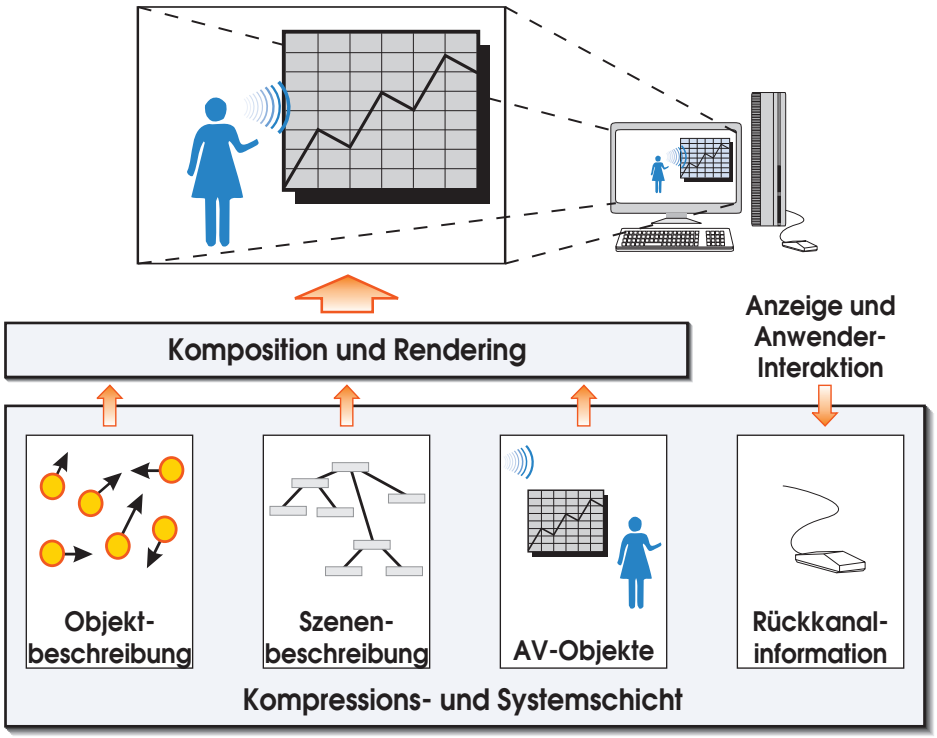
Prof. Dr.-Ing. Thomas Bonse • Prof. Dr.-Ing. Firoz Kaderali
unter Mitarbeit von Dr.-Ing. Michael Stepping

Digitale Bildcodierung

Grundlagen der digitalen Bildtechnik



Vorwort von Herrn Prof. Dr. Kaderali

Für die Erstellung dieses Kurses danke ich meinem Kollegen Prof. Dr. Thomas Bonse. Mein ehemaliger Mitarbeiter Dr.-Ing. Michael Stepping war ihm behilflich und hat neben der redaktionellen Mitarbeit auch die Umsetzung mit dem Redaktionssystem FuXML vorgenommen. Die Inhalte liegen als Kurs und als Buch dank seines engagierten Einsatzes vor. Die Buchdruckform (PDF) als auch die Internetform (HTML) mit zahlreichen interaktiven Beispiel (Animationen) liegen vor.

Die Erstellung der Grafiken erfolgte durch Herrn Bonse. Überarbeitungen wurden durch Herrn Stepping vorgenommen.

Hagen im Sommer 2005 und Sommer 2007

Gliederung

1 Einführung in die Bildtechnik

1.1 Übersicht – Warum Bilder „*codieren*“

Etwa mit Beginn der 90er Jahre beeinflussten vor allem zwei Entwicklungen den Umgang mit Bewegtbildern entscheidend: Es sind dies zum einen die rasanten Veränderungen im Bereich der digitalen Bildverarbeitung (*digitale Bildcodierung* und *Übertragung*) und zum anderen die sich hierauf aufbauenden zahlreichen neuen Anwendungsplattformen, wie beispielsweise Multimedia (*MM*), der Freizeitsektor, zahlreiche neue Kommunikations- und Datendienste sowie nicht zuletzt die Computer Animation (*CA*).

Eine entsprechende Bildqualität vorausgesetzt, benötigen die Signale bzw. die Daten von analogen bzw. digitalen Bewegtbildern eine relativ hohe Kanalbandbreite bei die Übertragung (TV-Kanalabstand: 8 MHz) oder eine hohe Kapazität bei der Speicherung auf entsprechenden Bild- und Datenträgern (z. B. *magnetische* oder *optische Speicherung*). Um eine entsprechende Ökonomie zu erreichen, ist die optimale Anpassung der Signalströme für den jeweiligen Anwendungsfall erforderlich. Insbesondere im digitalen Umfeld wird eine solche Anpassung im allgemeinen „*Codierung*“ genannt. Die Bildcodierung ist aber nicht prinzipiell auf digitale Bildsignale beschränkt. Auch beim analogen Fernsehen spricht man beispielsweise von Codierung, wenn es um die optimale Nutzung des TV-Übertragungskanals durch die Einführung der Zeilensprungtechnik geht. Auch die Einbettung des analogen *Farbartsignals* in das *Helligkeitssignals* stellt in diesem Zusammenhang eine Codierung dar. Es hat sich jedoch gezeigt, dass sich optimale Codierungen im analogen Umfeld oft nur mit erheblichem Aufwand realisieren lassen. Im digitalen Umfeld hingegen lassen sich erheblich vielfältigere Algorithmen entwickeln und in entsprechenden Computerumgebungen direkt verwirklichen. In sehr vielen Fällen geht es daher bei der *digitalen Bildcodierung* um eine Bewegtbildrepräsentation mit erheblich reduziertem Datenaufkommen, ohne dass es dabei zu einer sichtbaren Bildqualitätsverschlechterung kommt.

Der vorliegende Kurs „*Digitale Bildcodierung*“ beschäftigt sich mit den Grundlagen der digitalen Bildverarbeitung, wie sie heutzutage in zahlreichen Anwendungen genutzt werden. Als Anwendungsbeispiele sind hier zu nennen die *Digitale Fotografie*, die digitale Videodisk – *Digital Versatile Disk* (*DVD*) oder das digitale Fernsehen – *Digital Video Broadcasting* (*DVB*).

Viele der aktuell in der digitalen Bildcodierung verwendeten Algorithmen sind das Ergebnis jahrzehnte langer intensiver, interdisziplinärer Forschungs- und Entwicklungsarbeiten. Zum vollständigen Verständnis der heutige eingesetzten Verfahren zur Bildcodierung ist daher auch die Grundlagenkenntnis der Theorie der umfangreichen *analogen Bildtechnik* notwendig. Es würde jedoch den Rahmen dieses Kurses bei weitem sprengen, wollte man diese Grundlagen vollständig behandeln. So beschränkt sich dieser Kurs mit der Darstellung einiger weniger theoretischer Zusammenhänge, sofern diese dringend für das Verständnis der digitalen Bildco-

dierung erforderlich sind. Für eine Vertiefung sei auf die einschlägige Literatur verwiesen (z. B. [1.7] und [1.8]).

Inhaltsübersicht

Kapitel 1 befasst sich mit der *visuellen Physiologie*, der *Farbmetrik* und den *Grundlagen zur analogen Videotechnik*. **Kapitel 2** legt die *Grundlagen zur digitalen Bildverarbeitung*; der Übergang von der analogen zur digitalen Bildtechnik mit den üblichen *Formaten* und *Standards* wird dargestellt. In **Kapitel 3** wird die *digitale Quellencodierung für Stillbilder* behandelt. Zentraler Bestandteil dieses Kapitels ist der *JPEG*-Standard, der als quasi Ur-Vater vieler aktueller Codieralgorithmen der Bildtechnik gilt. Die weiter entwickelten Systeme, die auf der Wavelettransformation basieren (*JPEG 2000*) und die fraktale Bildcodierung sind ebenfalls Gegenstand dieses Kapitels. Die Prinzipien der digitalen Quellencodierung für Bewegtbilder bilden den Schwerpunkt des **Kapitel 4**. Als zentrale Verfahren werden *Motion-JPEG*, die *DV-Codierung* und die *prädiktive Bildcodierung* mit und ohne Bewegungskompensation vorgestellt. Die wichtigen Standards der digitalen Videocodierung sind Gegenstand des **Kapitel 5**. Vorgestellt werden unter anderem *ITU-T H.261*, *MPEG-1* und *MPEG-2*. Schwerpunkt des **Kapitel 6** ist die Beschreibung von Videostandards in ausgewählten Anwendungen. Hierzu werden gerechnet die Videocodierverfahren für die optischen Medien CD und DVD sowie die digitale Bildübertragung über TV-Broadcastnetze (*DVB-Verfahren*). *Verbesserte Codiersysteme für die Bildkommunikation* finden sich im abschließenden **Kapitel 7**. Hier werden die Grundideen der komplexen Standards *H.263*, *MPEG-4* und *H.264/Advanced Video Coding* (*AVC*) vorgestellt.

1.2 Physikalische und psychophysische Grundlagen – Was ist *sichtbar*?

Vorwiegend den Forschungsaktivitäten in der Physiologie, in der Psychologie und in der Medizin ist es zu verdanken, dass auf dem komplexen Gebiet der *visuellen Wahrnehmung* in den letzten 40 Jahren umfangreiche Erkenntnisse gesammelt werden konnten. Die interdisziplinäre Nutzung dieser Ergebnisse für die digitale Bildcodierung führte zu entscheidenden Fortschritten, insbesondere auf dem Gebiet der *Datenkompression*. Nachfolgend werden die physikalischen Grundlagen und die wichtigsten Erkenntnisse aus der visuellen Physiologie zusammengefasst dargestellt.

1.2.1 Sichtbarer Bereich der optischen Strahlung

Licht als elektromagnetische Strahlung

Im nachfolgenden soll unter *Licht* der Teil einer *optischen Strahlung* verstanden werden, der für das menschliche visuelle System wahrnehmbar ist. Die optische Strahlung wiederum kann physikalisch als elektromagnetische Welle beschrieben werden, in der sich die elektrische und magnetische Feldstärken periodisch ändern. Die Ausbreitungsgeschwindigkeit *c* einer solchen Welle ist im Vakuum gleich der Lichtgeschwindigkeit c_o:

$$c = c_o = 2,998 \cdot 10^8 \ \frac{m}{s}. \qquad \text{1.2-1}$$

Die Welle wiederholt sich periodisch in Raum und Zeit, so dass eine Periodendauer T definiert werden kann, welche die Anzahl der zeitlichen Perioden pro Sekunde als Frequenz f bzw. als Kreisfrequenz ω angibt:

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \qquad 1.2\text{-}2$$

Die räumliche Periodizität ist durch die Wellenlänge λ vorgegeben. Der formale Zusammenhang zwischen Wellenlänge λ und Frequenz f ist gegeben durch die Ausbreitungsgeschwindigkeit c:

$$c = \lambda \cdot f. \qquad 1.2\text{-}3$$

Sichtbares Licht

Der für das menschliche Auge wahrnehmbare Bereich des elektromagnetischen Wellenspektrums belegt nur einen relativ geringen Teil des Frequenzspektrums. Die zugehörigen Wellenlängen erstrecken sich etwa über den Bereich zwischen 380 und 780 nm. Das Farbtonempfinden des Auges hängt ausschließlich von der Wellenlänge ab. So erscheint Licht mit der Wellenlänge von 400 nm blau-violett, 700 nm rot, alle weiteren Regenbogenfarben liegen dazwischen. Abb. 1.2-1 zeigt das Spektrum der elektromagnetischer Wellen. Besonders hervorgehoben ist der Bereich des sichtbaren Lichts.

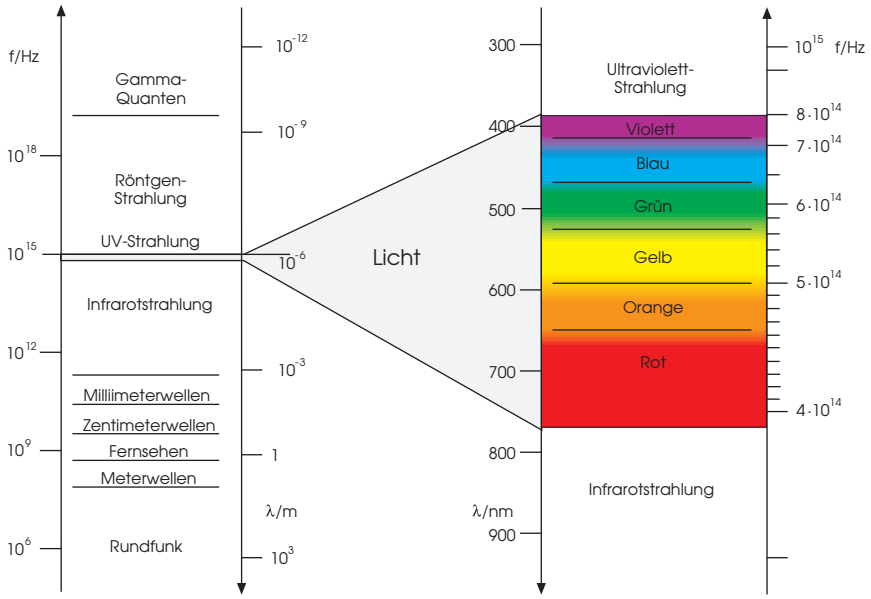


Abb. 1.2-1: Spektrum elektromagnetischer Wellen und sichtbarer Wellenlängenbereich

1.2.2 Die Helligkeitsempfindung

Hellempfindlichkeitsgrad $V(\lambda)$

Während die Frequenz einer elektromagnetischen Wellen eindeutig den *Farbton* des Lichts bestimmt, dienen die so genannten *fotometrische Größen* zur Messung der *Lichtintensität* und der *Helligkeit*. Dazu werden *lichttechnische Größen*, welche die energetischen Anteile des Lichts repräsentieren mit dem *spektralen Hellempfindlichkeitsgrad* $V(\lambda)$ des Auges gewichtet (vgl. [1.2]). Der relative spektrale Hellempfindlichkeitsgrad gibt die Frequenzabhängigkeit bezüglich der Augenempfindlichkeit an. Ein hell adaptiertes Auge („*Tagessehen*“) ist für den spektralen Bereich um 555 nm (*grün*) maximal empfindlich. Bei einem dunkel adaptierten Auge („*Nachtsehen*“) verschiebt sich dieses Maximum Richtung *blau* auf etwa 505 nm. Abb. 1.2-2. zeigt die spektrale Hellempfindlichkeit des menschlichen Auges.

Sehzellentypen

Der Grund für diese Verschiebung liegt in der unterschiedlichen Aktivität der beiden Sehzellentypen, der *Stäbchen* und der *Zapfen*, für dunkle und für helle Lichtumgebungen. Die Stäbchen besitzen eine größere Lichtempfindlichkeit und sind daher in dunkleren Umgebungen mehr aktiv. Für sie ist die Kurve für das dunkel adaptierte Auge charakteristisch (*grün*). Die Zapfen sind in den hellen Beleuchtungssituationen aktiv und bestimmen somit diese spektrale Hellempfindlichkeitskurve (*rot*). Zudem sind sie auch für die Farbunterscheidungsfähigkeit verantwortlich.

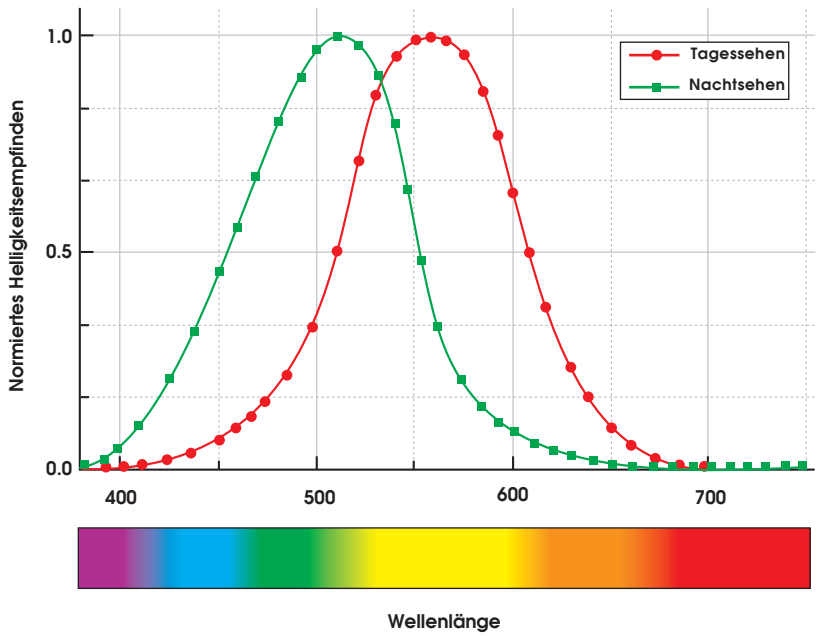


Abb. 1.2-2: Spektrale Hellempfindlichkeit des menschlichen Auges

Helligkeitsadaption

Die Hellempfindung ist also vom *Adaptionszustand* des Auges und der *Umfeldbeleuchtung* bestimmt. Die besondere Leistung des Auges besteht darin sich an Leuchtdichten anzupassen, die einen Bereich von 11 Zehnerpotenzen umfassen. Allerdings ist die Adaption mit verschiedenen Zeitkonstanten bis hin zu 30 Minuten verbunden [1.2]. Bei adaptiertem Auge können immerhin noch etwa 200 Helligkeitswerte (*Graustufen* oder *Luminanzwerte*) unterschieden werden.

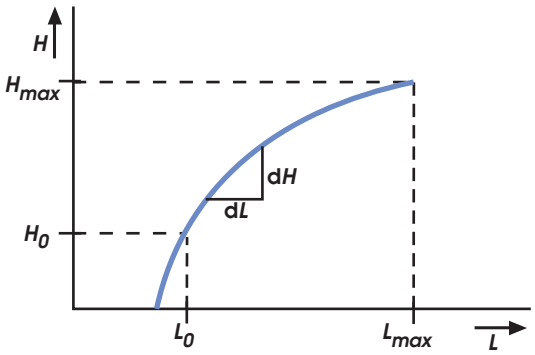


Abb. 1.2-3: Weber-Fechnerisches Gesetz

Weber-Fechnersches Gesetz

Der zuvor genannte Zusammenhang zwischen subjektiv wahrgenommener Helligkeit *H* und der relativer Leuchtdichte *L* ist im mittleren Leuchtdichtebereich *logarithmisch* (vgl. Abb. 1.2-3).

Dieser als *Weber-Fechnersches Gesetz* bekannte Zusammenhang lautet formal:

$$H = log(a \cdot L), \qquad 1.2\text{-}4$$

wobei *a* eine Konstante ist.

1.2.3 Die Ortsauflösung und Sehschärfe

Bereich der größten Schärfewahrnehmung

Der Gesichtssinn umfasst horizontal zwar insgesamt einen Winkel von etwa 180°, dennoch können innerhalb dieses Bereiches nicht alle Lichtreize gleichzeitig und in gleicher Auflösung vom Auge wahrgenommen werden. Tatsächlich ist der Bereich, der „auf einen Blick“ erfassbar und scharf wahrnehmbar ist, nur etwa 3° groß. Dieser Bereich korrespondiert mit dem überwiegenden Teil des Sehzellentyps „*Zapfen*“ auf der Netzhaut. Soll ein größerer Bildbereich vom Auge betrachtet werden, muss dieser mit zeitlich sequenziellen Augenbewegungen überstrichen und kognitiv zu einem Gesamtbild zusammengesetzt werden (vgl. [1.9]).

Laterale Hemmung

Die Trennbarkeit von Linien bzw. die Kontrastempfindlichkeit des Auges hängt vom *Modulationsgrad* und von der *Beleuchtungsstärke* ab. Die Empfindlichkeit ist bei 10 Linien pro Winkelgrad am höchsten und sinkt sowohl zu niedrigen als auch zu hohen Ortsfrequenzen hin ab. Insgesamt besitzt das Auge bezüglich seiner Ortsauflösung eine *Bandpasscharakteristik*. Diese Eigenschaft ist u. a. auf die so genannte *lateralen Hemmung* zurückzuführen: Wird eine Sehzelle durch einen Lichtreiz aktiv, hemmt diese Aktivität gleichzeitig die Aktivität unmittelbar benachbarter Sinneszellen. Abb. 1.2-4 skizziert diesen Zusammenhang.

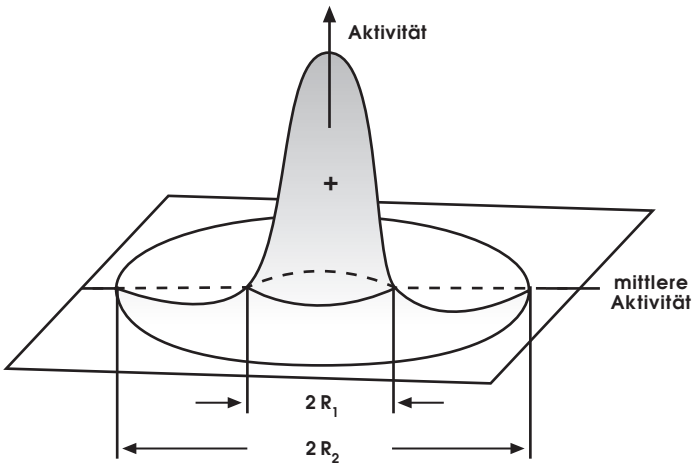


Abb. 1.2-4: Reizung im rezeptiven Feld einer Nervenfaser (nach [1.2])

Mach-Effekt

Die laterale Hemmung hat interessante und gleichzeitig wichtige Auswirkungen auf die menschliche Sehleistung. So verursacht sie den *Mach-Effekt*. Dieser Effekt ist besonders gut am so genannten *Hermanngitter* zu beobachten (vgl. Abb. 1.2-5).

