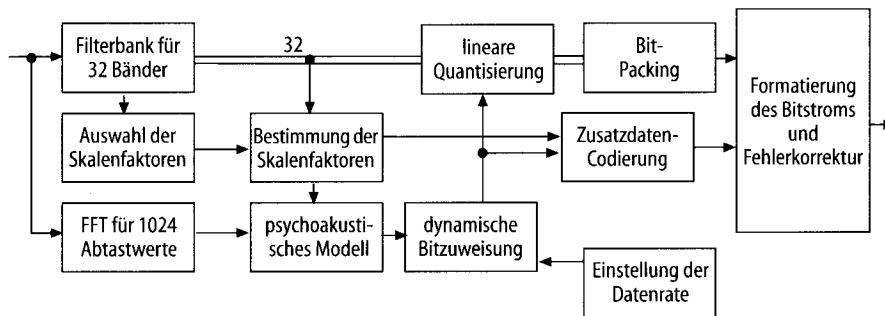


Bild: Blockschaltbild der Audiocodierung nach MPEG-Layer 2

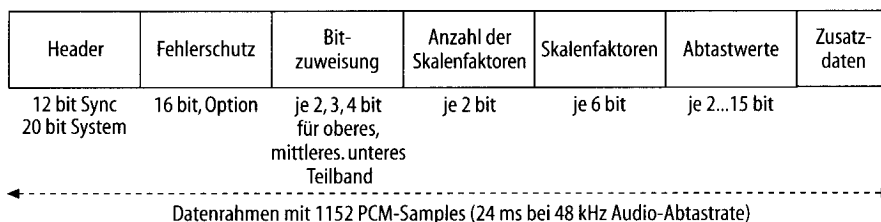


Bild: Audio-Datenrahmen nach MPEG-Layer 2

Die gewonnenen Daten werden in eine Frame-Struktur gebracht und mit einer typischen Datenrate von 128 kBit/s pro Kanal in den MPEG-Gesamtdatenstrom integriert. Die Maximaldatenrate beträgt 448 kBit/s bei Layer 2. Das Bild oben zeigt den Datenrahmen, der mit einem 32 Bit-Header beginnt und Systemdaten zum Audiosignal im Stereoformat

enthält. Optional kann anschließend ein Fehlerschutz verwendet werden. Es folgen die Angabe der Bitanzahl, die für jeden Abtastwert verwendet wird und dann die Skalenfaktoren, durch die Audioblöcke, die jeweils 12 Abtastwerte umfassen, geteilt werden. Vorher ist bei MPEG-2 eine Kennung für die Anzahl der Skalenfaktoren eingefügt. Auf die Skalenfaktoren folgen die eigentlichen Abtastwerte, die mit 2 bis 15 Bit codiert sind und daran schließen sich Zusatzdaten an, die für besondere Mehrkanalerweiterungen reserviert sind.

MPEG-1 erlaubt nur die Nutzung von zwei Audiokanälen, unabhängig oder als Stereokanäle. MPEG-2 hingegen enthält als Erweiterung die Möglichkeit der stereokompatiblen Surround-Codierung. Der Begriff Surround steht für das fünfkanalige 3/2 Format, das mit drei vor dem Hörer (Left, Center, Right) und zwei hinter dem Hörer platzierten Lautsprechern arbeitet (Left Surround, Right Surround) und gegenüber einfachen Stereosystemen eine deutliche Verbesserung der akustischen Raumabbildung gestattet (Bild unten).

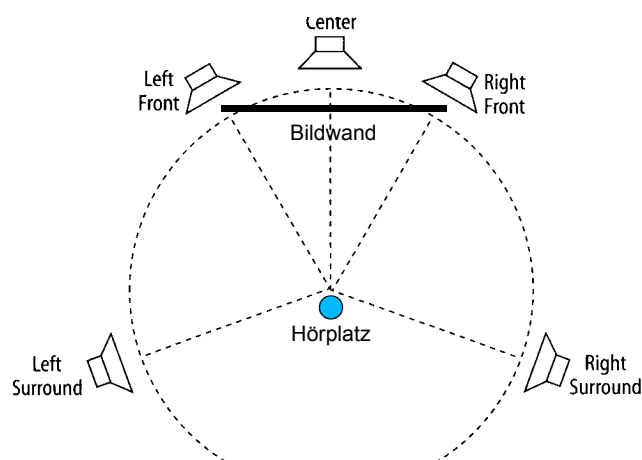


Bild: Lautsprecheranordnung beim 3/2 Surroundverfahren

MPEG-Systems

Die wesentliche Aufgabe des Teils »Systems« von MPEG ist die Definition der Multiplexstruktur für die verschiedenen Datenarten unter Berücksichtigung der Synchronisierung von Audio- und Videodaten mit ihren jeweils unterschiedlichen Datenraten.