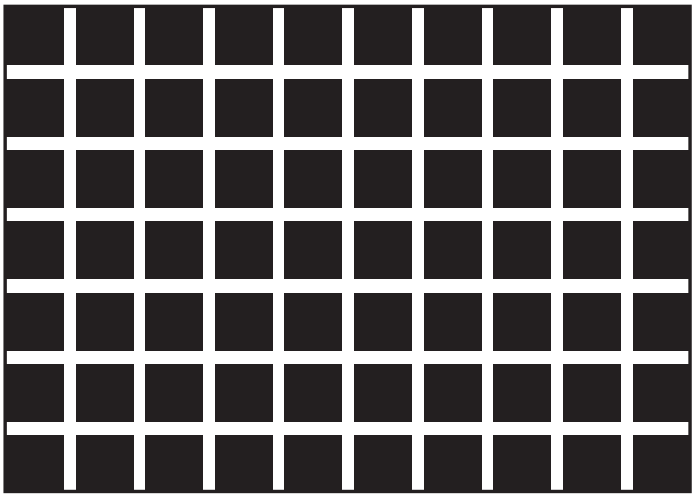


Abb. 1.2-5: Hermanngitter zur Verdeutlichung des Mach-Effekts

Hermanngitter

An den Kreuzungspunkten tauchen – quasi geisterhaft – scheinbare Verdunkelungen auf. Der Mach-Effekt ist auch an Objektkanten zu beobachten und führt in der Regel subjektiv zu einer *Kontrasterhöhung im Kantenbereich*. Das Ergebnis ist eine visuell *verbesserte Objektextraktion* auch bei wenig ausgeprägten Kontrastverhältnissen. Der für die Bildcodierung nutzbare Nebeneffekt ist die relative Unempfindlichkeit gegenüber leichten Verfälschungen im Kantenbereich selber. Leichte örtliche Verformungen sowie geringe Helligkeitsschwankungen werden hier maskiert oder zumindest wenig störend empfunden.

Anwendung

Die hier beschriebenen örtlichen Auflösungseigenschaften des Gesichtssinns werden besonders effektiv in Codierverfahren genutzt, die mit einer Transformationscodierung arbeiten. In Kapitel 4 werden solche Transformationscodierungen vorgestellt. Der weit verbreitete *JPEG*-Algorithmus ist ein Beispiel hierfür.

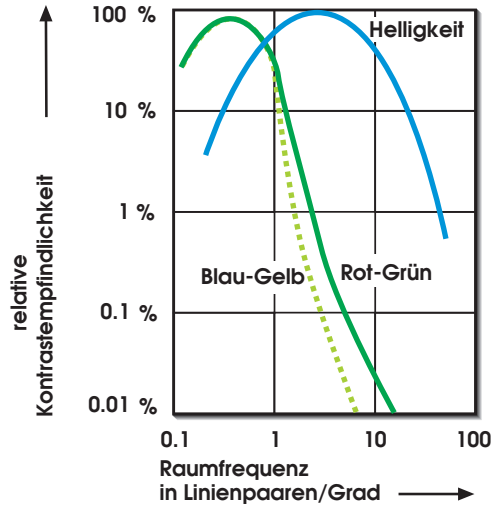


Abb. 1.2-6: Kontrastempfindlichkeit des visuellen Systems für farbart- und helligkeitsmodulierte räumliche Gitter (nach [1.11])

Kontrastempfindlichkeit Farbe und Helligkeit

Eine interessante visuelle Eigenschaft, die auch schon für eine Optimierung der analogen Farbbildcodierung genutzt wurde, ist die Tatsache, dass es unterschiedliche Kontrastempfindlichkeiten für farbartmodulierte und für rein helligkeitsmodulierte Gitter gibt. In [1.11] wurde dieser Sachverhalt intensiv untersucht.

In Abb. 1.2-6 ist das Ergebnis dargestellt. Es wird deutlich, dass die Kontrastempfindlichkeit für Farbartübergänge zu höheren Frequenzen wesentlich früher abfällt als für Helligkeitsübergänge. Somit ist die Auflösung des Gesichtsinns für Farbartübergänge (bei gleicher Helligkeit) wesentlich geringer als für Helligkeitsübergänge, oder anders ausgedrückt: Feine Objektdetails erkennt das menschliche Auge vorwiegend über einen Helligkeitssprung (*Kante*) und weniger über eine Änderung der Farbinformation.

Anwendung

Die unterschiedliche Empfindlichkeit des menschlichen visuellen Systems für Helligkeits- und Farbsprünge kann in der digitalen Bildcodierung vorteilhaft genutzt werden, in dem die Farbinformation örtlich 2 bis 4 mal niedriger aufgelöst codiert wird, als die Helligkeitsinformation. In Kapitel 2 werden diese Systeme vorgestellt, welche mit einer örtlichen Unterabtastung der Farbe arbeiten.

Abschließend sei bemerkt, dass die Auflösungsleistung des visuellen Systems nur für ruhende Bildvorlagen so groß ist, wie hier beschrieben. Soll das menschliche Auge unwillkürlich bewegten Objekten folgen, können diese meist in ihrer Struktur nur unscharf erkannt werden.

1.2.4 Die Zeitauflösung

Ähnlich wie in örtlicher Richtung weist das menschliche visuelle System auch in zeitlicher Richtung eine Bandpasscharakteristik auf. Konkret bedeutet das: Zeitlich sehr langsam wiederholte Sinnesreize werden weniger deutlich wahrgenommen als Sinnesreize, die in einem Frequenzbereich zwischen etwa 5 bis 8 Hz liegen. Höhere Frequenzen werden wieder weniger deutlich wahrgenommen. Dabei sind zeitliche

und örtliche Frequenzwahrnehmungen nicht unabhängig voneinander. Die Abb. 1.2-7 und Abb. 1.2-8 zeigen die empirisch aufgenommene örtlich-zeitliche *Schwellenempfindlichkeit* nach einer Messung aus [1.10].

Frequenz für eine Bewegungsverschmelzung

Für die Bewegtbildübertragung stellt sich weiterhin die wichtige Frage, wie viele Bilder pro Sekunde übertragen werden müssen, damit der Gesichtssinn einen in den Bildern dargestellten Bewegungsablauf als kontinuierlich zusammenhängend empfindet (*Bewegungsverschmelzung*). Untersuchungen haben gezeigt, dass ein gleichmäßig erscheinender Bewegungsablauf bereits mit einer Rate von etwa 20 Bildern pro Sekunde erzielbar ist [1.9]. Die Filmbildfrequenz beträgt bekanntermaßen 24 Hz. Trotzdem gibt es kein Bildwiedergabegerät, dass mit einer derart niedrigen Frequenz eine Bewegtbilddarstellung realisiert.

Örtlich-zeitliche Schwellenempfindlichkeit

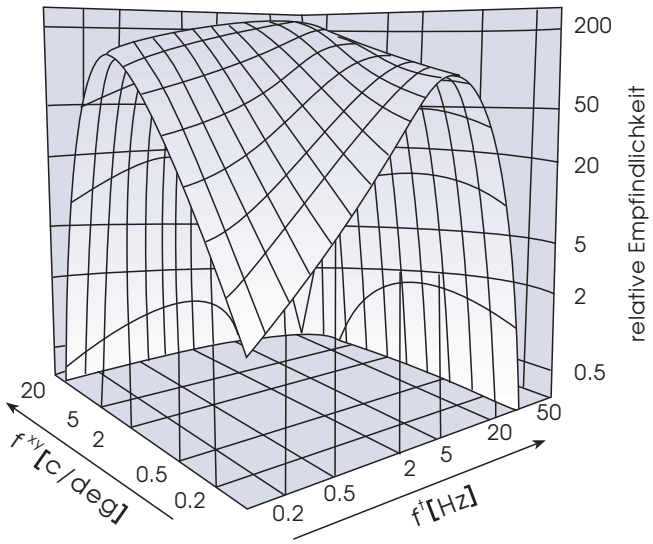


Abb. 1.2-7: Örtlich-zeitliche Schwellenempfindlichkeit – 3D-Darstellung (vgl. [1.9] und [1.10])

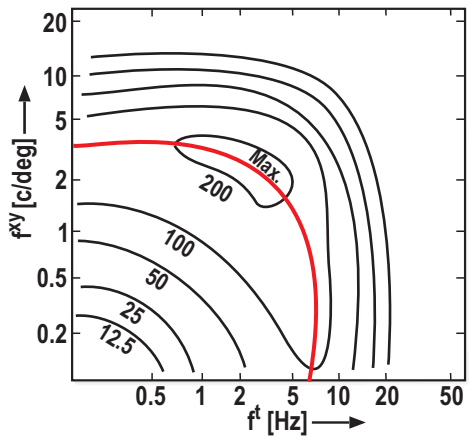


Abb. 1.2-8: Örtlich-zeitliche Schwellenempfindlichkeit – 2D-Querschnitt zur Verdeutlichung der Max-Kurve (vgl. [1.9])

Flimmer-Verschmelzungsfrequenz

Der mit Dunkelpausen verbundene Bildwechsel führt zu einer ständig wiederholten Erregung und Hemmung der Neuronen, was als sehr unangenehmes Flimmern empfunden wird. Dieses Flimmern wird als *Großflächenflimmern* bezeichnet, da es alle

Bildpunkte zugleich betrifft. Die so genannte *Flimmer-Verschmelzungsfrequenz*, bei der diese Flimmerempfindung verschwindet, liegt oberhalb von 50 Hz und steigt mit der Leuchtdichte der Bilddarstellung und entscheidend auch mit dem Betrachtungswinkel: Große Bilddarstellungen werden bei sonst gleichen Parametern eher als flimmernd wahrgenommen als kleine. Bei Wiedergabesystemen, bei denen das Bild einen großen Sehwinkel überstreicht (z. B. Computerdisplays), sollte die Bildwiederholfrequenz zwischen 70 Hz und 100 Hz liegen.

Dies ist der Grund, warum Kinofilmbilder, welche mit 24 Hz aufgenommen wurden, bei der Wiedergabe zweimal (oder dreimal) projiziert werden, so dass sich eine Verdopplung (bzw. Verdreifachung) der Dunkelpausen bei einer Flimmerfrequenz von 48 Hz (bzw. 72 Hz) ergibt. Mit dieser Maßnahme reduziert sich zwar deutlich der Flimmereindruck, doch kann es bei Bewegungen (bewegte Objekte, Kameraschwenk etc.) zu deutlich wahrnehmbaren Ruckelstörungen kommen, da die örtliche Position von bewegten Objekten bei dieser Darstellungsweise in den wiederholten Zwischenprojektionen der Erwartung des visuellen Systems in Bezug auf eine gleichmäßige Bewegung widerspricht.

Abb. 1.2-9 skizziert diesen Zusammenhang an Hand einer gleichmäßigen translatorischen Bewegung.

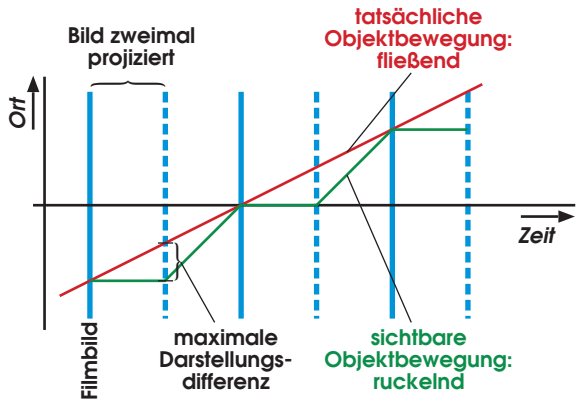


Abb. 1.2-9: Zur Entstehung von Ruckelstörungen bei der Kinoprojektion mit Bildwiederholung

Deutlich zu erkennen ist, dass die sichtbare Objektbewegung (grün) eine ungleichmäßigen Verlauf zeigt. Die maximale örtliche Differenz zwischen tatsächlicher Objektposition und dargestellter Position geschieht zum Zeitpunkt der Bildwiederholung durch eine erneute Projektion des Filmbildes.

Korrelation Ruckeln und Bildwiederholung

Wird das Filmbild zweimal wiederholt, entsteht eine Bildfolgefrequenz von 72 Hz. Das korrespondierende Bildflimmern reduziert sich damit weiter. Allerdings erhöht sich auch die maximale örtliche Differenz zwischen tatsächlicher Objektposition und dargestellter Position zum Zeitpunkt der zweiten Filmbildwiederholung. Als Konsequenz daraus verstärkt sich der Ruckeleindruck bei Objekt- und Kamerabewegungen. In der Praxis wird daher als Kompromiss aus Ruckeln und Flimmern bei der Kinobilddarstellung nur eine einfache Bildwiederholung angewendet.

Selbsttestaufgabe 1.2-1:

*Geben Sie den ungefähren **Wellenlängenbereich** für die sichtbare optische Strahlung an!*

Selbsttestaufgabe 1.2-2:

Wie heißen die beiden Sensorentypen der menschlichen Netzhaut? Welche Aufgaben kommen welchen Sensoren zu?

Selbsttestaufgabe 1.2-3:

Beschreibung und erklären Sie kurz den Mach-Effekt!

Selbsttestaufgabe 1.2-4:

Erklären Sie den Unterschied zwischen Bewegungsverschmelzungsfrequenz und Flimmerverschmelzungsfrequenz!

Selbsttestaufgabe 1.2-5:

Mit welcher Maßnahme lässt sich eine flimmerfreie Bilddarstellung beim Kinofilm realisieren? Welchen gravierenden Nachteil muss man dabei in Kauf genommen werden?

Selbsttestaufgabe 1.2-6:

Wie verhält sich das örtliche Auflösungsvermögen des menschlichen Auges in Bezug auf Helligkeitsübergänge im Vergleich zum Auflösungsvermögen bei Farbübergängen?

1.3 Einführung in die Farbmetrik – Was ist „*Farbe*“

In den nachfolgenden Abschnitten wird auf der Basis der farbmetrischen Eigenschaften des menschlichen Sehorgans die allgemeine *Farbtheorie* und *Farbenlehre* hergeleitet.

Wie bereits inAbschnitt 1.2.2 ausgeführt wurde, besitzt der menschliche Gesichtssinn zwei Typen von lichtempfindlichen Organen: Die *Stäbchen* und die *Zapfen*.

Zapfen

Die *Zapfen* haben ihre größte Dichte auf der Netzhaut innerhalb eines zentralen Bereiches (*Fovea Centralis*) von etwa 3° (bezogen auf den horizontalen und vertikalen Blickwinkel des Sehorgans). Diese Sinnesorgane sind zum einen verantwortlich für die Hellempfindlichkeitskurve „*Tagessehen*“ (vgl. Abb. 1.2-2) und zum anderen ermöglichen sie die Wahrnehmung von Farben. Dazu stellte bereits 1807 der englische Mediziner und Physiker Thomas Young in seinen „*Lectures of Natural Philosophy*“ die Hypothese auf, dass es im Auge drei Typen von Farbsehzellen geben muss. In der zweiten Hälfte des 19. Jahrhunderts baute der Physiologe und Physiker Hermann von Helmholtz die Theorie in seiner *Dreikomponententheorie* aus. Tatsächlich gibt es drei unterschiedliche Zapfenarten, die sich hinsichtlich der Zapfensehstoffe unterscheiden. So reagieren diese Zapfenarten auf die einfallende Lichtstrahlung im Spektralbereich von *blau* (etwa 400 bis 500 nm), *grün* (etwa 500 bis 600 nm) und *rot* (etwa 600 bis 700 nm) jeweils unterschiedlich. Über eine Mischung der drei Teilempfindungen entsteht der Farbeindruck. Abb. 1.3-1 zeigt die Absorptionsspektren der drei Zapfenarten.

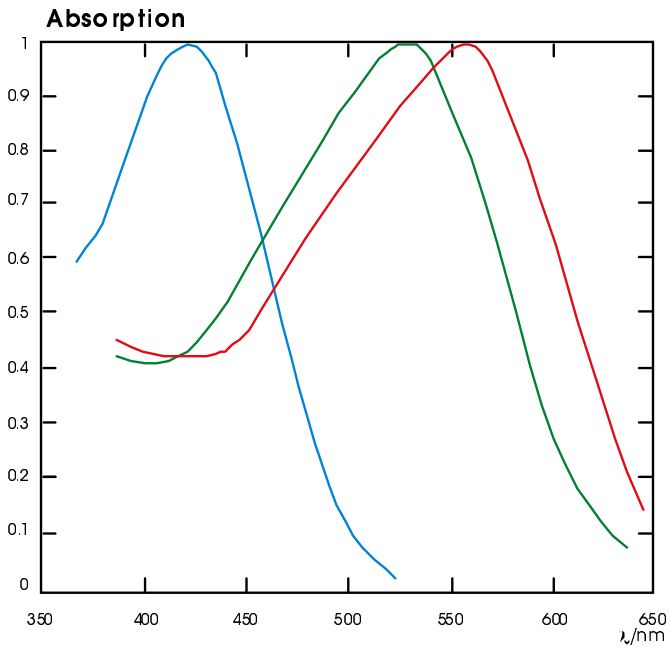


Abb. 1.3-1: Absorptionsspektren der drei Zapfenarten beim menschlichen Auge nach [1.5]

Stäbchen

Im Gegensatz zu den Zapfen, gibt es von den *Stäbchen* nur eine Art. Diese Zellenart besitzt ebenfalls einen Sehstoff, nämlich *Rhodopsin*, der bei der Absorption von

Licht ausgebleicht wird. So ist für diese Sehzellen eine Helladaption über einen großen Leuchtdichtebereich möglich. Insgesamt können diese Sehzellen noch bei wesentlich geringeren Leuchtdichteverhältnissen Helligkeiten differenzieren als die Zapfen, obschon sie dabei keine Farbunterscheidungsfähigkeit besitzen („*Bei Nacht sind alle Katzen grau.*“). Die spektrale Empfindlichkeitskurve für das „*Nachtsehen*“ aus Abb. 1.2-2 beruht im Wesentlichen auf den zuvor genannten Eigenschaften der Stäbchen.

Lange vor den Erkenntnissen über die Eigenschaften des menschlichen Auges war experimentell erprobt, dass sich die verschiedenen Farbeindrücke aus drei geeigneten Grundfarben (z. B. *rot*, *grün* und *blau*) ermischen lassen (z. B. bei der Malerei). Es zeigt sich aber, dass sich aus den Grundfarben nach zwei unterschiedlichen Mischtechniken jeweils sehr verschiedene Mischfarben erzielen lassen.

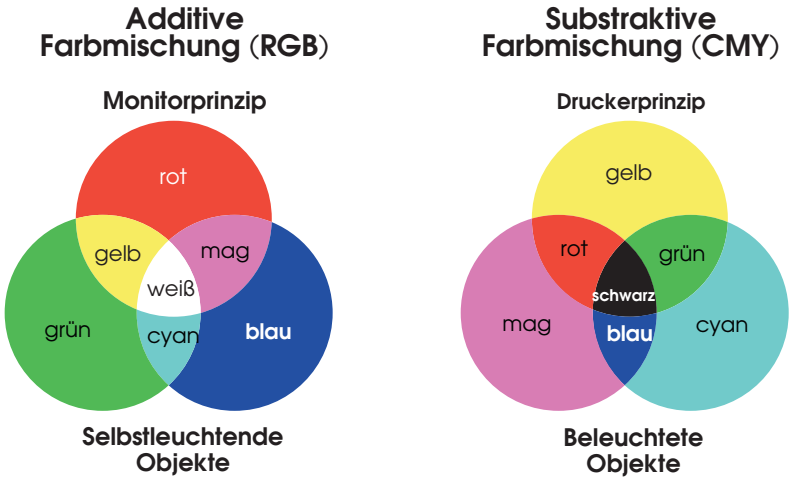
Subtraktive Farbmischung

Ein Beispiel für die so genannte *subtraktive Farbmischung* ist das Zusammenführen verschiedenfarbiger Farbstoffe, wie es bei der Malerei geschieht. Dieses Mischen von übereinander liegenden Farbstoffen kann man als das Zusammenfügen von Farbfiltern mit unterschiedlichen farblichen Absorptionseigenschaften verstehen. So erhält man beispielsweise *grünliches* Licht, wenn man vor eine *weiße* Lichtquelle ein *blaues* und dahinter ein *gelbes* Filter setzt. Der resultierende Farbeindruck entsteht also durch die nach der optischen Filterung verbleibenden Farbanteile einer ursprünglich weißen Lichtquelle. Dazu wird angenommen, dass die weiße Lichtquelle eine weitgehend kontinuierliche Energieverteilung über das ganze sichtbare Spektrum besitzt. Diese Annahme ist in der Regel für natürliche Lichtquellen gegeben. Die subtraktive Farbmischung wird vielfach auch bei Farbdruckern verwendet, die mit drei oder mehr Grundfarben arbeiten. Als Grundfarben werden dann aber oft *gelb*, *magenta*, *cyan* und ggf. weitere verwendet.

Additive Farbmischung

Dazu entsteht im Gegensatz bei der *additiven Farbmischung* der resultierende Farbeindruck, in dem kleine, dicht beieinander liegende, selbst leuchtende Farbpunkte aus so großen Entfernung betrachtet werden, dass die einzelnen Farbpunkte nicht mehr separat aufgelöst werden können, sondern ineinander *verschmelzen*. Ebenso ist die additive Mischung auch durch die Projektion von einem roten, grünen und blauen Bildauszug auf eine gemeinsame Bildwand möglich. Das Wesentliche der additiven Mischung liegt darin, dass die Farbanteile der verschiedenen Farben additiv von den Sehzellen des menschlichen Auges wahrgenommen werden. Die additive Mischung funktioniert, solange eine Trennung der einzelnen Farbanteile vom menschlichen Auge nicht in örtlicher oder zeitlicher Richtung möglich ist. In dem zuvor genannten Beispiel mit den kleinen Farbpunkten ist das dann der Fall, wenn ein hinreichend großer Betrachtungsabstand gegeben ist. Die zeitliche Trägheit des menschlichen Auges (vgl. auch Abb. 1.2-7 und Abb. 1.2-8) ermöglicht auch eine additive Farbmischung ohne dass die Farbanteile gleichzeitig vorhanden sein müssen. Vielmehr kann die Farbdarstellung auch gelingen, wenn zeitlich sequenziell gezeigte Farbblitze so schnell auftreten, dass sie vom Auge hinreichend gut zu einem Gesamtfarbeindruck integriert werden. Eine Reihe von Farbbildprojektoren arbeiten nach diesem Prinzip (*Farbkreisel*).

Animation 1.3-2 zeigt die beiden Farbmischverfahren auf der Basis von *rot*, *grün* und *blau* (*Additive Farbmischung*) und *gelb*, *magenta* und *cyan* (*Subtraktive Farbmischung*).



Animation 1.3-2: Additive und subtraktive Farbmischung

Die additive Farbmischung findet bei fast allen elektronischen Bilddarstellungen in der analogen und digitalen Bildtechnik Anwendung. Auf dieser Basis werden auch die nachfolgenden technischen Farbbilddarstellungen vorgestellt. In Anlehnung an die Farbphysionomie des menschlichen Auges kann man für den Farbmonitor (hier: Kathodenstrahlmonitor) primäre Farbphosphore *rot* (*R*), *grün* (*G*) und *blau* (*B*) definieren. Bei der Herstellung solcher Bildschirmphosphore und bei der Farbmischung ist man jedoch einigen Beschränkungen unterworfen. So lassen sich nicht in jedem Fall zur natürlichen Bildvorlage spektral identische Farbdarstellungen mit dem Monitor erzielen. Eine mögliche Lösung dieses Problems besteht darin, so genannte *metamere Farben* auf dem Monitor zu generieren. Dieses sind Farbmischungen, die für das menschliche Auge zwar gleich aussehend sind, aber eventuell eine andere spektrale Zusammensetzung besitzen.

Graßmannsche Gesetze

Dieses führt uns zu den drei *Graßmannschen Gesetzen* der Farbmetrik:

1. *Für das Ergebnis einer additiven Mischung ist nur das Aussehen nicht die spektrale Zusammensetzung der Komponenten maßgeblich.*

Der Satz drückt aus, dass in additiven Farbmischungen die Komponenten durch ihre *Valenzen* bestimmt sind während ihre spektrale Zusammensetzung keine Rolle spielen. Man kann jede Mischungskomponente durch eine metamere Komponente gleicher Valenzen ersetzen, ohne dass sich die Valenz der gesamten Mischung ändert. Mischungen aus gleich aussehenden Farben ergeben dabei wieder gleich aussehende Farben.

2. *Alle Farbmischungen verlaufen stetig. Die Stetigkeit von Mischungen drückt sich in der Stetigkeit der Spektralwertkurven aus.*

3. *Zur Festlegung einer Farbe sind 3 Bestimmungsstücke notwendig und hinreichend.*

Diese Feststellung drückt die Erfahrungstatsache aus, dass drei Farbkomponenten *rot* (*R*), *grün* (*G*) und *blau* (*B*) notwendig sind (aber auch ausreichen), um mit einer additiven (äußeren) Mischung alle übrigen Farben zu ermischen. Als alternative Bestimmungsstücke bietet sich auch das Tripel aus *Helligkeit*, *Farbton* und *Farbsättigung* an.

Bei der Einführung eines technischen Farbkoordinatensystems gibt es verschiedene Möglichkeiten. Sie alle müssen grundsätzlich den Graßmannschen Gesetzen genügen. Von daher sind viele Farbkoordinatensysteme dreidimensional (z. B. der *RGB*-Farbwürfel-Raum, vgl. [1.5]). Leider sind die 3D-Koordinatensysteme recht unübersichtlich und wenig anschaulich. Um Aussagen über die „Farbe“ – bestehend aus Farbton und Farbsättigung – machen zu können reicht oft auch eine 2D-Darstellung, die dann das dritte Bestimmungsstück – die Helligkeit – unberücksichtigt lässt. Dies ist der Grund, warum die meisten standardisierten Farbkoordinatensysteme nur zweidimensional ausgelegt sind.

Farbdreieck, Farbtafel Begriffe

Ein einfaches Beispiel aus der Bildtechnik ist das so genannte*Farbdreieck* oder die *Farbtafel*. Hierin sind folgende Begriffe definiert:

- Als *Farbwerte* sind die Bildschirmprimärvalenzen *R*, *G* und *B* definiert.
- Eine *Farbvalenz* (*F*) ist festgelegt durch: (*F*) = *R*(*R*) + *G*(*G*) + *B*(*B*).
- *Helligkeit* $(F)_{\text{grau}} = a\,[\,1(R) + 1(G) + 1(B)\,]$.
- Die *Farbart* ist definiert durch den *Farbton* und die *Farbsättigung*.
- Die *Farbwertanteile* sind definiert durch

$$r = \frac{R}{R+G+B};\; g = \frac{G}{R+G+B};\; b = \frac{B}{R+G+B} \qquad \text{1.3-1}$$

Unbuntpunkt

Für das Farbdreieck wird ein gleichseitiges Dreieck mit den Bildschirmprimärvalenzen *R*, *G* und *B* als Eckpunkte angesetzt (vgl. Abb. 1.3-3). Der Mittelpunkt des Dreiecks stellt den so genannten *Unbuntpunkt* dar, ein Punkt, an dem Farbton und Farbsättigung gleich Null sind.

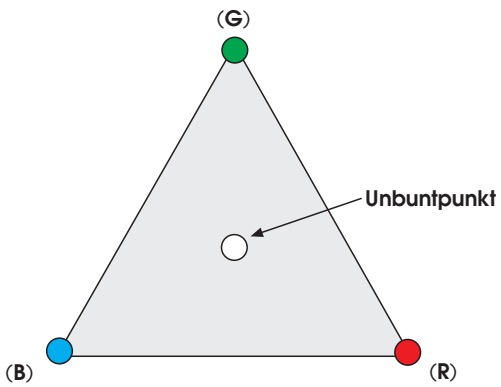


Abb. 1.3-3: Farbtafel

In der Entwicklungsphase des analogen Farbfernsehens wurde die Definition der Bildschirmprimärvalenzen auf der Basis von herstellbaren Farbphosphoren für die Kathodenstrahlröhre definiert. Für Europa wurden die so genannten *EBU-Phosphore* (EBU Tech. 3213 [1.12]), für die USA die *Phosphore* nach der *SMPTE*-Norm RP 145 [1.13] festgelegt. Diese beiden Farbsysteme unterscheiden sich geringfügig voneinander. Beiden ist aber eines gemeinsam: Es sind grundsätzlich nicht alle in der Natur vorkommenden Farbarten (Farbton, Farbsättigung) exakt darstellbar. Die mangelnde Darstellungsfähigkeit beruht insbesondere auf der beschränkten Möglichkeit der zugrunde gelegten Bildschirmphosphore, hohe Farbsättigungen für alle Farbtonbereiche gleichermaßen gut wiedergeben zu können.

Metamere Farbmetrik

Mit Hilfe der Gesetze der *metameren Farbmetrik* lassen sich allerdings dreidimensionale Farbkoordinatensysteme konstruieren, mit Hilfe dessen sich jeder in der Natur vorkommende spektrale Verteilungsverlauf gleich aussehend – metamer – aus einer Linearkombination der drei spektralen Verteilungskurven nachbilden lässt, die dieses Farbkoordinatensystem aufspannen. Die so genannte *CIE-Normfarbtafel* wird zunächst durch drei spektrale Verteilungsdichten $X(\lambda)$, $Y(\lambda)$ und $Z(\lambda)$ gebildet.

Die Besonderheiten dieses Farbkoordinatensystems sind:

- Die so genannten *Normspektralwertkurven* $X(\lambda)$, $Y(\lambda)$ und $Z(\lambda)$ sind überall positiv.
- Die $Y(\lambda)$-Kurve stimmt mit der Hellempfindlichkeitsverteilung des menschlichen Gesichtssinns für das Tagsehen überein (vgl. Abb. 1.2-2). Damit ist die $Y(\lambda)$-Kurve bei einer Wellenlänge von 555 nm exakt auf den Wert 1 normiert.

Als weitere Besonderheit muss beachtet werden, dass die $X(\lambda)$-Kurve zwei Maxima besitzt. Abb. 1.3-4 zeigt die Normspektralwertkurven $X(\lambda)$, $Y(\lambda)$ und $Z(\lambda)$ der *CIE-Normfarbtafel* von 1931 für einen 2°-Normalbeobachter.

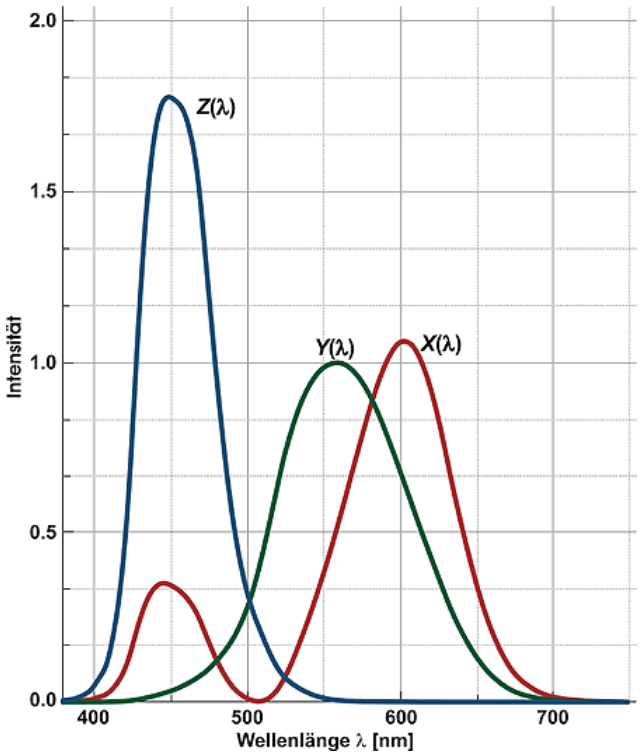


Abb. 1.3-4: Normenspektralwertkurven für den farbmetrischen Normalbeobachter (2°)

Als Koordinaten im rechtwinkligen Farbdreieck *X*, *Y*, *Z* lassen sich die Farbwertanteile *x*, *y* und *z* in Analogie zu Gl. 1.3-1 wie folgt berechnen:

$$x = \frac{X}{X+Y+Z};\; y = \frac{Y}{X+Y+Z};\; z = 1 - x - y \qquad \text{1.3-2}$$

Da alle Spektralwertkurven nur positive Anteile besitzen, die die so genannte *Spektralwertkurve*, die alle Farbtöne mit der in der Natur maximal möglichen Sättigung zeigt, überall mit positiven Anteilen beschreibbar. Innerhalb dieser Kurve liegen dann alle möglichen Farbarten, auch die, die ein Monitor darstellen kann. Für die Farb-Transformation zwischen dem *RGB*-Farbwürfel-Raum und der CIE-Normfarbtafel werden die folgenden Transformationsgleichungen verwendet:

EBU-Phosphore in der CIE-Normfarbtafel

$$\begin{aligned} x &= 0.640R + 0.290G + 0.150B \\ y &= 0.330R + 0.600G + 0.060B \\ z &= 0.030R + 0.110G + 0.790B \end{aligned} \qquad \text{1.3-3}$$

$$\begin{aligned} x &= 0.630R + 0.310G + 0.155B \\ y &= 0.340R + 0.595G + 0.070B \\ z &= 0.030R + 0.095G + 0.775B \end{aligned} \qquad \text{1.3-4}$$

SMPTE-Phosphore in der CIE-Normfarbtafel

Abb. 1.3-5 zeigt die *CIE-Normfarbtafel* mit den eingezeichneten Koordinaten der Primärvalenzen nach *SMPTE*- und *EBU-Norm*. Die beiden ursprünglich für das analoge Farbfernsehen definierten Farbdreiecke der EBU (Europa) und der SMPTE

(USA) sind relativ ähnlich. Weiterhin ist der Zug der Spektralfarben (*Spektralfarbenzug*) eingezeichnet. Er ist mit den Wellenlängen (in [nm]) parametrisiert. Der Spektralfarbenzug wird durch die Purpurgerade geschlossen und kennzeichnet so alle reellen Farben.

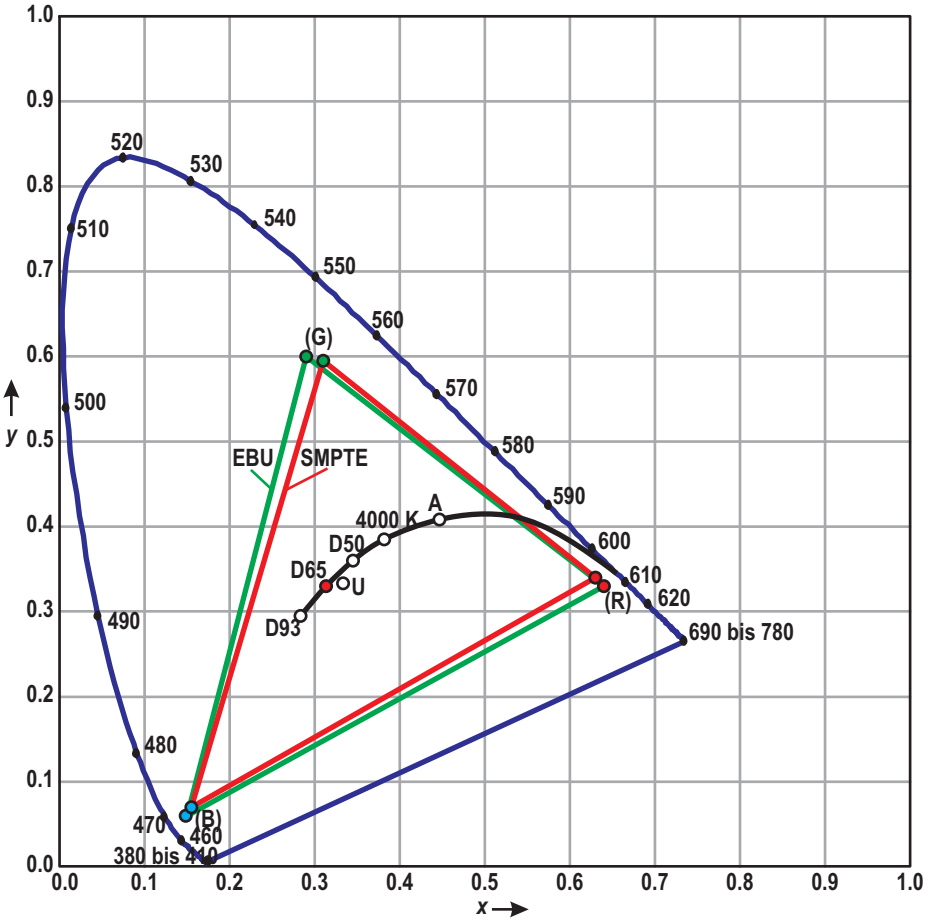


Abb. 1.3-5: CIE-Normfarbtafel

Die *CIE-Normfarbtafel* besitzt im farbmetrischen Sinne einen theoretischen Unbuntpunkt (Weißpunkt) *U* mit den Koordinaten $x = y = z = 1/3$. Jede Gerade, die von diesem Weißpunkt ausgeht, kennzeichnet eine Farbart mit gleichem Farbton aber verschiedener Farbsättigung.

Als *praktischen Unbuntpunkt* wird in der Regel nicht der Weißpunkt *U*, sondern ein Farbort gewählt, der auf der schwarz eingezeichneten Kurve liegt. Diese Kurve gibt die Farborte der so genannten *Planckschen Strahlung*[1] von wichtigen Lichtquellen an. Sie ist in Grad Kelvin [K] skaliert und gibt den Farbort an, den eine plancksche Strahlungsquelle mit der angegeben Temperatur besitzt (*Farbtemperatur*). Einige wichtige Punkte sind mit Buchstaben und Zahlen gekennzeichnet. So wird für die beiden analogen Farbfernsehsysteme der EBU und SMPTE als Weißpunkt der Farbort *D65* festgelegt

Weißpunkt D65

1 Die Begriffe Plancksche Strahlung und Farbtemperatur soll hier nicht näher hergeleitet werden. Weiterführende Hinweise finden sich beispielsweise in [1.3]

$$D65(x, y, z) = (0.3127, 0.3290, 0.3582) \quad \text{1.3-5}$$

Er kennzeichnet eine plancksche Strahlungsquelle mit der Farbtemperatur von 6500 K. Dieses entspricht in etwa den Farbstimmungsverhältnissen für Tageslicht im Außenbereich (Sonnenlicht). Für Kunstlicht im Innenbereich (Glühbirne) liegt die Farbtemperatur etwa bei 2800 bis 3000 K und entspricht so näherungsweise dem Normlicht *A*.

So wie alle reellen Farben, dass heißt Farben, die in der Natur vorkommen, nur innerhalb des vom Spektralfarbenzug umschlossenen Bereiches liegen, lassen sich mit den *EBU*- bzw. *SMPTE-Primärvalenzen* grundsätzlich auch nur Farben darstellen, die innerhalb des jeweiligen Farbdreiecks liegen.

Aus den zuvor genannten Gleichungen lässt sich schließlich auch der *LuminanzanteilY* ermitteln. Die nachfolgende Gleichung gilt insbesondere auch für den digitalen Standards *ITU-R BT.601* (vgl. Kapitel 2):

$$Y = 0,299R + 0,587G + 0,114B. \quad \text{1.3-6}$$

Selbsttestaufgabe 1.3-1:

Erklären Sie den Unterschied zwischen additiver und subtraktiver Farbmischung! Geben Sie jeweils ein technisches Beispiel an!

Selbsttestaufgabe 1.3-2:

Was sind metamere Farben?

Selbsttestaufgabe 1.3-3:

Worin besteht die Einschränkung eines EBU- oder SMPTE-Farbmonitors in Bezug auf die Farbdarstellung?

Selbsttestaufgabe 1.3-4:

Geben Sie die Farbwerte des Unbuntpunktes für EBU- und SMPTE-Farbfernsehsysteme in der CIE-Normfarbtafel an!

1.4 Einführung in die analoge Videotechnik

Wie es anfing

Für die Übertragung von bewegten Bildern („*Video*“) muss das sich ändernde zweidimensionale Abbild einer realen Szene in ein geeignetes elektrisches Signal für die Übertragung umgewandelt werden. Um diesen Vorgang vereinfacht beschreiben zu können sei im folgenden nur die Abbildung und Übertragung der Helligkeitsverteilung der Bildvorlage betrachtet.

Die Bildvorlage wird in horizontale, aneinander angrenzende Zeilen zerlegt. Diese Aufgabe übernimmt ein optisch-elektrischer Wandler, der auch das korrespondierende Signal zeitsequenziell an ein Übertragungssystem weiterreicht.

Empfangsseitig wird, nach entsprechender Aufbereitung, das elektrische Signal einem elektrisch-optischen Wandler, dem Monitor, zugeführt und wieder als ein Abbild der Helligkeitsverteilung der Bildvorlage wiedergegeben.

Bildfeldzerlegung

Der Vorgang der *Bildfeldzerlegungin Zeilen* und die entsprechende Rückwandlung im Empfänger muss zeitsynchron geschehen. Dazu werden im Aufnahmesystem zusätzliche Synchronsignale generiert und ebenfalls zum Empfangssystem übertragen. Abb. 1.4-1 skizziert dieses Prinzip der analogen Fernsehübertragung.