Um die Synchronität zu erreichen, ist es erforderlich, die Datenströme in Portionen aufzuteilen, sie zwischenspeichern und im Multiplex zu übertragen. Das Multiplexverfahren bestimmt die Strukturierung der Daten, es ist in der Lage, nicht nur mit festen, sondern auch mit flexiblen Datenraten zu arbeiten. Weiterhin können die Übertragungskapazitäten mehreren Programmen zugeordnet werden.

Die Daten werden in zunächst in Paketen aufgeteilt, die jeweils nur eine Datenart (Bild, Ton, Daten) enthalten (Bild auf der nächsten Seite).

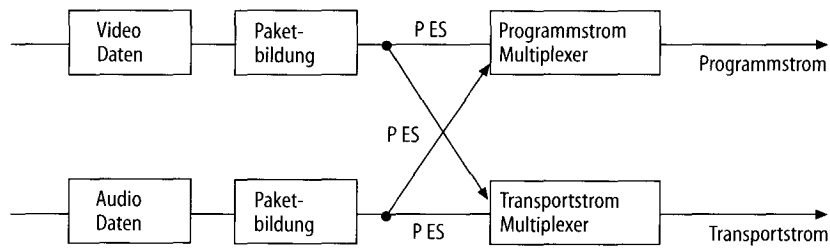


Bild: Bildung des Packet Elementary Stream

Dieser Grunddatenstrom wird als Packet Elementary Stream (PES) bezeichnet.

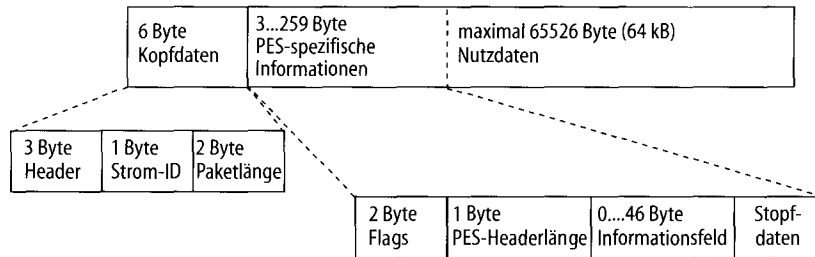


Bild: Aufbau eines PES-Pakets

Das Bild oben zeigt den Aufbau der Datenpakete. Der Beginn wird mit einem Header von 6 Byte Umfang gekennzeichnet, hierin befindet sich die Information über die Art der Daten (Audio, Video, Daten) und über die PES-Länge, die variabel sein darf. Ein Videostream mit Nummer »nnnn« wird durch die Bitfolge »1110 nnnn« gekennzeichnet. Darauf folgt ein Feld variabler Größe, das PES-spezifische Informationen enthält und schließlich die eigentlichen Nutzdaten (Payload) mit einem Umfang von maximal 65526 Byte, einer Zahl, die im Header gerade mit zwei Byte dargestellt werden kann. Die PES-spezifischen Informationen enthalten sog. Flags, die die Übertragung von periodisch eingesetzten Zeitmarken zur Synchronisation (PTS und DTS, s. u.) anzeigen. Ein weiteres Flag wird z. B. gesetzt, wenn im PES die Daten eines Paketzählers übertragen werden mit Hilfe dessen erkannt werden kann, ob bei der Übertragung ein Paket verloren ging. Die Pakete des Elementarstroms dürfen nach der MPEG-Definition nicht direkt übertragen werden. Sie werden entweder in einen Programm-Strom oder einen Transport-Strom integriert. Der Program Stream ist für wenig fehleranfällige Anwendungen (z. B. CD-ROM, DVD) konzipiert, er erlaubt Pakete variabler Länge und arbeitet mit nur einer Zeitbasis. Der Programmstrom besteht aus einer Folge von PES, die alle zu einem Programm gehören. Die PES-Folge ist periodisch durch Pack- und System Header unterbrochen, die Synchroninformationen bereitstellen und einen Rahmen bilden, der den PES in Gruppen unterteilt (Bild unten).

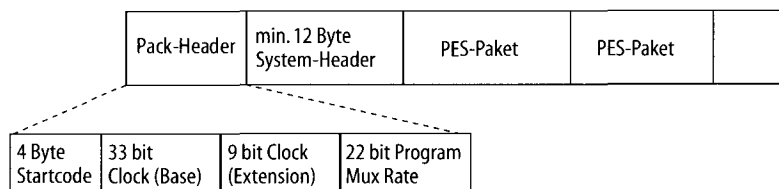


Bild: Struktur des Programmstroms

Der Programmstrom erlaubt die Verwendung langer Pakete. Ein Datenverlust ist schwer auszugleichen, daher sollte der Programmstrom nur bei Übertragungsarten angewandt

werden, bei denen sehr wenige Fehler auftreten.