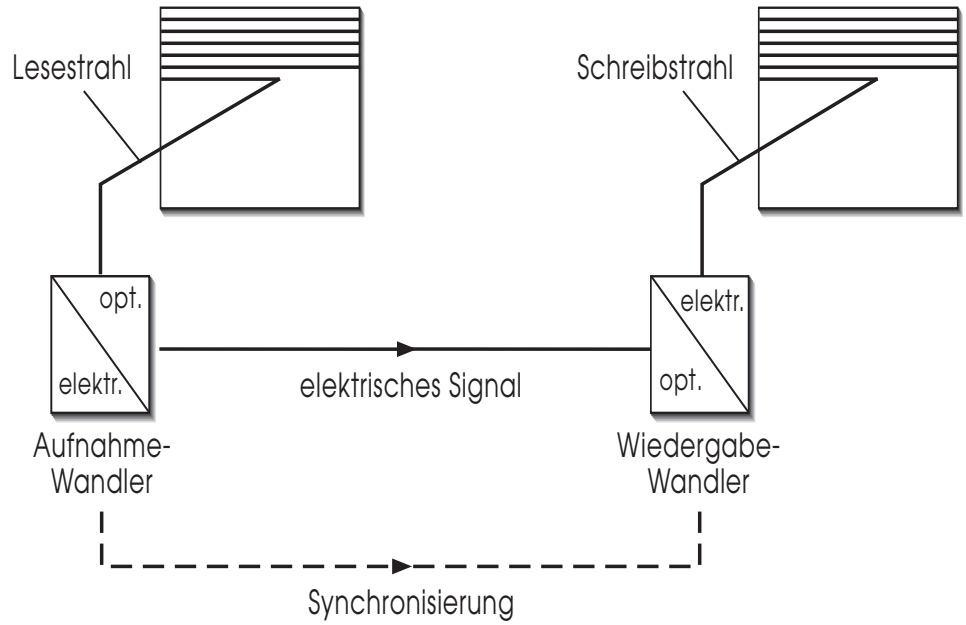


Abb. 1.4-1: Prinzip der Fernsehübertragung

In den Anfängen der analogen Bildübertragung gab es eine Reihe von technischen Randbedingungen, die eingehalten werden mussten. Die hieraus entwickelten Normen sind auch heute noch Standard, wenngleich viele technische Einschränkungen nicht mehr gelten. Aus Gründen der Kompatibilität ist eine Veränderung im Sinne des technischen Fortschritts innerhalb eines Standards nur schwer zu verwirklichen.

Die wichtigsten Eigenschaften der analogen Video- und Fernsehtechnik werden nachfolgend kurz zusammenfassend dargestellt.

1.4.1 Analoge Fernsehabtastung

Ablenksignale für H und V

Das in Abb. 1.4-1 skizzierte Prinzip der Fernsehbildübertragung geht von der Annahme aus, dass eine Art *Lesestrahl* zeilenweise die Bildvorlage abtastet. Tatsächlich hatte dieses Modell des Lesestrahls bei der Einführung des Fernsehens als technische Vorlage die Bildaufnahmeröhre. In modernen Bildwandlern stimmt die Abtastung mit dem Lesestrahl nur insofern, als dass beide Sensoren das gleiche zeitlich abgetastete Zeitsignal erzeugen. Physikalisch haben sich die Verfahren prinzipiell geändert. Trotzdem wollen wir nachfolgend bei diesem Bild des Abtastlesestrahls und des korrespondierenden Schreibstrahls bleiben, um den Abtastvorgang als solchen besser untersuchen zu können.

So tastet der Lesestrahl die Bildvorlage horizontal und vertikal von oben links nach unten rechts ab. Man kann diesen Vorgang mit der Bewegung des Gesichtsfeldes beim Lesen eines Textes vergleichen: Lesen innerhalb der Zeile von links nach rechts mit anschließenden raschen Zurückspringen zum Beginn der folgenden Textzeile. Für die Führung eines solchen Lesestrahls benötigt man eine periodische Ablenkung horizontal (*schneller Sägezahn*) und vertikal (*langsamer Sägezahn*).

Abb. 1.4-2 zeigt die beiden *Ablenksignale*: (horizontal – blau und vertikal – rot). Es ist auch deutlich zu sehen, dass die Rücksprungzeit vom Ende einer Zeile zur nächst folgenden Zeile nicht beliebig kurz sein kann. Das gleiche gilt für den Rücksprung am Ende der letzten Zeile eines Bildes zur ersten Zeile des nachfolgenden Bildes. Es kann demnach nicht die komplette Zeit *aktiv* für das Lesen einer Bildvorlage genutzt werden; es wird unterschieden zwischen aktiver und nomineller Zeilenzahl, sowie zwischen aktiver und nomineller Zeilendauer.

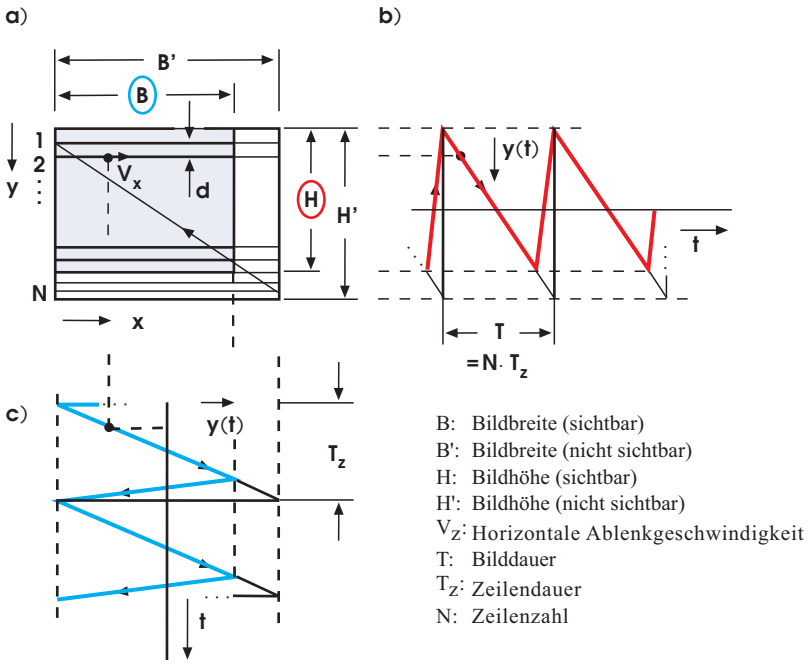


Abb. 1.4-2: Sichtbarer und nichtsichtbarer Bildbereich und die Ablenkung des Lese- bzw. Schreibstrahles

Notwendige Zeilenzahl

Die Qualität des reproduzierbaren Bildes hängt entscheidend von der Anzahl der

verwendeten Zeilen ab. Zahlreichen Untersuchungen hatten das Ziel, eine allgemeine Vorschrift für die Anzahl der notwendigen Zeilen zu finden. Eine solche Näherungsformel lautet:

$$N_{\text{aktiv}} = \frac{2500}{E/H}, \qquad \text{1.4-1}$$

mit *E* als Betrachtungsabstand und *H* als sichtbare Bildhöhe (vgl. Abb. 1.4-2). Bei einem Betrachtungsabstand der dem fünffachen der Bildhöhe entspricht ergibt sich damit eine mindestens notwendige Anzahl von 500 aktiven Zeilen. Schließlich standardisiert wurden die folgende Werte:

$$\begin{aligned} &50 \text{ Hz-Länder:} N = 625;\ N_{\text{aktiv,analog}} = 575;\ N_{\text{aktiv,digital}} = 576 \\ &60 \text{ Hz-Länder:} N = 525;\ N_{\text{aktiv,analog}} = 485;\ N_{\text{aktiv,digital}} = 480 \end{aligned} \qquad \text{1.4-2}$$

Weitere wichtige Normen für die beiden Systeme betreffen das *Bild-Seitenverhältnis* sowie die *Bildwechselfrequenz*.

Bild-Seitenverhältnis

Unter dem *Bild-Seitenverhältnis* versteht man das Verhältnis der Bildbreite *B* zur Bildhöhe *H*. Für beide Systeme galt ursprünglich:

$$\frac{B}{H} = \frac{4}{3}. \qquad \text{1.4-3}$$

Die vielen Breitbildformate beim heutigen Kino haben zu der Einführung eines weiteren Bild-Seitenverhältnisses (ursprünglich nur für *HDTV*, weltweit) geführt:

$$\frac{B}{H} = \frac{16}{9}. \qquad \text{1.4-4}$$

Beide Bild-Seitenverhältnisse sind heute üblich. In der Praxis wird das Format durch das Bildwiedergabesystem festgelegt. Dies kann zu schwarzen Balken oben und unten („*Letterbox*“) bei der Wiedergabe eines 16:9-Signals auf einem 4:3- Bildschirm bzw. zu schwarzen Balken links und rechts („*Side-Panel*“) bei der Wiedergabe eines 4:3-Signals auf einem 16:9-Bildschirm führen.

Bildwechselfrequenz

Ähnlich wie beim Kino wird bei der analogen Bildtechnik die bewegte Bildvorlage als eine Folge von Einzelbildern übertragen. In Abschnitt 1.2.4 wurde bereits dargestellt, dass im Kino dazu 24 Bilder pro Sekunde verwendet werden, die aber mittels Flügelblende bei der Filmprojektion jeweils zweimal[2] gezeigt werden. Es entsteht somit eine effektive *Bildwechselfrequenz* von 48 Hz. Bei gut abgedunkelten Räumen und den damit verbunden geringen Leuchtdichten bewegt sich das wahrnehmbare Flimmern im vertretbaren Rahmen. In der analogen Bildtechnik ist eine solche empfängerseitige Bildwiederholung nicht so leicht zu realisieren. Um dennoch

2 In einigen wenigen Fällen ist sogar die dreimalige Wiederholung eines einzelnen Filmbildes üblich. Dann ergibt sich die effektive Bildwechselfrequenz zu 72 Hz.

eine weitgehend flimmerfreie Bilddarstellung zu erreichen, wird die Bildvorlage abweichend von Abb. 1.4-1 und Abb. 1.4-2 im so genannten Zeilensprungverfahren abgetastet und übertragen.

Zeilensprungabtastung

Zunächst verkoppelt man die Bildfrequenz mit der Netzfrequenz des jeweiligen Landes, um mögliche Interferenzstörungen mit Lichtquellen ausschließen zu können. So wird in Europa statt der Abtastung aller 625 Zeilen innerhalb von 40 ms (25 Hz) zunächst nur die Hälfte der Zeilen abgetastet (also 312,5), und zwar im ersten Schritt alle ungeraden Zeilen und danach alle geraden Zeilen. So entstehen zwei Halbbilder, welche mit der Halbbildfrequenz *f*v von 50 Hz übertragen werden. Abb. 1.4-3 zeigt die so entstandenen abgetasteten Halbbilder.

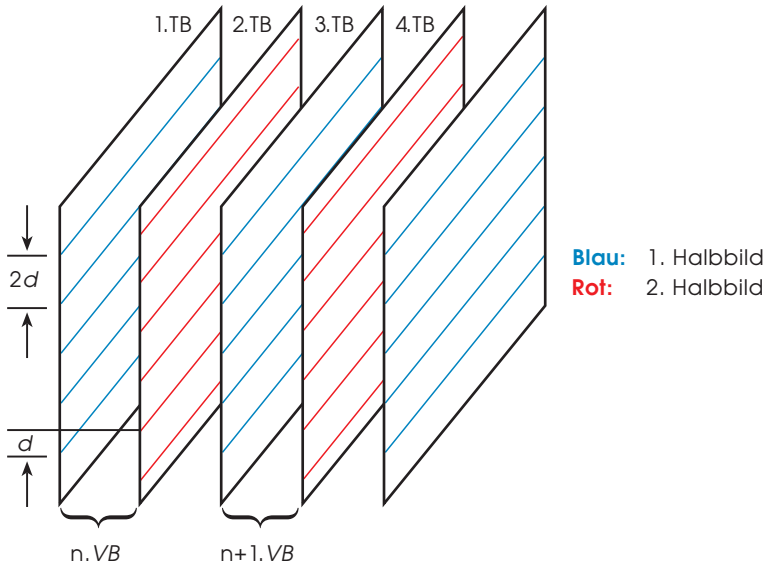


Abb. 1.4-3: Bildfolge bei der Zeilensprungabtastung

Die so genannte Zeilenfrequenz *f*h (auch: Zeilenwechselfrequenz, Horizontalfrequenz) errechnet sich direkt gemäß der Randbedingungen von Abb. 1.4-2 aus der Halbbildfrequenz *f*v = 50 Hz, der Anzahl der nominellen Zeilen *N* = 625:

$$f_h = \frac{N}{2} \cdot f_v = 15625\ Hz. \qquad 1.4\text{-}5$$

Für das amerikanische TV-System mit $f_v = 59.94\ \ Hz$[3] und *N* = 525 ergibt sich:

$$f_h = \frac{N}{2} \cdot f_v = 15734.264\ Hz. \qquad 1.4\text{-}6$$

3 Beim Übergang vom Schwarzweiß- zum Farbfernsehsystem wurde eine geringfügige Änderung der Zeilen- und Halbbildfrequenz vorgenommen, um Übersprechstörungen zwischen Farb-, Luminanz- und Tonsignal weitestgehend unterdrücken zu können.

1.4.2 Das FBAS-Signal

Das ***F**arb**b**ild**a**ustast**s**synchron-Signal* (*FBAS-Signal*) enthält alle Bild- und Synchroninformationen so, wie sie zeitsequenziell übertragen werden. Abb. 1.4-4 zeigt beispielhaft den Verlauf dieses Signals für eine Zeile (Europäische Fernsehnorm).

Analoges Zeilensignal

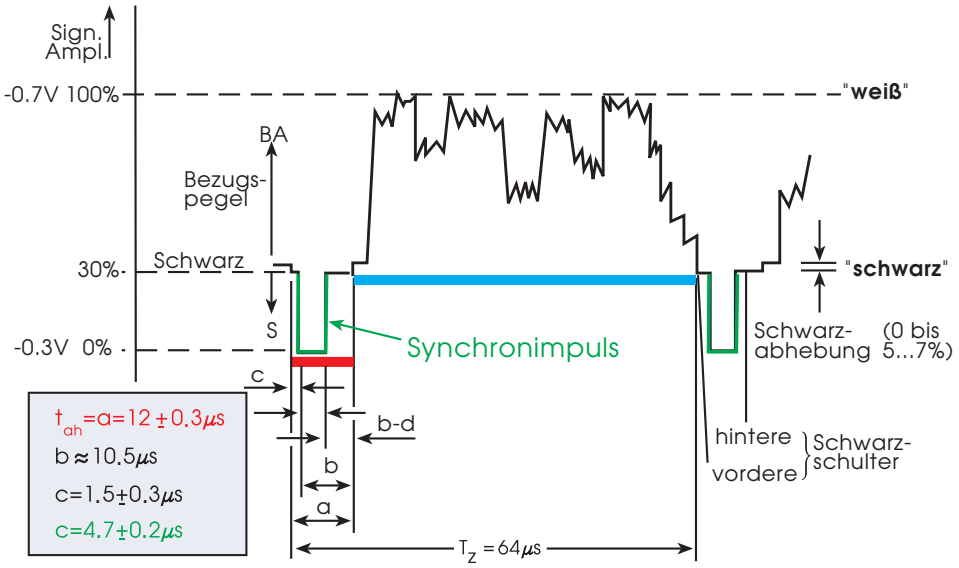


Abb. 1.4-4: FBAS-Signal

Man kann zwischen dem aktiven Bildbereich (*blau*) und dem nicht sichtbaren Bereich (*horizontale Austastlücket*$_{ah}$) einer Zeile (*rot*) unterscheiden. Innerhalb des aktiven Zeilenbereiches verändert sich das Signal abhängig von der Helligkeit der korrespondierenden Bildvorlage: Für ein *weiß* wird eine Spannung von +0.7 V erzeugt, ein *schwarz* liegt nahe bei 0 V. Dazwischen liegen alle übrigen Grauwerte. Im nichtsichtbaren Bereich ist ein Synchronimpuls mit –0.3 V Amplitude enthalten. Er ist etwa 4.7 µs lang und trägt dafür Sorge, dass die einzelnen Zeilen am Empfänger wieder exakt synchron übereinander geschrieben werden können.

Die nominelle Zeilendauer T_z ergibt sich aus der Zeilenfrequenz f_h nach

$$T_z = \frac{1}{f_h} = 64\mu s. \qquad 1.4\text{-}7$$

Die horizontale Austastlücke t_{ah} beträgt etwa 12 µs. Somit verbleibt für die aktive Zeilendauer ein Wert von 52 µs.

Die *Farbinformation* der Bildvorlage wird beim europäischen Farbfernsehen in der Form einer hochfrequenten Schwingung (*Farbhilfsträger*, $f_{sc} = 4.43\ MHz$ auf der Basis der *quadraturmodulierten* und *gammakorrigierten*[4] Farbdifferenzsignale

Farbhilfsträger
Farbdifferenzsignale

$$U = E_U = 0.493(E'_B - Y) \quad \text{und}$$
$$V = E_V = 0.877(E'_R - Y) \qquad \text{1.4-8}$$

in das Schwarzweißsignal eingebettet. In Matrixschreibweise lässt sich der Übergang der Farbkomponenten E'_R, E'_G, und E'_B nach *Y*, *U* und *V* dann unter Verwendung der Gl. 1.3-6 folgendermaßen ausdrücken:

$$\begin{pmatrix} U \\ V \\ Y \end{pmatrix} = \begin{pmatrix} -0.147 & -0.289 & +0.436 \\ +0.615 & -0.515 & -0.100 \\ +0.299 & +0.587 & +0.114 \end{pmatrix} \cdot \begin{pmatrix} E'_R \\ E'_G \\ E'_B \end{pmatrix} \qquad \text{1.4-9}$$

In der digitalen Bildtechnik werden Komponentensignale ebenfalls oft verwendet. Die Farbdifferenzsignale verwenden dann jedoch andere Konstanten in der Tabelle 1.4-1. Zur Unterscheidung ist dann oft auch die Schreibweise C_B und C_R üblich (vgl. Abschnitt 2.2.2).

Für die Synchronisation des Farbsignals wird auf der *hinteren Schwarzschulter* (vgl. Tabelle 1.4-1) ein weiteres Signal, der so genannte *Farbburst*, übertragen, welcher aus etwa 10 Schwingungen des Farbhilfsträgers besteht.

Tab. 1.4-1: *Parameter analoger Fernsehstandards nach ITU-R.BT 470*

	Europa (PAL, Norm G/B)	**USA (NTSC,Norm M)**
Zeilenzahl, nominell *N*:	625	525
Zeilenzahl, aktiv N_{aktiv}, $(N_{aktiv,digital})$:	575 (576)	485 (480)
Zeilendauer T_z:	64 µs	63.55 µs
Zeilenfrequenz f_z:	15625 Hz	15734 Hz
Horizontale Austastlücke t_{ah}:	12 µs	10.9 µs
Vollbildfrequenz f_B:	25 Hz	29.97 Hz
Halbbildfrequenz f_{HB}:	50 Hz	59.94 Hz
Halbbilddauer T_{HB}:	20 ms	16.6 ms
Vertikale Austastlücke t_{av}:	25 Zeilen	20 Zeilen
Bild-Seitenverhältnis:	4:3	4:3
Farbhilfsträgerfrequenz f_{sc}	4.43 MHz	3.58 MHz
Grenzfrequenz Bildsignal $f_{B_{max}}$:	5 MHz	4.2 MHz

Tabelle 1.4-1 fasst noch einmal die wichtigsten Zahlen der beiden analogen Fern-

Parameter analoger Fernsehsysteme

4 Der Zusammenhang zwischen elektrischem Signalwert und korrespondierendem Helligkeitswert muss kameraseitig bewusst nichtlinear sein, damit bei der Wiedergabe über einen Kathodenstrahlmonitor ohne weitere Maßnahmen die Darstellung insgesamt wieder linear ist. Die Nichtlinearität des Monitors wird also bereits auf der Kameraseite berücksichtigt und entsprechend ausgeglichen.

sehstandards *PAL* (Europa) und *NTSC* (USA) zusammen. Weitere Details, insbesondere auch über die Unterschiede in der Toncodierung, lassen sich in [1.7] und [1.8] nachlesen.

Die nachfolgende Abb. 1.4-5 zeigt den Signalverlauf einer Zeile eines 75%-FBAS-Farbbalkentestsignals (PAL-Standard). Es veranschaulicht die Signale des Luminanzsignals (Graustufentreppe) und das überlagerte, hochfrequente Farbartsignal der Farbdifferenzkomponenten *U* und *V*.

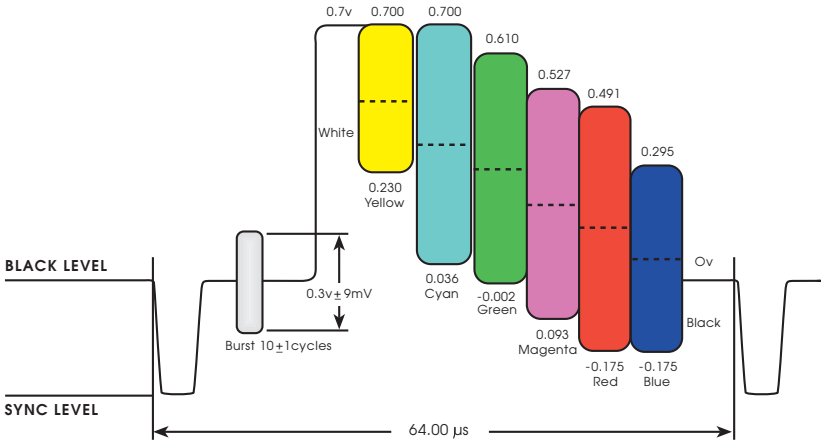


Abb. 1.4-5: 100/75%-Farbbalkentestsignal nach dem PAL-Standard

Selbsttestaufgabe 1.4-1:

Beschreiben Sie kurz die Bildfeldzerlegung bei der analogen Bildtechnik!

Selbsttestaufgabe 1.4-2:

Warum wird zwischen nomineller und aktiver Zeilenzahl unterschieden?

Selbsttestaufgabe 1.4-3:

Welche Aufgabe hat die horizontale Austastlücke?

Selbsttestaufgabe 1.4-4:

Geben Sie eine Transformationsgleichung an, mit der man aus dem Farbvektor (E'_R, E'_G, E'_B) den Vektor (Y, U, V) für ein EBU-Farbsystem ermitteln kann!

2 Digitale Bilddarstellung und -übertragung

2.1 Abtastung und Quantisierung

Das Prinzip der Fernsehübertragung, das in Abb. 1.4-1 gilt auch bei der digitalen Erzeugung und Übertragung von Bildern. Auch hier gibt es eine Bildfeldzerlegung in Zeilen und sequenziell aufeinander folgender Bilder. Zwei Prozesse kommen hier jedoch hinzu: Die örtliche und zeitliche Abtastung des analogen Bildsignals sowie die Diskretisierung (*Quantisierung*) der Werte.

Abb. 2.1-1 zeigt den Prinzip der Signalaufbereitung. Ausgehend von den analogen Komponentensignalen *Y* und *B–Y, R–Y* wird eine Analog-Digital-Wandlung in den drei Signalkanälen vorgenommen (Abtastung: Diskretisierung und Quantisierung). Die digitalen Komponentensignale werden anschließend zunächst zu einem *N*-Bit breiten Zeitmultiplexsignal zusammengefasst. Für die Übertragung wird oft zudem noch ein Parallel-Seriell-Wandler nachgeschaltet, um die Übertragung in einem seriellen Komponentensignal (*Digital Serial Components, DSC*) zu erreichen.

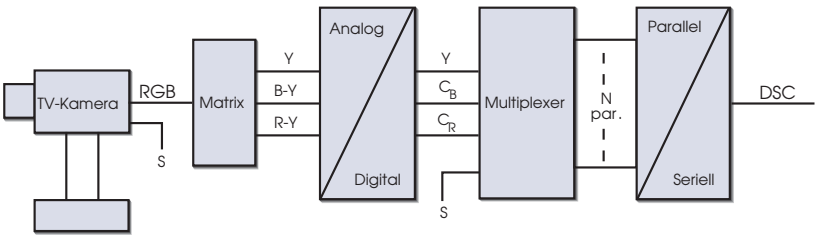


Abb. 2.1-1: Aufbau einer Bildaufnahmeeinheit mit digitalem Signalausgang (nach [2.2])

Die für die Analog-Digital-Wandlung erforderliche zeitliche Abtastung bzw. die örtliche Diskretisierung geschieht dabei in folgender Weise: Zunächst wird das analoge Eingangssignal auf eine maximale Signalfrequenz f_max mit einem Tiefpass bandbegrenzt. Dieses Signal wird dann periodisch mit kurzen Impulsen abgetastet. Dieses ist der Vorgang der Diskretisierung. Es entsteht eine Folge von analogen Spannungswerten, die auf einen beschränkten Zahlenraum „gerundet“ werden. Diesen Vorgang nennt man Quantisierung. Als Ergebnis entsteht am Ausgang eine unendliche Folge von *N*-bit breiten Codeworten, welche das ursprüngliche analoge Eingangsignal repräsentieren. 2^N gibt hierbei den Zahlenraum an, welcher für die Quantisierung benutzt wird. In der digitalen Bildtechnik ist eine Wortbreite von *N* = 8 und *N* = 10 üblich. Daraus ergeben sich für eine Graustufendarstellung 256 bzw. 1024 verschiedene Werte. Diese Anzahl reicht in den meisten Anwendungsfällen aus, ohne dass der Betrachter Graustufensprünge an ursprünglich kontinuierlich verlaufenden Helligkeitsverläufen wahrnimmt.

Entscheidender für die Bildqualität eines digitalen Bildsignals ist vielmehr die richtige Wahl der Abtastfrequenz f_T. Aus der Signaltheorie ist bekannt, dass für eine fehlerfreie Rekonstruktion des analogen Eingangssignals aus der o.g. Zahlenfolge gelingen kann, wenn die Abtastfrequenz f_T mindestens doppelt so hoch ist, wie die höchste vorkommende Frequenz des Eingangssignals.

Abtasttheorem nach Shannon

Diesen Zusammenhang nennt man das *Abtasttheorem nach Shannon*, welches formal lautet:

$$f_T \geq 2 \cdot f_{\max}. \qquad \text{2.1-1}$$

Da im allgemeinen nicht von einer idealen Bandbegrenzung des eingangsseitig verwendeten Tiefpasses ausgegangen werden kann, muss die Abtastfrequenz in der Praxis oft sehr viel größer sein, als die doppelte Nutzsignalfrequenz.

Die nachfolgenden Beispielbilder veranschaulichen noch einmal den Einfluss der Diskretisierung und der Quantisierung. Abb. 2.1-2 zeigt ein Beispielbild, bei dem die Grauwerte im Rahmen der Darstellungsgenauigkeit als orts- und wertekontinuierlich angesehen werden sollen.

Abb. 2.1-2: Orts- und wertekontinuierliches Originalbild (analoges Eingangssignal)

Beispiel Diskretisierung

Abb. 2.1-3 zeigt, wie sich eine Diskretisierung (örtliche Abtastung) auswirkt, wenn das Abtasttheorem nicht eingehalten wird, weil entweder der Eingangstiefpass keine ausreichende Bandbegrenzung gewährleistet bzw. die Abtastfrequenz für das Eingangssignal zu niedrig gewählt wurde.

Abb. 2.1-3: Ortsdiskretisierung (Werte weiterhin kontinuierlich)

Beispiel Quantisierung

Abb. 2.1-4 zeigt die Störungen, wenn zwar das Abtasttheorem eingehalten wird, die Grauwerte jedoch nicht mehr auf einen ausreichend großen Zahlenraum quantisiert werden.

Abb. 2.1-4: Werte zu stark quantisiert, (Ortsauflösung kontinuierlich bzw. Abtasttheorem eingehalten)

Abb. 2.1-5 visualisiert die Störungen, welche durch eine zu grobe Diskretisierung und einer zu geringen Werte-Quantisierung entstehen.

Abb. 2.1-5: Werte zu stark quantisiert, örtliche Diskretisierung zu grob

Das durch die Quantisierung verursachte Störsignal wird allgemein auch als *Quantisierungsrauschen* bezeichnet. Dabei geht man davon aus, dass bei klein gewählten Quantisierungsstufen eine Zufallsfolge entsteht, welche mit einem gleichverteilten vom ursprünglichen Signal weitgehend unabhängigem Spektrum korreliert. Damit lässt sich näherungsweise ein *Signal-Störabstand* ermitteln:

Signal-Störabstand

$$S_Q \approx (6 \cdot N + 10.8)\ dB. \qquad \text{2.1-2}$$

Ein Herleitung findet sich in [2.1] und [2.3].

2.2 Digitale Bildformate, Farbdarstellungen und Standards

2.2.1 Farbabtastraster

Die bei der Diskretisierung entstandenen Abtastwerte beziehen sich üblicherweise auf das aktive Bild[5] mit einer Anzahl von *n* Pixel je Zeile und N_{aktiv} Zeilen je Bild. Die Werte können sowohl *progressives* Bildmaterial repräsentieren oder aus einer *Zeilensprungabtastung* hervorgehen. Obwohl dem Zeilensprungverfahren viele Nachteile anhaften, basieren viele digitale Bildnormen weiterhin auf diesem Standard. Insbesondere in den klassischen Fernseh-Umgebungen wird aus Kompatibilitätsgründen noch das Zeilensprungverfahren auch in der digitalen Bildtechnik verwendet (Beispiel: *Digital Video Broadcast - DVB*, vgl. Kapitel 6).

Auf der Suche nach Einsparpotentialen der digitalen Repräsentation eines analogen Bildes werden bereits bei der Abtastung Eigenschaften des menschlichen visuellen Systems in Bezug auf die *Kontrastempfindlichkeit* genutzt. In Abb. 1.2-6 (Abschnitt 1.2.3) wurde die Kontrastempfindlichkeit für farbart- und helligkeitsmodulierte räumliche Gitter betrachtet. Als Ergebnis konnte festgehalten werden, dass das räumliche Auflösungsvermögen für reine Farbartübergänge geringer ist, als bei Helligkeitsübergängen. Signaltheoretisch betrachtet dürfen demnach die Farbart beinhaltenden Signalanteile (= *Farbdifferenzsignale*) *B–Y* und *R–Y* vor der Abtastung stärker bandbegrenzt werden als das Luminanzsignal *Y*. Damit ist auch eine niedrigere Abtastfrequenz für die Farbdifferenzsignale möglich. Um einfache örtliche Bezüge der digitalisierten Codeworte für Luminanz und Chrominanz zu haben, stehen die Abtastfrequenzen im Verhältnis 2:1 oder 4:1 zueinander. Zur Unterscheidung werden diese digitalisierten und örtlich unterabgetasteten Farbdifferenzsignale mit

Digitale Farbdifferenzsignale C_B und C_R

$$C_B = (E'_B - E'_Y)_{\perp,\downarrow s} \qquad \text{2.2-1}$$

bzw.

$$C_R = (E'_R - E'_Y)_{\perp,\downarrow s} \qquad \text{2.2-2}$$

bezeichnet, wobei E'_Y, E'_B und E'_R die gammakorrigierten Signale von *Y*, *B* und *R* bezeichnen, das Symbol $\perp$ den Abtastvorgang beschreibt und $\downarrow s$ die Reduktion um den Faktor *s* im Verhältnis zur Abtastung des Luminanzsignals *Y* angibt.

Mit diesen Vorüberlegungen werden die so genannten *4:4:4-* und *4:2:2-Abtastformate* definiert. Sie bezeichnen das Verhältnis der Abtastfrequenzen zwischen *Y*, C_B und C_R, bezogen auf eine Basisfrequenz von

$$f_{\perp,Basis} = 3.375 \text{ MHz}. \qquad \text{2.2-3}$$

5 d. h. Bildbereiche der horizontalen oder vertikalen Austastlücke bleiben unberücksichtigt.

So ergibt sich für ein Abtastformat von 4:4:4 eine Abtastfrequenz von

$$f_{\perp} = 13.5 \text{ MHz} \qquad 2.2\text{-}4$$

sowohl für das Luminanzsignal *Y* als auch für beide Farbdifferenzsignale C_B und C_R. Es geschieht keine örtliche Auflösungsreduktion in der Farbe. Dem gegenüber ergeben sich beim 4:2:2-Abtastformat Frequenzen für die Farbdifferenzsignale von

$$f_{\perp,\downarrow 2} = 13.5 \text{ MHz}/2 = 6.75 \text{ MHz}. \qquad 2.2\text{-}5$$

Die Wahl der Basisfrequenz $f_{\perp,Basis}$ mit dem Wert von 3.375 MHz ist nicht willkürlich, sondern sorgt für eine starre Verkopplung mit der Zeilenfrequenz der jeweiligen Farbfernsehnorm:

625-Zeilen-System (PAL-Norm Europa): $3.375 \text{ MHz} = 216.0 \cdot 15625 \text{ Hz}$
525-Zeilen-System (NTSC-Norm USA): $3.375 \text{ MHz} = 214.5 \cdot 15734 \text{ Hz}$.

Die Verkopplung sorgt für ein orthogonales Abtastraster, so dass die digital abgetasteten Bildpunkte übereinander und in jedem Teilbild immer an der gleichen örtlichen Position liegen.

Abb. 2.2-1 veranschaulicht die Luminanz- und Chrominanzpositionen für verschiedene Abtastformate. Angegeben sind die Luminanzwerte (weiß) sowie die Chrominanzwerte C_B (blau) und C_R (rot) und deren örtlicher Zugehörigkeitsbereich (rot umrandet). Darüber hinaus ist der Datenreduktionsfaktor angegeben, der sich mit dem jeweiligen Abtastformat gegenüber einer 4:4:4-Abtastung erzielen lässt.

Abtastformate

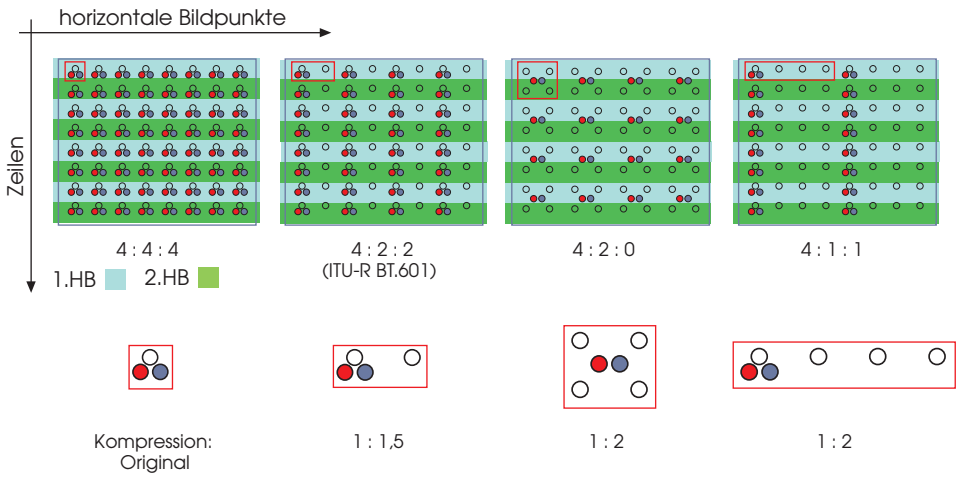


Abb. 2.2-1: Luminanz- und Chrominanzpositionen für verschiedene Abtastformate

Es ist zu erkennen, dass die Abtastformate 4:2:2 bzw. 4:1:1 zu einer örtlichen Auflösungsreduktion rein horizontal um den Faktor 2 bzw. 4 führen. Tatsächlich spielt das 4:4:4-Abtastformat in der digitalen Bildübertragung kaum eine Rolle. Spezialanwendungen im virtuellen Studio (*Blue-Box-Verfahren*) oder in der Medizin- und Radartechnik sind die Domänen für dieses Format. Das klassische digitale Studiosignal hingegen kommt mit einem Abtastformat von 4:2:2 aus (vgl. Abb. 2.2-1b).

Für die Übertragung zum Endnutzer wird eine weitere Reduzierung der örtlichen Farbauflösung vorgenommen. Dabei gibt es ausgehend vom 4:2:2-Format zwei verschiedene Varianten, die in unterschiedliche Länderstandards integriert wurden.

4:1:1-Abtast- und Übertragungsformat

Leicht zu realisieren ist eine weitere horizontale Unterabtastung um den Faktor 2 (vgl. Abb. 2.2-1d). Dann ergibt sich insgesamt eine horizontale Reduzierung der Farbauflösung auf 25 %, vertikal bleibt die volle Auflösung erhalten. Bei der Magnetbandaufzeichnung (*DV*-Bildcodierung, *DVCPro 25*, vgl. Kapitel 4) und der datenreduzierten Übertragung von Videosignalen basierend auf 525-Zeilen-Systeme (z. B. USA) wird dieses Format verwendet.

4:2:0-Abtast- und Übertragungsformat

Um horizontal und vertikal in etwa eine gleiche örtliche Farbauflösung zu haben, wird in vielen Anwendungen oft statt der zusätzlichen horizontalen Auflösungsreduktion die vertikale Reduktion bevorzugt (vgl. Abb. 2.2-1c). Diese zunächst recht plausible Vorgehensweise entpuppt sich aber gerade bei Zeilensprungsignalen als etwas tückisch. Verwendet man nämlich 4 im Quadrat örtlich benachbarte Luminanzpixel, auf die sich ein C_B-C_R-Paar beziehen soll, wird man feststellen, dass die übereinander liegenden Luminanzwerte aus verschiedenen Halbbildern und damit zeitlich aus unterschiedlichen Bewegungsphasen stammen (vgl. Abb. 1.4-3). Die Signalverarbeitung zur Berechnung der Chrominanzwerte des 4:2:0-Abtastrasters geschieht also in dem man durch eine örtlich-zeitliche Mittelwertbildung virtuelle „*Zwischenzeilen*“ errechnet, auf die sich die Chrominanzwerte beziehen. Dieser Vorgang wird oft auch als Formatkonversion bezeichnet.

Man muss sich daher im Klaren darüber sein, dass neben der vertikalen Auflösungsreduktion auch eine zeitliche Bewegungsverschleifung in den Farbkomponenten unvermeidlich ist. Diese Problematik ist aber in den Fällen weniger gravierend, in denen das analoge Eingangsbild auf dem PAL-Standard basiert, da bereits hier für die Chrominanz-Luminanz-Trennung ein örtlich-zeitliches Filter im Chrominanzpfad verwendet wird, das eine ähnliche Filterung vornimmt, wie die, die bei der 4:2:0-Abtastung erforderlich ist. Das ist auch der Grund, warum das 4:2:0-Abtastformat oft auf 625-Zeilen-Systemen (z. B. PAL-Europa) aufgesetzt wird (*DV-Codierung* – Kapitel 4, *DVB-Verfahren* – Kapitel 6).

Bei der Schreibweise „4:2:0-Abtastformat“ muss man zudem beachten, dass die Zahlen nicht mehr die Systematik mit den Vielfachen des Basiswertes der Abtastfrequenz anwenden; es handelt sich hierbei vielmehr um eine symbolische Schreibweise zur Unterscheidung der verschiedenen Abtastraster. Tatsächlich kann man schließlich noch das Abtastformat 4:1:0 definieren, das eine 4-fache horizontale und 2-fache vertikale Unterabtastung der Farbkomponenten bezeichnet.

2.2.2 Digitale Studionorm ITU-R BT.601

Die erste digitale Studionorm wurde vom *Comité Cosultatif International des Radiocommunications* (*CCIR*) 1982 als Empfehlung (*Recommendation*) 601 „*Encoding Parameters of Digital Television for Studios*“ verabschiedet. In der Folgezeit gab es neben der Umorganisation des *CCIR* zum heutigen *ITU* (*International Telecommunication Union*) auch einige Ergänzungen zur ursprünglichen Norm 601. Für die nachfolgenden Betrachtungen soll die Version *ITU-R BT.601-4* aus dem Jahr 1994 herangezogen werden (vgl. [2.5]).

Diese Norm gibt die digitale Darstellung von 525-Zeilen- und 625-Zeilen-Systeme an, welche nach dem 4:2:2-Abtastformat codiert werden.

Die Pegel werden vor der Abtastung und Quantisierung wie folgt normiert: Das Y-Signal deckt analog einen Spannungsbereich zwischen 0 bis 700 mV ab. Dieser Bereich wird digital auf einen Bereich zwischen 0 und 1.0 V normiert. Damit auch die Farbdifferenzsignal C_B und C_R des Farbbalkensignals (vgl. Abb. 1.4-4) auf einen Bereich zwischen -0.5 V und +0.5 V abgebildet werden können, dürfen die korrespondierenden analogen Signale nur zwischen -350 mV und +350 mV liegen. Dies wird möglich, wenn die Farbdifferenzsignale C_B und C_R abweichend von Gl. 2.2-1 und Gl. 2.2-2 unter Berücksichtigung der Gleichung für die EBU-Primärvalenzen (vgl. Gl. 1.3-5) wie folgt gebildet werden:

Farbdifferenzsignale bei *ITU-R BT.601*

$$C_B = 0.564 \cdot (E'_B - E'_Y)\ \ C_B = -0.169E'_R - 0.331E'_G + 0.500E'_B \qquad \text{2.2-6}$$

$$C_R = 0.713 \cdot (E'_R - E'_Y)\ \ C_R = 0.500E'_R - 0.419E'_G - 0.081E'_B. \qquad \text{2.2-7}$$

In diesem Zusammenhang beachte man den Unterschied bei Normierung zwischen der analogen Bildübertragung nach Gl.1.4-8 und den o. g. Gl. 2.2-6 und Gl. 2.2-7.

ITU-R BT.601-Parameter

Die nachfolgende Tabelle (Tabelle 2.2-1) fasst die wesentlichen Parameter der Norm ITU-R BT.601 für das 525-Zeilen- und 625-Zeilen-System zusammen. Einige Besonderheiten sind zu beachten. So sind beispielsweise für beide Fernsehsysteme die gleiche Abtastfrequenz (*Y*: 13.5 MHz, C_R, C_B: 6.75 MHz) und die gleiche Anzahl von aktiven Pixeln (*Y*: 720, C_R, C_B: 360) festgelegt worden. Das hat zur Konsequenz, dass sich die Anzahl der in die Austastlücke fallenden Abtastwerte zwischen beiden Normen geringfügig unterscheidet.

Tab. 2.2-1: *Wesentliche Parameter der ITU-R BT.601*

Fernsehstandard (korrespondiert mit)	525 Zeilen-System (60 Hz) (*NTSC* Norm M)	625 Zeilen-System (50 Hz) (*PAL* Norm B/G)
Anzahl aktive Zeilen	480	576
Zeilendauer	63.55 µs	64 µs
Anzahl Abtastwerte über gesamte Zeilendauer (nominell)	*Y*: 858 C_B, C_R: 429	*Y*: 864 C_B, C_R: 432
Anzahl Abtastwerte je aktiver Zeile (53.33 µs)	*Y*: 720 C_R, C_B: 360	
Abtastfrequenz	*Y*: 13.5 MHz C_B, C_R: 6.75 MHz	
Bandbegrenzung der zu codierenden Signale (±0,05 dB)	*Y*: 5.75 MHz C_R, C_B: 2.75 MHz	
Quantisierung	Linear : 8 Bit oder 10 Bit (bevorzugt)	
Nutzbare Stufenzahl 8 Bit 10 Bit	*Y*: 220, C_R, C_B: 225 *Y*: 877, C_R, C_B: 897	
Abtaststruktur	Orthogonal (übereinander liegende Abtastwerte)	

Darüber hinaus sind die Pixel nicht exakt quadratisch. Um dieses zu erreichen müssten für ein 4:3 Bildseitenverhältnis und den 576 abgetasteten aktiven Zeilen aus beiden Halbbildern insgesamt

$$n_{square} = 576 \text{ Pixel} \cdot \frac{3}{4} = 768 \text{ Pixel} \qquad 2.2\text{-}8$$

je Zeile vorhanden sein. In einigen Spezialanwendungen werden abweichend von der Norm tatsächlich 768 statt 720 aktive Pixel pro Zeile verwendet.