Der Transport Stream ist dagegen für stör anfällige Broadcast-Anwendungen gedacht. Im Hinblick auf einfach zu konstruierende Empfangsgeräte ist die Paketlänge fest vorgegeben, dabei sind mehrere Zeitbasen zugelassen. Das Bild unten zeigt schematisch die Multiplexbildung. PSI steht darin für programmspezifische Informationsdaten, z. B. Informationen über Multiplexbestandteile oder Verschlüsselungen.

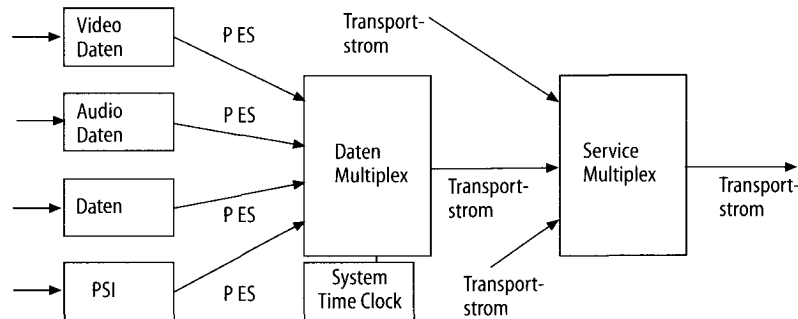


Bild: Daten- und Transportstrom-Multiplex

In dem für den Broadcast-Bereich wichtigen Transport Stream hat jedes Paket einen Umfang von 188 Byte, von denen 184 Byte für Nutz- oder Adaptiondaten zur Verfügung stehen. Der im Vergleich lange Elementarstrom wird in Stücke aufgeteilt, die in den Nutzdatenbereich passen (Bild unten).

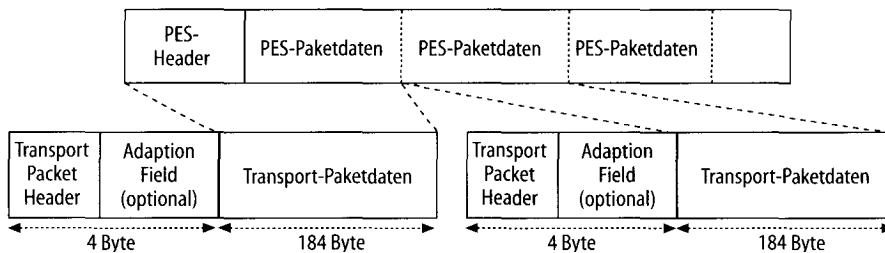


Bild: Zuordnung von PES- und Transportstrom-Paketen

Wenn das letzte TS-Paket nur noch einen Rest des PES aufnimmt, wird der freibleibende Teil durch sog. Stopf-Bytes gefüllt. Im Adaptionfeld werden Steuerdaten untergebracht, die nicht in jedem Fall benötigt werden. Ein Paket wird immer von 4 Byte Kopfdaten eingeleitet.

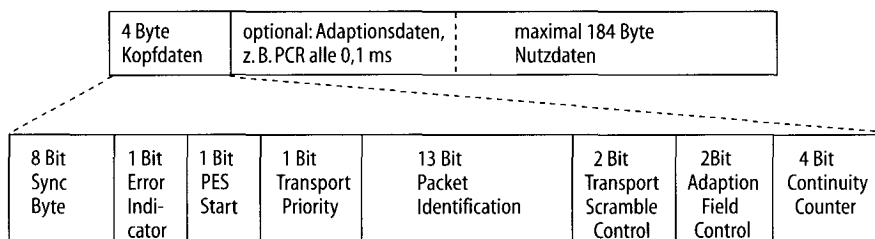


Bild: Bedeutung der Kopfdaten im Transportstrom-Paket

Das Bild oben zeigt die Bedeutung der 32 Bits. Der Anfang des TS-Pakets wird vom Synchronwort mit dem Wert »0100 0111« bestimmt. Das TEI-Bit zeigt an, ob bei der Übertragung ein nicht korrigierbarer Fehler aufgetreten ist. Die 13 Bit der Packet Identification (PID) dienen der Identifikation der Nutzdaten und dem

Auffinden eines einzelnen Elementarsignals im Multiplex. Der Continuity Counter nummeriert schließlich alle Pakete mit der selben PID gemäß der gesendeten Reihenfolge.