Aus den Angaben in Tabelle 2.2-1 lässt sich weiterhin entnehmen, dass die aktive Zeilendauer

$$T_{Z,\perp} = \frac{720}{13.5\ MHz} \approx 53.33\ \mu s \qquad 2.2\text{-}9$$

von dem korrespondierenden Zahlenwert

$$T_{Z,analog} = 52\ \mu s \qquad 2.2\text{-}10$$

des analogen Umfeldes abweicht. Das wiederum hat zur Folge, dass nach der Analog-Digital-Wandlung einer analogen Zeile nur effektiv etwa

$$n_{effektiv} = \frac{720\ Pixel}{53.33\ \mu s} \cdot 52\ \mu s \approx 702\ Pixel \qquad 2.2\text{-}11$$

genutzt werden. Es verwundert daher auch nicht, wenn in vielen Übertragungen nur 704 aktive Pixel verwendet werden[6].

Signalpegelbereich bei ITU-R BT.601

Bei der Quantisierung der Signalpegel auf 8- bzw. 10-Bit Codeworte werden

6 Aufgrund der besonderen Eigenschaften der allgemein eingesetzten Quellencodierung (blockbasierte DCT, vgl. Kapitel 3) muss die aktive Pixelzahl pro Zeile durch 8 teilbar sein.

die Extremwerte 0 und 255 (bei 8-Bit-Codierung) bzw. 0 und 1023 (bei 10-Bit-Codierung) nicht verwendet. Sie sind reserviert für spezielle Meldungen, die Synchronisationszwecken dienen (vgl. Abschnitt 2.3). Weiterhin werden beim abgetasteten Luminanzsignal die Bereiche zwischen 1 und 15 bzw. 1 und 63 als so genannten *Feetroom* und die Bereiche zwischen 236 und 254 bzw. 941 und 1022 als so genannten *Headroom* verwendet. Diese Bereiche dienen als Über- bzw. Untersteuerungsreserve für Überschwinger des Analogsignals. Bei den Chrominanzsignalen sind dies die Bereiche zwischen 1 und 15 bzw. 1 und 63 sowie zwischen 241 und 254 bzw. 961 und 1022.

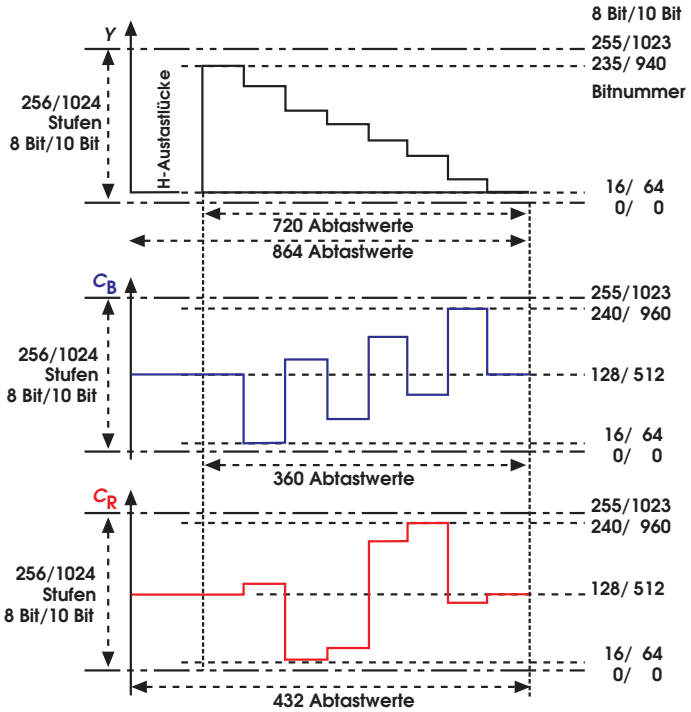


Abb. 2.2-2: Abtast- und Quantisierungswerte nach ITU-R BT.601 (Farbbalken)

In Abb. 2.2-2 sind die Abtast- und Quantisierungswerte nach ITU-R BT.601 eines 100/75%-Farbbalkensignals (vgl. Abb. 1.4-5) für eine 8- und 10-Bit Quantisierung dargestellt.

16:9 und ITU-R BT.601

Beim Übergang von einem Bildseitenseitenverhältnis von 4:3 auf 16:9 (vgl. Gl. 1.4-3 und Gl. 1.4-4) ist es notwendig, die Anzahl der horizontalen Abtastwerte um den Faktor 1.33 auf 960 aktive Pixel zu erhöhen. Damit wird auch eine Abtastfrequenz für die Luminanz von 18 MHz notwendig. Auch dieser Fall wird in der Norm ITU-R BT.601 berücksichtigt.

In der Praxis zeigt sich jedoch, dass eine Erhöhung der horizontalen Abtastwerte bei einem 16:9 Bildseitenverhältnis nicht zwingend notwendig ist. Im Hinblick auf existierende Endgeräte im 16:9 PALplus-Format ist die mit effektiven 702 Abtastwerten erreichbare horizontale Auflösung immer noch höher, als die der analogen Endgeräte. Selbstverständlich sind dann die abgetasteten Pixel noch weniger quadratisch, sondern entsprechen Quadern mit einem Seitenverhältnis von

$$\frac{B}{H_{Pixel}} = \frac{\frac{576}{9}}{\frac{720}{16}} = \frac{64}{45} \approx 1.422. \qquad \text{2.2-12}$$

Digitales Compositesignal

Der Vollständigkeit halber sei hier auch noch darauf hingewiesen, dass auch ein Digitalstandard für das abgetastete *FBAS*-Compositesignal definiert ist. Um die Störungen bei der Abtastung in Bezug auf den Farbhilfsträger gering zu halten, wird das FBAS-Signal exakt mit der vierfachen Farbhilfsträgerfrequenz f_{sc} abgetastet:

$$f_{\perp,Composite} = 4 \cdot f_{sc} = 4 \cdot (283.75 \cdot 15625Hz + 25Hz) \approx 17.73\ MHz. \qquad \text{2.2-13}$$

Dieser Standard wird heute nur noch bei einigen wenigen digitalen Aufzeichnungsverfahren verwendet. Nachteilig ist die beschränkte 8 Bit-Auflösung, die PAL-typischen Störungen (z. B. Cross-Störungen zwischen Helligkeits- und Farbartsignal) sowie die recht beschränkte Farbauflösung des zugrunde liegenden FBAS-Signals.

Selbsttestaufgabe 2.2-1:

Formulieren Sie das Abtasttheorem nach Shannon!

Selbsttestaufgabe 2.2-2:

Definieren Sie die Begriffe Diskretisierung und Quantisierung!

Selbsttestaufgabe 2.2-3:

Erläutern Sie die Abtastformate 4:4:4, 4:2:2, 4:1:1 und 4:2:0! Welche Besonderheit ist bei 4:2:0 zu beachten?

Selbsttestaufgabe 2.2-4:

Geben Sie die Abtastfrequenz der Luminanz für ein 625-Zeilen-System nach der ITU-R BT.601 Norm an! Begründen Sie die Wahl dieser Abtastfrequenz!

Selbsttestaufgabe 2.2-5:

Wieso wird bei der Quantisierung nicht der komplette Coderaum von 8 bzw. 10 Bit für die Signalpegel genutzt?

2.3 Digitale Bildübertragungssignale und Schnittstellen in der professionellen Bildtechnik

Basierend auf dem digitalen Abtaststandard ITU-R BT.601 gibt es eine Schnittstellen- und physikalische Übertragungsnorm, welche in der ITU-R BT.656 [2.6] definiert ist. Diese Norm definiert sowohl eine parallele als auch eine serielle Übertragung von digitalen Video- und Audiodaten in Studioqualität.

2.3.1 Die parallele Bildübertragung

Bei der Parallelübertragung mit 8 oder 10 Bit breiten Datenworten wird pro Bit je ein Leitungspaar verwendet. Zusätzlich ist eine ebenfalls symmetrisch ausgelegte Taktleitung (Taktrate: 27 MHz) notwendig. Nimmt man die Masseleitung hinzu, ergibt sich für das Multicore-Kabel eine Kapazität von 23 Leitungen. Die hohe Anzahl von Adern führt zu meist recht unflexiblen und teuren Kabeln. Zudem können mit diesem Standard nur recht beschränkte Entfernungen (ca. 50 m) überbrückt werden. Diese Einschränkungen haben dazu geführt, dass die parallele Norm in modernen Umgebungen nur selten verwendet wird.

2.3.2 Die serielle Bildübertragung

Serial Digital Interface – SDI

Die serielle Norm (*Serial Digital Interface – SDI*), auch als ANSI/SMPTE 259M standardisiert, ist so ausgelegt, dass eine Datenübertragung über herkömmliche 75 Ohm-Koaxialleitungen möglich ist. Demnach kann die bisherige Kabelinfrastruktur der analogen Studioumgebungen ohne Änderung weiter genutzt werden. Das Interface kann gleichermaßen Datenworte mit 8 oder 10 Bit Breite verarbeiten. Die parallel vorliegenden Datenworte werden in einem Multiplexer seriell gewandelt. Die Verwürfelung der NRZI-Codierung[7] (*Non Return to Zero Inverse*) gewährleistet eine Übertragung ohne zusätzlichen Bit-Takt (*SNRZI – Scrambled Non Return to Zero Inverse*). Die Gesamtdatenrate beträgt maximal 270 Mbit/s. Dieses Signal wird oft auch *DSC-270* genannt (*DSC – Digital Serial Component*, 270 Mbit/s). Es benötigt mit der o.a. Codierung eine genutzte Bandbreite im Kabel von mindestens 135 MHz. Die geringe Dämpfung der Koaxialleitungen macht Leitungslängen von gut 250 m möglich, ohne dass weitere Einrichtungen zur Verstärkung (*Taktrückgewinnung, Repeater* o.ä.) erforderlich wären.

Die Aufbereitung der 4:2:2-Komponentensignale *Y*, C_B und C_R zu einem Zeitmultiplexsignal mit 8 bzw. 10 Bit breiten Datenstrom geschieht über eine Zeitkom-

7 Bei der NRZI-Codierung wird die logische „0“ als ein Gleichspannungswert (z. B. +400 mV oder -400 mV) und eine logische „1“ als ein Gleichspannungssprung (von +400 mV nach -400 mV oder umgekehrt) codiert. Insgesamt liegt die Spannung so bei 0.8 V_{ss}.

pression der *Y*-, C_B- und C_R-Signale in einem Multiplexer, der mit der doppelten Taktrate von

$$f_{SDI,Multiplex} = 27\ MHz,\ bzw.\ T_{SDI,Multiplex} \approx 37.037\ ns. \qquad 2.3\text{-}1$$

C_B-Y_1-C_R-Y_2-Multiplex

In einer 4er-Gruppe werden nacheinander die KomponentensignaleC_B-Y_1-C_R-Y_2 übertragen. Abb. 2.3-1 veranschaulicht den zugehörigen Multiplexprozess.

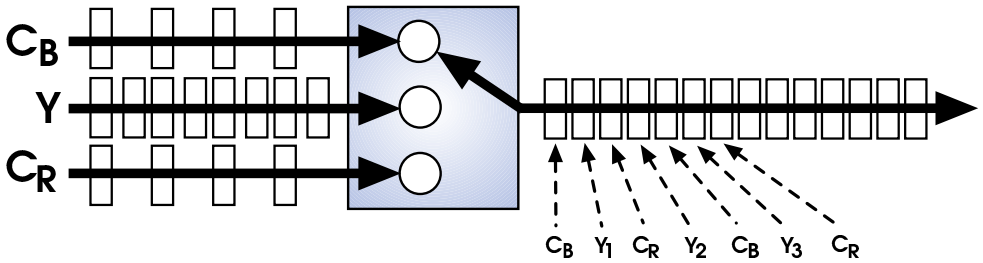


Abb. 2.3-1: Erzeugung des C_B-Y_1-C_R-Y_2-Multiplex-Signals

Eine so zusammengefasste Bildzeile besitzt 1440 aktive Abtastwerte und benötigt in Analogie zu Gl. 2.2-9:

$$T_{Z,\perp} = 1440 \cdot T_{SDI,Multiplex} = \frac{1440}{27\ MHz} \approx 53.33\ \mu s. \qquad 2.3\text{-}2$$

Abb. 2.3-2 zeigt die zeitlichen Aufteilung innerhalb der digitalen Videozeile und gibt den Zusammenhang mit der entsprechenden analogen Bildzeile an.

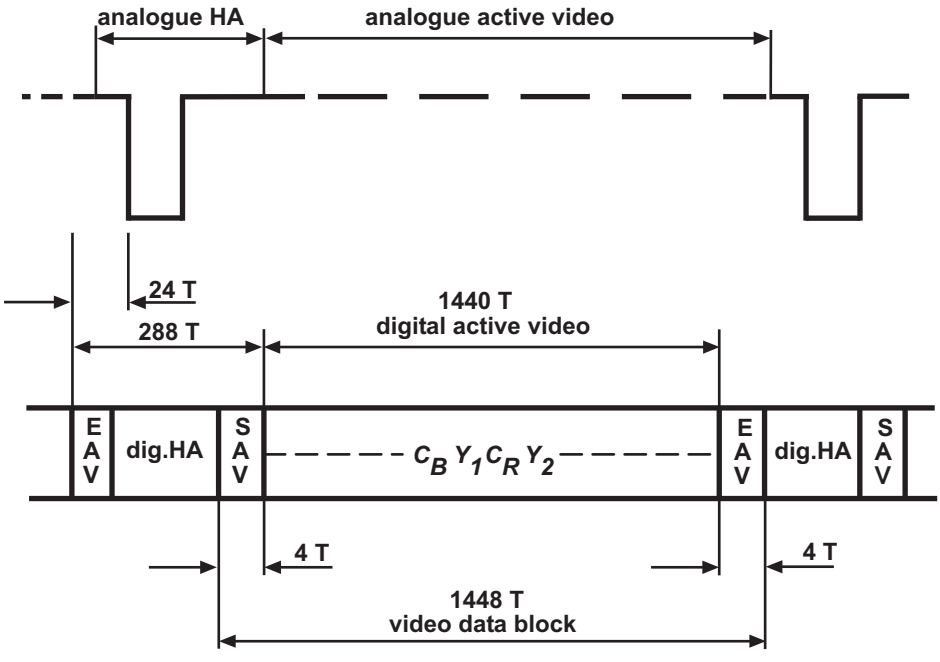


Abb. 2.3-2: Zusammenhang zwischen analoger und digitaler Bildzeile (hier: $T = T_{Z,\perp}$)

Mit der 4:2:2 Farbabtastung ergibt sich die erforderlich Datenrate für den aktiven Teil des digitalen Studiovideosignals (625-Zeilen Norm) zu

$$R_{SDI,aktiv,8Bit} = 2 \cdot 720 \, \frac{Pixel}{Line} \cdot 576 \, \frac{Line}{Frame} \cdot 25 \, \frac{Frame}{s} \cdot 8 \, \frac{Bit}{Pixel}$$
$$= 165888000 \; Bit/s = 158.2 \; Mbit/s \qquad 2.3\text{-}3$$

und

$$R_{SDI,aktiv,10Bit} = 2 \cdot 720 \, \frac{Pixel}{Line} \cdot 576 \, \frac{Line}{Frame} \cdot 25 \, \frac{Frame}{s} \cdot 10 \, \frac{Bit}{Pixel}$$
$$= 207360000 \; Bit/s = 197.8 \; Mbit/s. \qquad 2.3\text{-}4$$

SAV – Start of Active Video
EAV – End of Active Video

Für die Synchronisation sind die je 4-Codeworte breiten *ZeitreferenzsignaleSAV* (*Start of Active Video*) und *EAV* (*End of Active Video*) definiert. Die ersten drei Codeworte dienen zur Empfängersynchronisation, wobei das erste 8- bzw. 10-Bit breite Codewort nur aus Einsen besteht, die beiden folgenden aus Nullen. Das vierte Wort beinhaltet die eigentliche Information über Halbbildnummer, über die Lage innerhalb oder außerhalb der vertikalen Austastlücke und ob es sich um den Start (*SAV*) oder das Ende (*EAV*) der aktiven Videozeile handelt. Abb. 2.3-3 skizziert die Einzelheiten der Zeitreferenzsignale für die 8 und 10 Bit Technologie.

	11111111	00000000	00000000	1	F	V	H	P_3	P_2	P_1	P_0		

8 Bit

	1111111111	0000000000	0000000000	1	F	V	H	P_3	P_2	P_1	P_0	0	0

10 Bit

F: "0" Zeile aus 1. Halbbild, "1" Zeile aus 2. Halbbild
V: "0" außerhalb, "1" während vertikaler Austastlücke
H: "0" in SAV und "1" in EAV
P0...P3: Schutz- und Parity-Bits

Abb. 2.3-3: Zeitreferenzsignale SAV und EAV für 8 Bit und 10 Bit Codierung

Nutzung der Austastlücken

Ähnlich wie in der analogen Bildtechnik sind auch in der digitalen Bildtechnik Zeitspannen für eine horizontale und eine vertikale Austastlücke vorgesehen. Gemäß der Schnittstellennorm ITU-R BT.656 können diese Zeitspannen zusätzliche Informationen wie beispielsweise digitalen Begleitton oder Timecodesignale beinhalten. Das DSC-270 Signal besitzt eine Brutto-Bitrate von 270 Mbit/s. Der aktive Bildanteil benötigt gemäß Gl. 2.3-4 selbst bei einer 10 Bit Quantisierung nur knapp 200 Mbit/s. Es existiert also eine Reserve von etwa 70 Mbit/s für zusätzliche Daten im SDI-Signal. Die nachfolgende Abb. 2.3-4 zeigt die Bereiche sowie die nutzbare Datenrate in den beiden Austastlücken.

Embedded Audio

In der Praxis werden innerhalb der horizontalen Austastlücke etwa 6 Mbit/s benötigt, um 4 Audiokanäle nach dem *AES/EBU*-Standard[8] (48 kHz Abtastfrequenz, 20 Bit Quantisierung) einzubetten [2.10]. Das spart eine zusätzliche Audioverkabelung. Da in vielen Studios die Verarbeitung von Bild und Ton jedoch örtlich getrennt sind, wird von dieser Möglichkeit oft nicht Gebrauch gemacht, sondern weiterhin die existierende Audioverkabelung auch für AES/EBU-Signale benutzt.

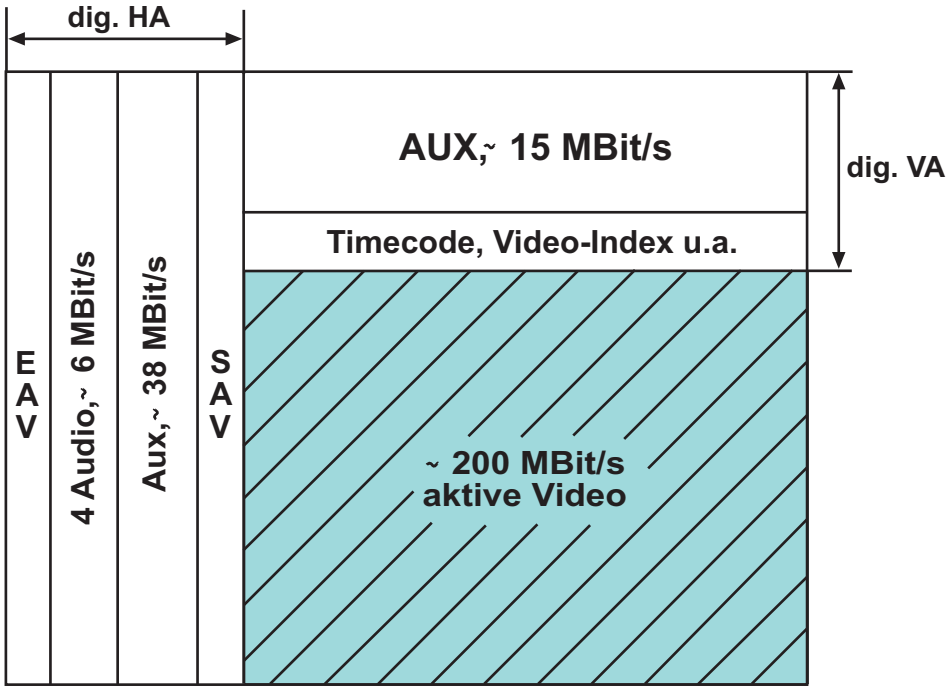


Abb. 2.3-4: Bereiche für Zusatzdaten im DSC-270 Studiosignal (10 Bit Video)

Für die Timecode-Verkopplung im Studio bietet sich die zusätzliche Einlagerung eines Timecode-Signals in der vertikalen Austastlücke an. Prinzipiell lassen sich noch eine Menge weiterer Info-Daten unterbringen. Hierzu gibt es aber bislang keine einheitlichen Absprachen oder Normen.

8 Die AES/EBU-Norm beschreibt das gebräuchliche Signal in der digitalen professionellen Audiotechnik. Nähere Einzelheiten dazu lassen sich in [2.7] nachlesen.

2.3.3 Hochauflösende Bildübertragung

Das Zusammenwachsen zwischen Kinofilm- und Videotechnik hat dazu geführt, dass sich auch hochauflösende, digitale Videonormen unter dem Oberbegriff „*HDTV*“ (*High Density Television*) etabliert haben. Die zum ITU-R BT.601 Standard korrespondierende Norm für die digitale Repräsentation von hochaufgelösten Bildsignalen lautet ITU-R BT.709. Eine entsprechende Schnittstellennorm wird mit dem Namen „*HD-SDI*“ (*High Density – Serial Digital Interface*) bezeichnet und ist u. a. als ITU-R BT.1120 [2.8] bzw. SMPTE 292M bekannt.

Bezüglich der Wahl der Abtastfrequenz für das Y-Signal hat man beim HD-Videosignal folgende Überlegungen angestellt: Die Anzahl der horizontalen Abtastpunkte und die Zeilenzahl sollen jeweils um den Faktor zwei erhöht werden.

Abtastfrequenz HD-SDI

Darüber hinaus sollen horizontal zusätzlich so viele Abtastwerte hinzugefügt werden, wie für den Übergang des Bildseitenverhältnisses von 4:3 auf 16:9 notwendig sind. Es resultiert daraus eine Erhöhung der Abtastfrequenz um den Faktor

$$r_{\perp,SD\rightarrow HD} = 2 \cdot 2 \cdot \frac{3 \cdot 16}{4 \cdot 9} = \frac{16}{3} \approx 5.33. \qquad \text{2.3-5}$$

Dieser Wert hat sich nicht durchsetzen können. Vielmehr wurde ein Faktor

$$r_{\perp,SD\rightarrow HD} = 5.5 \qquad \text{2.3-6}$$

einheitlich für 50 Hz und 60 Hz-Umgebungen gewählt, der zu folgenden Abtastfrequenzen für HD-Signale führt:

$$\begin{aligned} Y &: f_{\perp,HD} = 22 \cdot f_{\perp,Basis} = 5.5 \cdot f_{\perp,SD} = 5.5 \cdot 13.5\ MHz = 74.25\ MHz, \\ C_B, C_R &: f_{\perp,HD,\downarrow 2} = \frac{f_{\perp,HD}}{2} = 37.125\ MHz. \end{aligned} \qquad \text{2.3-7}$$

Für die 4:2:2 Farbabtastung wird wieder der von SDI her bekannte C_B-$Y1$-C_R-$Y2$ Multiplex verwendet. Damit ergibt sich dann für HD-Video eine Interfacedatenrate von

$$D_{HD-SDI,nominell} = 5.5 \cdot 270\ Mbit/s = 1485\ Mbit/s. \qquad \text{2.3-8}$$

HD-Aufnahme Szenario

Abb. 2.3-5 zeigt die Übertragungskette von der Aufnahme bis hin zum HD-SDI Signal.

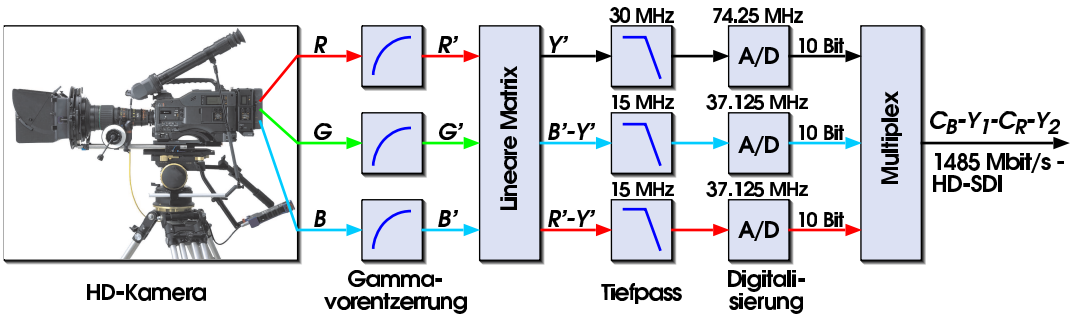


Abb. 2.3-5: Von der analogen RGB-Quelle zum seriellen HD-SDI Signal

Das im ITU-R BT.709 vorgeschlagene Abtastformat für die hochauflösende Videotechnik wird in vielen praktischen Anwendungen nicht verwendet. Vielmehr haben sich je nach Anwendungsfall eine Reihe von proprietären HD-Abtastformate durchgesetzt. Obwohl daraus recht unterschiedliche Datenraten resultieren, werden die Signale oftmals dennoch einheitlich über eine HD-SDI-Schnittstelle nach ITU-R BT.1120 verteilt.

Im Unterschied zu den Überlegungen bei der Abtastung nach der ITU-R BT.601 Empfehlung werden bei der HD-Abtastung oft für die 50 Hz- und 60 Hz-Umgebung jeweils die gleiche Pixelanzahl horizontal und vertikal angestrebt. Darüber hinaus gelten neben dem 16:9 Bildseitenverhältnis auch quadratische Pixel als zwingend notwendig, um computergenierte Inhalte verzerrungsfrei einbetten zu können.

Dieser Ansatz führt schließlich zu zwei häufig eingesetzten Standards innerhalb der *ANSI / SMPTE 274M*- und *296M*-Norm:

Gebräuchliche HD-Abtastraster

- Progressive Abtastung mit 1280 x 720 aktiven Pixeln
- Herkömmliche Zeilensprungabtastung (*Interlace*) mit 1920 x 1080 Pixeln.

Die nachfolgende Tabelle 2.3-1 fasst die wichtigsten Eigenschaften für diese beiden Abtastformate zusammen.

Tab. 2.3-1: *Wesentliche Abtastparameter von HD-Videosystemen*

Standard gemäß	SMPTE 296M		SMPTE 274M	
(Voll-) Bildfrequenz	25 Hz	30 Hz	50 Hz	60 Hz
Abtastung	interlace		progressiv	
$f_\perp$	*Y*: 74.25 MHz; C_B,C_R: 37.125 MHz			
Zeilendauer	35.55 µs	29.63 µs	26.67 µs	22.22 µs
Anzahl Zeilen (nominell)	1125		750	
Anzahl Zeilen (aktiv)	1080		720	
Anzahl Abtastwerte über gesamte Zeilendauer (nominell)	*Y*: 2640 C_B,C_R: 1320	*Y*: 2200 C_B,C_R: 1100	*Y*: 1980 C_B,C_R: 990	*Y*: 1650 C_B,C_R: 825
Anzahl Abtastwerte je aktiver Zeile	1920 (25.86 µs)		1280 (17.24 µs)	

2.3.4 Bildformate in der Computerbilddarstellung

Abtastraster nach VESA und ITU

In Computerumgebungen sind die Auflösungsverhältnisse oft nach den Vorschlägen der *VESA* (*Video Electronic Standard Association*) realisiert worden. Dabei wird abweichend von einigen digitalen Fernsehstandards von einem Bildseitenverhältnis von 4:3 bzw. 5:3 ausgegangen, die Pixel werden als quadratisch angesehen und die Abtastung selbst erfolgt grundsätzlich progressiv. Die Bildwiederholfrequenz ist nicht starr festgelegt, sondern variiert in einem Bereich zwischen 60 Hz und über 100 Hz.

In der nachfolgenden Tabelle 2.3-2 sind die wichtigsten Auflösungsformate nach *VESA* zusammengefasst dargestellt, ergänzt durch einige wichtige Vertreter aus der professionellen Videotechnik.

Tab. 2.3-2: *Computerabtastraster im Vergleich zu digitalen Videostandards*

Bezeichnung	Anzahl Bildpunkte H x V	Bildseitenverhältnis	empfohlene Monitor Bilddiagonale [Inch]
VGA	640 x 480	4:3	14“
SD - 720i[9]	720 x 576	5:4	
(SDTV)	768 x 576	4:3	15“
SVGA	800 x 600	4:3	15“
XGA	1024 x 768	4:3	16“
	1152 x 864	4:3	17”
HD - 720p[10]	1280 x 720	16:9	
(SXGA)	1280 x 960	4:3	17”
SXGA	1280 x 1024	5:4	17”
UXGA	1600 x 1200	4:3	19”
(UXGA)	1800 x 1440	4:3	20“
HD - 1080i/p[11]	1920 x 1080	16:9	
QXGA	2048 x 1536	4:3	21“

Für die Übertragung der Videodaten des Computers zum Monitor haben sich ebenfalls eine Reihe von digitaler Schnittstellenstandards etabliert. Die wichtigsten werden nachfolgend kurz vorgestellt.

PanelLink, TMDS

PanelLink und Transition Minimized Differential Signalling (*TMDS,* digital)

Der *PanelLink*-Standard bildet die Grundlage für viele digitale Schnittstellen zur Ansteuerung von Grafikdisplays. Dabei basiert dieses System auf dem *TMDS*-Verfahren, bei dem 4 Doppeladern für das digitale R-, G-, B- und Taktsignal genutzt

9 nach ITU-R BT.601; das Bildseitenverhältnis würde gelten, wenn quadratische Pixel angenommen werden, sonst ist das Bildseitenverhältnis von 4:3 definiert; vgl. auch Abschnitt 2.2.2; i. a. Zeilensprungabtastung

10 nach SMPTE 274M

11 nach SMPTE 296M

werden. Man nennt eine solche Verbindung auch *TMDS*-Link. Der Pixeltakt auf einem solchen *TMDS*-Link kann bis zu 165 MHz betragen. Die symmetrischen Pegel von $\pm$ 500 mV sorgen für eine störungsfreie Übertragung auch über größere Distanzen hinweg.

P&D ***Plug & Display-Port* (*P&D* , analog/digital)**

P&D ist ein Interfacestandard für analoge und digitale Videoübertragung und wurde 1997 von der *VESA* standardisiert. Die Besonderheit liegt in der zusätzlichen, optionalen Integration der seriellen Schnittstellen *USB* und *IEEE 1394*, welche für viele Anwendungen in Computerumgebungen genutzt werden. Vorteilhaft ist, dass bisherige analoge Monitore per Adapterstecker weiter verwendet werden können, sofern sie die standardisierte Gerätekennung über den *Data Display Channel* (*DDC*) beherrschen. Viele moderne Monitore können ihre Geräteidentifikation auch schon über das veraltete 15-polige VGA-Interface an den Computer senden. Die Integration der beiden zusätzlichen Schnittstellen *USB* und *IEEE 1394* führt zu relativ hohen Preisen bei dieser Schnittstellentechnik. Daher konnte sich dieses Interface nicht auf breiter Ebene durchsetzen.

DFP ***Digital Flat Panel Port* (*DFP*, digital)**

DFP stellt eine abgespeckte Version der *P&D-Interface* Norm dar und wurde 1999 standardisiert. Es werden lediglich digitale Signale berücksichtigt und eignet sich daher besonders für rein digital ansteuerbare Pixeldisplays (z. B. LCD-Panel). Hierbei muss allerdings berücksichtigt werden, dass der maximale Pixeltakt lediglich 85 MHz betragen darf. Damit ist je nach Bildwiederholrate die Auflösung auf dem Display auf *SXGA* bzw. *SVGA* beschränkt.

DVI ***Digital Visual Interface* (*DVI*, analog/digital)**

DVI ist ein 1999 von der *Digital Display Working Group* (*DDWG*) etablierter Standard, der weit verbreitet ist [2.9]. Er kann analoge und digitale Signale verarbeiten und ist für digitale Quellen abwärts kompatibel zu *P&D* und *DFP*. Im Unterschied zu diesen Standards sind bei *DVI* jedoch zwei *TMDS*-Links integriert, auf die der digitale Videodatenstrom gleichmäßig verteilt werden kann. Es ergibt sich ein maximaler Pixeltakt von 330 MHz, mit dem sich *QXGA*-Auflösungen mit einer Bildwiederholfrequenz von 85 Hz realisieren lassen.

2.3.4.1 *High Definition Multimedia Interface* (*HDMI*, digital)

HDMI

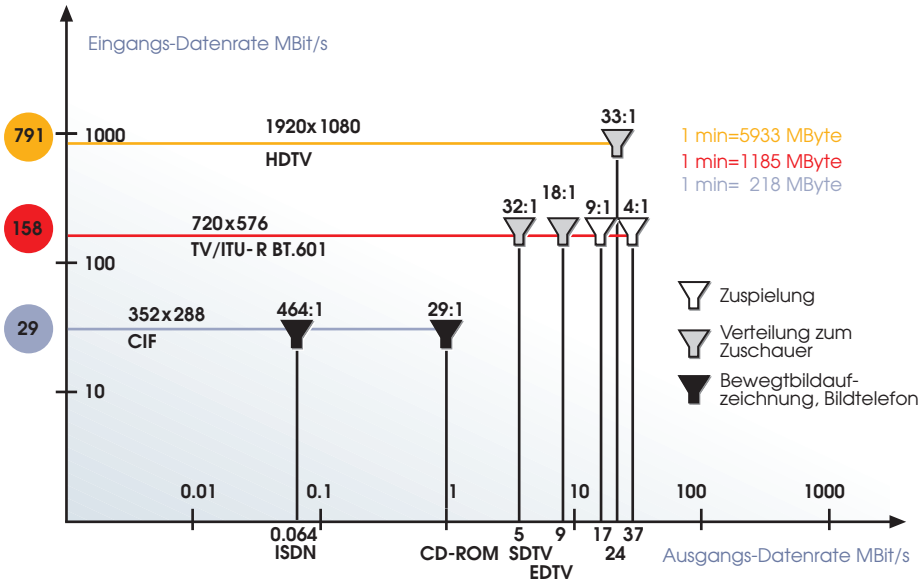
HDMI wurde 2002 von den Firmen Hitachi, Matsushita (Panasonic), Philips, Silicon Image, Sony, Thomson und Toshiba in der *HDMI-1.0*-Spezifikation zusammengefasst. Bei *HDMI* handelt es sich um eine zur *DVI*-1.0-Schnittstelle abwärtskompatible Schnittstelle für hochaufgelöste digitale Audio- und Videoübertragung zwischen Endgeräten. Die maximale Bandbreite liegt bei ca. 5 GBit/s und ist damit auch für die anspruchvolle Bildkommunikation geeignet. Um den Copyright-Anliegen der Filmindustrie gerecht zu werden, ist innerhalb des Standards die so genannte *High-bandwidth Digital Content Protection* (*HDCP*) eingefügt worden, welche für eine verschlüsselte Übertragung sorgt.

2.4 Daten- und Kompressionsraten – Motivation für die Bilddatenkompression

Im letzten Abschnitt wurde gezeigt, dass digitale Videodatenströme mit einer Standardbildauflösung von 720 x 576 Bildpunkten eine Nettodatenrate von 158 Mbit/s für eine 8 Bit Quantisierung (vgl. Gl. 2.3-4), für 10 Bit Quantisierung sogar eine Datenrate von 200 Mbit/s benötigen (vgl. Gl.2.3-3). Hochaufgelöste Videobilder mit 1920 x 1080 Bildpunkten erfordern netto bei 8 Bit Quantisierung eine Datenrate von 791 Mbit/s, brutto (einschließlich Audio, Austastlücken etc.) bei 10 Bit Quantisierung sogar bis zu 1485 Mbit/s (vgl. Gl. 2.3-8).

Es ist unmittelbar einsichtig, dass diese großen Datenmengen auf konventionellen Übertragungsmedien nicht wirtschaftlich transportiert werden können. Die Übertragungskapazität eines typischen digitalen Satellitentransponders liegt beispielweise bei etwa 40 Mbit/s. Ähnliche Werte gelten auch für die Verbreitung über Kabelnetze. Bei der digitalen terrestrischen Ausstrahlung liegt die korrespondierende Kapazität innerhalb eines konventionelle Fernsehkanals nur bei etwa der Hälfte, 20 Mbit/s.

Kompressionsverhältnisse



Animation 2.4-1: Kompressionsverhältnisse bei der Videodatenraten-Reduktion (nach [2.3])

Diese Diskrepanz zwischen Datenaufkommen und Kapazität führt dazu, dass das digitale Videosignal für die Übertragung mittels eines geeigneten Datenreduktionsverfahrens deutlich *komprimiert* werden muss. Animation 2.4-1 veranschaulicht die notwendige Videodaten-Reduktion für drei typische Videosignalformate[12] als Eingangssignal und die Übertragungskapazität einiger Übertragungsmedien.

In Fällen, in denen eine reduzierte Bildgüte (z. B. *VHS*-Qualität) ausreichend ist, ist es sinnvoll, bereits für das Eingangsmaterial ein Format mit geringerer Auflösung zu verwenden. Hierzu bietet sich beispielsweise das *CIF – Common Interchange Format* mit einer Auflösung von 352 x 288 Bildpunkten (Animation 2.4-1, blau) an. Material in diesem Format wird meist progressiv mit 25 Vollbildern pro Sekunde abgetastet ist. Die Unterabtastung der Farbe geschieht im Verhältnis von 4:2:0. Selbst mit der daraus resultierenden Datenrate von 29 Mbit/s ist für einen *ISDN*-Übertragungskanal noch eine erhebliche Kompression um den Faktor 464 notwendig (symbolisiert durch entsprechend gekennzeichnete Trichter). Bei einer Übertragung mit einer 1-fachen *CD-ROM* Datenrate von ca. 1 Mbit/s ist noch eine Kompression um den Faktor 29 notwendig.

Ein Standardfernsehsignal nach *ITU-R BT.601* muss um den Faktor 32 komprimiert werden, um Platz in einem 5 Mbit/s Datenkanal zu haben. Mit Standard-Kompressionsalgorithmen, wie sie in Kapitel 4 vorgestellt werden, lässt sich damit Fernseh-Standardqualität (*SDTV – Standard Density Televion*) erreichen (Animation 2.4-1, rot). Für eine höhere Qualität (*EDTV – Enhanced Density Television*) oder eine Zuspielung zwischen Studios darf die Datenrate der Signale nur etwa um den Faktor 15 bis 18 reduziert werden.

12 Netto-Eingangs-Datenrate nur für 8 Bit quantisiertes Video mit 1024 Bit = 1kbit; 1024 kbit = 1 Mbit

Wird davon ausgegangen, dass eine Kompressionsrate von etwa 1:33 für die Erreichung einer Standardqualität beim Zuschauer ausreichend ist, muss für ein hochaufgelöstes Bildsignal (*HDTV – High Density Television*) ein Kanal mit mindestens 24 Mbit/s Kapazität vorhanden sein (Animation 2.4-1, orange).

Bei der Reduktion bzw. Kompression der Bilddatenrate lassen sich zwei Fälle unterscheiden:

1. Reduktion der *Redundanz*,
2. Reduktion der *Irrelevanz*.

Verlustlose Codierung

Unter Redundanz wird der Anteil des Signals verstanden, welcher mit einem geeigneten Verfahren so entfernt werden kann, dass er exakt, d. h. verlustlos ohne die Verwendung von zusätzlicher Information am Empfänger rekonstruiert werden kann. Man nennt daher eine Redundanzreduktion auch eine *verlustlose Codierung*.

Verfahren zur Redundanzreduktion sind auch außerhalb der digitalen Bildcodierung weit verbreitet. Überall dort, wo es um das Herausnehmen bereits bekannter Information aus dem Datenfluss geht, wird eine Redundanzreduktion vorgenommen. Ein Beispiel für Redundanzverfahren ist die *variable Lauflängencodierung* (z. B. *Huffman-Codierung*), die im Kapitel 3 behandelt wird.

Unter der Irrelevanzreduktion in der Bildcodierung wird ein Weglassen von Information verstanden, die in Bezug auf das menschliche Sinnesorgan Auge unwesentlich oder nicht wahrnehmbar ist. In Kapitel 1wurden hierzu bereits einige Eigenschaften des visuellen Systems aufgeführt, die erfolgreich für eine Irrelevanzreduktion genutzt werden können (z. B. Verdeckungseffekte).

Redundanz und Irrelevanz

Abb. 2.4-2 skizziert den Zusammenhang der beiden Bereiche anschaulich. Das Datenaufkommen eines Eingangsbildes ist symbolisch aufgeteilt dargestellt in vier Bereiche.

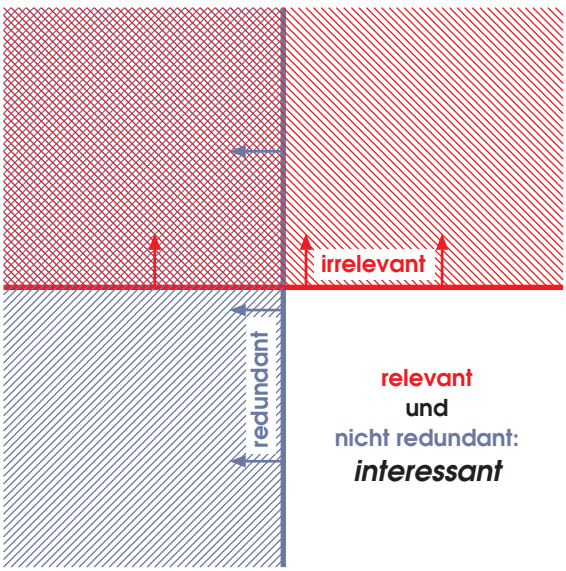


Abb. 2.4-2: Irrelevanz und Redundanz

Horizontal wird die Aufteilung in redundante und nicht redundante Information gezeigt. Vertikal kommt die Aufteilung in irrelevante und relevante Informationen hinzu. Interessant und damit für die Übertragung tatsächlich notwendig ist der Teil der Daten, welche sowohl relevant als auch nicht redundant sind (vgl. Abb. 2.4-2 rechts unten).