Aufgrund der unterschiedlichen und variablen Laufzeiten für die verschiedenen Bestandteile des Transportstroms ist die Synchronisierung der Teile eine besondere Aufgabe, die alle Signalverarbeitungsstufen im Encoder und Decoder und den Übertragungskanal berücksichtigen muss. Zur Synchronisation wird eine System Time Clock (STC) als Zeitreferenz eingesetzt, sie arbeitet bei MPEG-1 mit 90 kHz und bei MPEG-2 mit 27 MHz. Ein Zeitraum von 24 Stunden wird mit 42 Bit festgelegt. Der Datenstrom wird regelmäßig mit einer aus der STC abgeleiteten Zeitmarke (Program Clock Reference, PCR) versehen, damit der Encoder-Takt auf der Empfangsseite regeneriert werden kann. Es ist festgelegt, dass die Referenz mindestens alle 100 ms im Datenstrom eingesetzt wird (bei DVB alle 40 ms). Wenn Audio- und Videodaten übertragen werden, die zu unterschiedlichen Programmen gehören, so ist es möglich, dass sie separate Zeitmarken aufweisen. Jede Änderung des Multiplexes ist also mit der Änderung der PCR-Markierung (Restamping) verbunden. Der Decoder erzeugt einen eigenen Takt, der fortwährend mit der PC-Referenz verglichen und ggf. korrigiert wird. Es wird also vorausgesetzt, dass auf der Sende- und Empfangsseite jeweils eine Systemuhr vorhanden ist, die die aktuelle Uhrzeit festlegt. Auf der Sendeseite wird den Datenpaketen beim Verlassen des Multiplexers die aktuelle (System-) Uhrzeit aufgeprägt. Durch Auswertung der Zeitmarken wird die Systemzeit auf der Empfangsseite rekonstruiert. Die Audio- und Videodaten sind bezüglich ihrer Abtastraten taktverkoppelt. Mit Hilfe der Zeitmarken ist so gewährleistet, dass die Ton- und Bildsignale synchron sind. Neben der Systemzeitbasis sind relative Zeitmarken für die Decodierung und die eigentliche Wiedergabe definiert (Decoding and Presentation Time Stamp, DTS und PTS), die wie die PCR in regelmäßigen Abständen im Adaption Field nach den PES-Header-Daten übertragen werden. Diese Unterscheidung ist erforderlich, da sich, schon wegen der veränderten Bildreihenfolge für die bidirektionalen Prädikation, der Decodierungszeitpunkt vom Präsentationszeitpunkt unterscheiden kann.

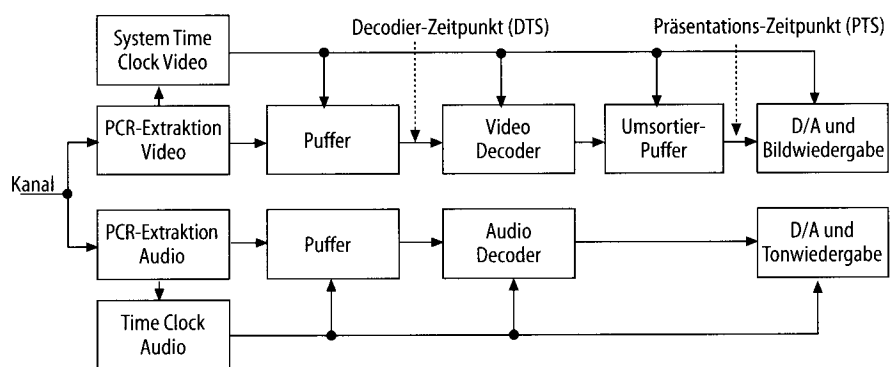


Bild: Synchronisation im Decoder

Das Bild oben zeigt die Lage der Decodier- und Präsentationszeitpunkte und weiterhin die Daten zur Systemkontrolle, die das Demultiplexing beeinflussen können.

Um auf die MPEG-Datencontainer zu zugreifen und die darin enthaltenen Bestandteile extrahieren zu können, sind im Systems-Teil programmspezifische Informationen (PSI) definiert. MPEG selbst legt vier Tabellen fest, für weitere Anwendungen, wie z. B. DVB, können eigene Tabellen definiert werden. Die erste der vier Tabellen enthält eine Liste der Programme, die im Transport-Multiplex übertragen werden (Program Association Table, PAT), die zweite einen Verweis auf die Paket-Identifikation für ein Programm (Program

Map Table, PMT), damit die Video- und Audiodatenpakete dem richtigen Programm zugeordnet werden können. Zwei weitere Tabellen bieten Platz für Informationen zur Entschlüsselung von Programmen, die nur bedingten Zugriff erlauben (Conditional Access Table, CAT) und für Informationen zum Übertragungssystem (Network Information Table, NIT).