Selbstlernaufgabe 2.6

Erläutern Sie kurz den Aufbau und die Funktion der Zeitreferenzsignale SAV und EAV für eine 8 Bit Quantisierung!

Selbstlernaufgabe 2.7

Für welche Daten können die horizontale und vertikale Austastlücke beim DSC-270 Studiosignal genutzt werden?

Selbstlernaufgabe 2.8

Auf welchen Auflösungsstandard muss ein Computermontiorbild mindestens eingestellt sein, um ein Videobild nach *ITU-R BT.601* Bildpunkt genau darstellen zu können? Ist die Darstellung verzerrungsfrei, wenn jedem Pixel des digitalen Videobildes ein Pixel auf dem Computermonitor zugeordnet wird? Begründung!

Selbstlernaufgabe 2.9

Definieren Sie die Begriffe Redundanz und Irrelevanz!

Selbstlernaufgabe 2.10

Es sollen Daten aus einem Computerprogramm datenreduziert übertragen werden. Welches der beiden Verfahren *Redundanzreduktion* und *Irrelevanzreduktion* ist für diese Aufgabe geeignet? Begründen Sie Ihre Antwort!

3 Digitale Quellencodierung für Einzelbilder

Ein klassisches Übertragungssystem für digital codierte Bildsignale skizziert Abb. 3-1. Das hier aufgezeigte Modell der Nachrichtenübertragung gilt nicht nur für die Bildcodierung, sondern kann prinzipiell auch für andere Anwendungen genutzt werden.

Modell Nachrichtenübertragungskette

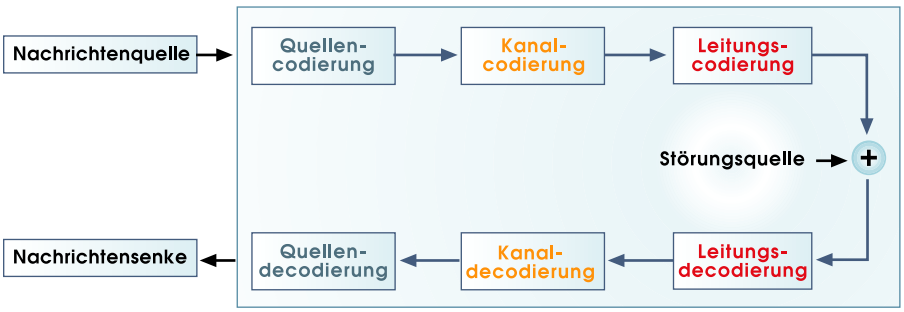


Abb. 3-1: Digitale Übertragung – von der Nachrichtenquelle bis hin zur Nachrichtensenke

Quellencodierung

Den einzelnen Blöcken kommen dabei jeweils individuelle Aufgaben zu. Die *Quellencodierung* befreit die Daten der diskreten Bilddaten-Quelle zunächst von *Redundanz* und *Irrelevanz* gemäß der Vorüberlegungen aus Abschnitt 2.4. Die nachfolgenden Abschnitte (Abschnitt 3.1 bis Abschnitt 3.5) beschäftigen sich mit der Quellencodierung von Einzelbildern.

Kanalcodierung

In der *Kanalcodierung* wird nun unter Berücksichtigung der zu erwartenden Kanaleigenschaften (Bandbreite, Verzerrungen, Rauschen etc.) gezielt Redundanz hinzugefügt, um auch bei einer gestörten Übertragung dennoch eine fehlerfreie Rekonstruktion der Quelldaten am Empfänger erreichen zu können. Die Kanalcodierung wird daher oft auch als *Fehlerschutzcodierung* bezeichnet. Die bewusste Trennung von Quellencodierung und Kanalcodierung ermöglicht eine weitgehend unabhängige Anpassung des Übertragungssystems an die Eigenschaften der Nachrichtenquelle und -senke sowie an die Eigenschaften des Übertragungskanals selbst. Abschnitt 6.4 beschäftigt sich mit dieser Thematik.

Leitungscodierung

Die *Leitungscodierung* sorgt für eine adäquate Aufbereitung des Signals in Bezug auf die physikalische Übertragung. Zu den Hauptaufgaben der Leitungscodierung gehören die Multiplexbildung der Datensignale, die *Impulsformung* und die *Frequenzumsetzung*. Weitere Einzelheiten werden exemplarisch für den *DVB*-Standard in Abschnitt 6.5 behandelt.

Das so aufbereitete Signal wird nun über einen gestörten Kanal übertragen. Auf der Empfangsseite werden die auf der Sendeseite durchgeführten Codierschritte in umgekehrter Reihefolge wieder rückgängig gemacht. Als Ergebnis erhält man das decodierte Nachrichtensignal, das sich in Bezug auf die Nachrichtensenke nur unwesentlich von dem Signal der Nachrichtenquelle unterscheiden soll.

3.1 Transformation als Basis der digitalen Quellencodierung

3.1.1 Motivation

Nach den Überlegungen in Abschnitt 2.4 wird deutlich, dass das effektive Datenaufkommen für die Übertragung eines Bildes oft um mehr als den Faktor 30 reduziert bzw. *komprimiert* werden muss. Es wird also ein Algorithmus gesucht, der eine Bilddatei auf bis zu etwa 3 % ihrer originären Größe schrumpfen lässt, ohne dass die subjektiv empfundene Qualität allzu sehr darunter leidet.

In praktischen Versuchen mit natürlichen Bildvorlagen kann man schnell feststellen, dass eine verlustlose *Redundanzreduktion* nur zu einer Verringerung des Datenaufkommens auf etwa 60 bis 80 % führt. Mit solchen Verfahren kann also allein keine ausreichende Kompression erreicht werden. Ein erster Schritt in Richtung *Irrelevanzreduktion* wird bei der Reduzierung der Farbauflösung (vgl. Abschnitt 2.2.1: *Farbunterabtastung*) gemacht. Weiterhin wäre auch eine stärkere Quantisierung oder eine örtliche Unterabtastung (*Diskretisierung*) möglich. Während die Reduktion der Farbauflösung in Grenzen meist nicht wahrnehmbar ist, führt die gröbere Quantisierung oder eine weitere örtliche Unterabtastung sehr schnell zu deutlich sichtbaren Fehlern, wie in Kapitel 2 gezeigt wurde (vgl. Abb. 2.1-3 bis Abb. 2.1-5).

Dekorrelation

Ersichtlich kann mit den zuvor genannten Verfahren eine effektive Bilddatenkompression nicht vorgenommen werden, da örtlich benachbarte Pixel zwar zueinander *korrelieren*, aber oft keineswegs gleich gesetzt werden dürfen. Es muss also ein Verfahren gefunden werden, das eine *Dekorrelation* des Bildsignals vornimmt, bevor eine Quantisierung und ein Redundanzreduktion angesetzt werden kann. Eine solche Dekorrelation gelingt im allgemeinen dann, wenn man die Bildvorlage zunächst in Blöcke von beispielsweise 8 x 8 Bildpunkten (*Pixeln*) zerlegt und anschließend diese Blöcke einer Transformation unterzieht. Anschaulich kann man sich oft unter dieser Transformation eine Umwandlung des Bildblockes in den Frequenzbereich vorstellen, wenngleich es sich bei vielen diskreten Transformationen nicht um ein exaktes Abbild des Bildblockes im Frequenzbereich handelt.

Transformationscodierer

Codierverfahren, die eine solche Transformation für die Datenkompression verwenden, werden *Transformationscodierer* genannt. Sie haben sich in der heutigen Audio- und Bildcodierung als die effektivsten Verfahren zur Datenkompression erwiesen. In vielen Untersuchungen konnte gezeigt werden, dass die Quantisierung im Transformationsbereich zu einer höchst effizienten Reduktion der Datenrate führen kann, ohne dass sich dabei die subjektive Bildqualität entscheidend verschlechtert.

3.1.2 Eigenschaften der Transformationscodierung

Die Transformationen der verschiedenen Transformationscodierer haben folgende gemeinsame Eigenschaften:

- Es handelt sich immer um lineare Transformationen, d. h. eine Bildzeile $x = x_0, ..., x_{N-1}$ wird immer nach dem Schema **Gesetze zur Transformation**

$$u = \underline{A}\, x \text{ mit } u_i = \sum_{i=0}^{N-1} a_{i,j} x_i \qquad \text{3.1-1}$$

durch eine Transformationsmatrix $\underline{A}$ in eine dekorrelierte Bildzeile $u = u_0, ..., u_{N-1}$ überführt.

- Die Transformationsmatrix $\underline{A}$ ist in der Regel orthogonal, d. h. die Matrix der Rücktransformation $\underline{A}^{-1}$ ergibt sich durch *Transposition* von $\underline{A}$:

$$\underline{A}^{-1} = \underline{A}^{T} \qquad \text{3.1-2}$$

- Die Transformation ist *energieerhaltend*[13].
- Die Transformation wird so ausgewählt, dass die Daten nicht nur dekorreliert werden, sondern die Energie auf wenige Koeffizienten konzentriert wird (*Energy Compaction*). Damit ist der Betrag der meisten Koeffizienten sehr klein. Bei einer späteren Quantisierung können diese Koeffizienten, je nach Anforderung an die Qualität, zu Null gesetzt werden, wodurch ein hoher Kompressionsgrad erzielt werden kann. **Energie Compaction**

Für die Audiocodierung kann die hier beschriebene eindimensionale Transformationscodierung genutzt werden. Dabei wird das eindimensionale Zeitsignal durch die Summe gewichteter Koeffizienten eindimensionaler Zeitsignale mit unterschiedlichen *Basisfunktionen* dargestellt. In der Bildcodierung geschieht die Transformationscodierung, indem einzelne rechteckige Bildausschnitte in Form der Summe von gewichteten Koeffizienten zweidimensionaler, orthogonaler *Basisfunktionen*[14] dargestellt werden.

3.1.3 Auswahlkriterien für Basisfunktionen

Es stellt sich die Frage, wie ein Set von *Basisfunktionen* aussehen muss, damit die gewichtete Summe aus diesen Basisfunktionen wieder dem originalen Audio- bzw. Bildsignal entspricht, oder diesem zumindest recht ähnlich wird. Nachfolgende Abb. 3.1-1 zeigt einige dieser Transformationen und die zugehörigen Basisfunktionen für den eindimensionalen Fall und für eine Blockgröße von 8 x 8 Pixeln. **Basisfunktionen**

13 In einigen wenigen Fällen kann diese Forderung nicht exakt eingehalten werden.

14 Bei der Bildcodierung kann man unter diesen Basisfunktionen anschaulich auch *Basisstrukturen* oder *Basismuster* verstehen.

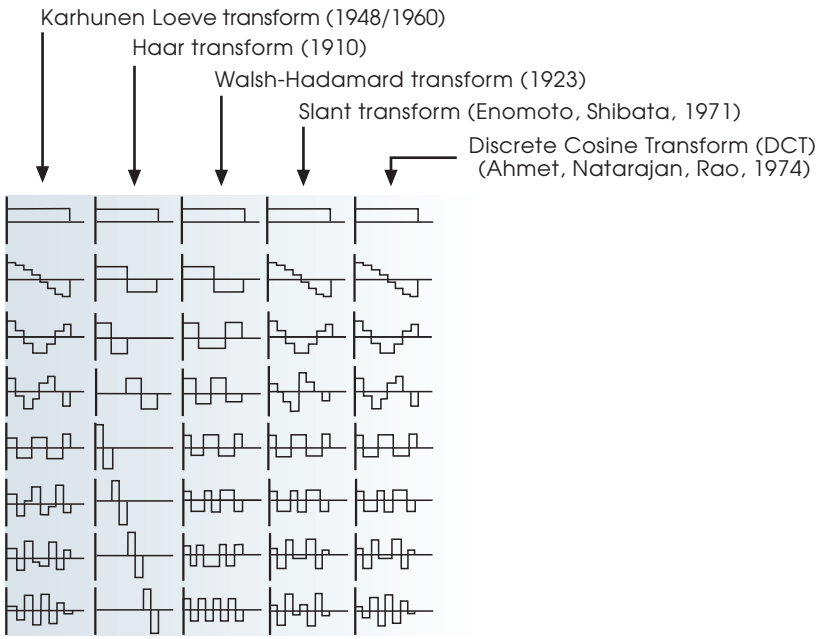


Abb. 3.1-1: Vergleich verschiedener Transformationen anhand eindimensionaler Basisfunktionen

Beispiele Basisfunktionen

Obwohl deutliche Unterschiede zwischen den Transformationen zu erkennen sind, lässt sich für viele Transformationen doch eine Grundregel festhalten: Die angegeben Basisfunktionen repräsentieren von oben nach unten zunehmend höhere Frequenzanteile des Signals.

Bei der Statistik von natürlichen Bildvorlagen ist eine Energiekonzentration hin zu niedrigen Frequenzanteilen feststellbar. Der energetisch überwiegende Teil eines Bildes lässt sich so oft mit wenigen gewichteten und niederfrequenten Basisfunktionen beschreiben.

3.1.4 Von der diskreten Fourier Transformation zur diskreten Cosinus Transformation

Diskrete Fourier Transformation

Wie zuvor beschrieben werden für die Bildcodierung zweidimensionale, *separierbare Basisfunktionen* herangezogen. So errechnen sich bei der so genannten *diskreten Fourier Transformation* (*DFT*) die Basisfunktionen als eine Summe komplexer Exponentialfunktionen:

$$F_{DFT}(u,v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi \frac{x \cdot u}{M}} e^{-j2\pi \frac{y \cdot v}{N}}. \qquad 3.1\text{-}3$$

Dabei sind *u* und *v* die zu *x* und *y* korrespondierenden, transformierten Koordinaten (*Orts-Frequenzen*). *M* und *N* geben die Blockgröße für die Transformation des Bildsignals *f*(*x*, *y*) an.

Bild und die zugehörigen „Frequenzen“

Abb. 3.1-2 veranschaulicht die zur *DFT* ähnliche *Fast Fourier Transformation* (*FFT*) anhand der monochromen Testbildvorlage *Lena*[15].

Abb. 3.1-2: Testbildvorlage *Lena* (links) und Abbild nach der *Fast Fourier Transformation* (rechts)

Die Darstellung der Transformation zeigt wunschgemäß die Energiekonzentration des Bildes in einem relativ kleinen Bereich um den Ursprung herum. In diesem Beispiel ist überdies eine dünne Konzentrationslinie diagonal von links oben nach rechts unten zu erkennen. Sie repräsentiert die Diagonalstrukturen, welche durch den Hut im Bild hervorgerufen werden.

Diskrete Cosinus Transformation

In der Praxis kommt überwiegend die *Diskrete Cosinus Transformation* (*DCT*) zum Einsatz, eine nahe Verwandte der oben beschriebenen *DFT*. Dazu wird nur der Cosinus-Anteil der *DFT* übernommen, was bei der algorithmischen Umsetzung zu Vereinfachungen und damit zu Geschwindigkeitsvorteilen führt. Formal wird für den zweidimensionalen Fall die *DCT* wie folgt ermittelt:

$$F_{DCT}(u,v) = C(u)\cdot C(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cdot\cos\left[\frac{(x+\frac{1}{2})u\cdot\pi}{M}\right]\cos\left[\frac{(y+\frac{1}{2})v\cdot\pi}{N}\right],$$

$$C(u) = \begin{cases} \sqrt{\frac{1}{M}} & \text{für } u = 0 \\ \sqrt{\frac{2}{M}} & \text{für } u \neq 0 \end{cases}; \qquad C(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{für } v = 0 \\ \sqrt{\frac{2}{N}} & \text{für } v \neq 0 \end{cases}.$$

3.1-4

JPEG-Standard

Die DCT bildet die Basis für den 1992 von der *Joint Picture Experts Group* (*JPEG*) verabschiedeten Kompressionsstandard ISO/IEC 10918-1 [3.7] bzw. ITU-T T.81 [3.6] für Stillbilder, kurz *JPEG-Standard* genannt. Die besonderen Eigenschaften dieses Standards liegen darin, dass die Qualität in einem weiten Rahmen gemäß der Anforderungen einstellbar ist. Neben der verlustbehafteten Komprimierung ist innerhalb des Standards auch eine verlustlose Codierung, also eine reine Redundanzreduktion möglich. Es gibt keine prinzipiellen Einschränkungen bezüglich der Anzahl der Gesamtpixel der Bildvorlage (Auflösung). Damit ist der JPEG-Algorithmus praktisch auf beliebiges Bildmaterial anwendbar. Durch die Möglich-

15 Das oft verwendete und weit verbreitete Testbild „*Lena*“ oder „*Lenna*“ ist der Ausschnitt eines auf der Titelseite des *Playboy Magazin* von November 1972 abgebildeten Photos.

keit der sequenziellen Abarbeitung (*Butterfly-Algorithmus*) wird der Rechenaufwand recht gut beherrschbar. Obwohl inzwischen verbesserte Codierverfahren, beispielsweise auf Basis der *Wavelet Transformation* (vgl. Abschnitt 3.4) für besonders niedrige Datenraten verfügbar sind, erfreut sich der JPEG-Algorithmus auch heute noch großer Beliebtheit bei der Einbettung von photografischen Bildern in Web-Pages. Abb. 3.1-3 zeigt alle Funktionsblöcke, die für eine Bildcodierung nach dem JPEG-Standard notwendig sind.

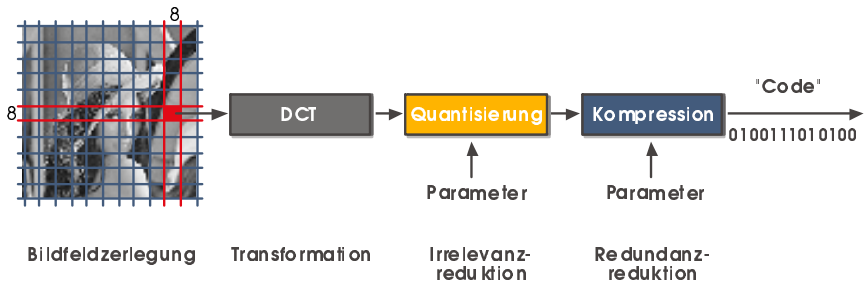


Abb. 3.1-3: Funktionsblöcke der JPEG-Bildcodierung

Die einzelnen Funktionsblöcke werden in den nachfolgenden Abschnitten genauer beschrieben.

3.1.5 Bildfeldzerlegung und DCT-Basisbilder

Bildfeldzerlegung

Beim JPEG-Algorithmus geschieht die für die blockbasierte DCT erforderliche Bildfeldzerlegung in 8 x 8 Pixel große Blöcke. Für ein Bild mit einer Standardauflösung nach ITU-R BT.601 (vgl. Abschnitt 2.2.2) ergeben sich so

$$n_{\text{DCT - Blöcke,601}} = 90 \cdot 72 = 6480 \text{ Blöcke}. \qquad 3.1\text{-}5$$

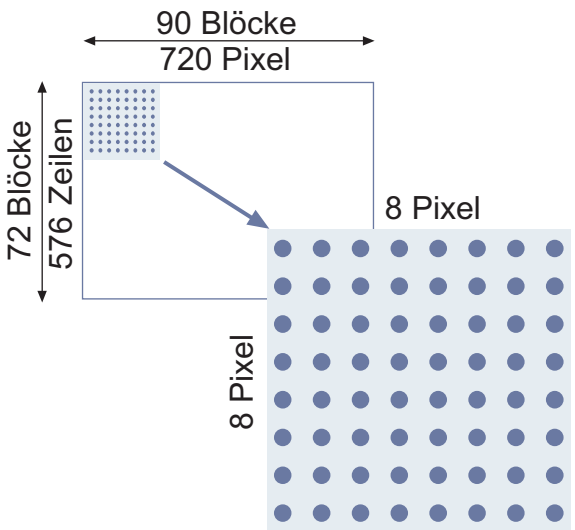


Abb. 3.1-4: Bildfeldzerlegung beim JPEG-Verfahren

Abb. 3.1-4 veranschaulicht den Sachverhalt. Für die Werte $M = N = 8$ ergibt sich Gl.3.1-4 zu

DCT bei JPEG

$$F_{JPEG}(u,v) = \frac{1}{4}C(u)\cdot C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cdot\cos\left[\frac{(2x+1)u\cdot\pi}{16}\right]\cos\left[\frac{(2y+1)v\cdot\pi}{16}\right],$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases} \;;\; C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } v = 0 \\ 1 & \text{für } v \neq 0 \end{cases} . \qquad 3.1\text{-}6$$

Die entsprechende Rücktransformation (*IDCT*) errechnet sich mit $M = N = 8$ zu

Inverse DCT bei JPEG

$$f_{JPEG}(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)\cdot C(v)F(u,v)\cdot\cos\left[\frac{(2x+1)u\cdot\pi}{16}\right]\cos\left[\frac{(2y+1)v\cdot\pi}{16}\right],$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases} ;\; C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } v = 0 \\ 1 & \text{für } v \neq 0 \end{cases} . \qquad 3.1\text{-}7$$

Die Basisbilder der DCT für $M = N = 8$ sind in Abb. 3.1-4 dargestellt.

DCT-Basisbilder

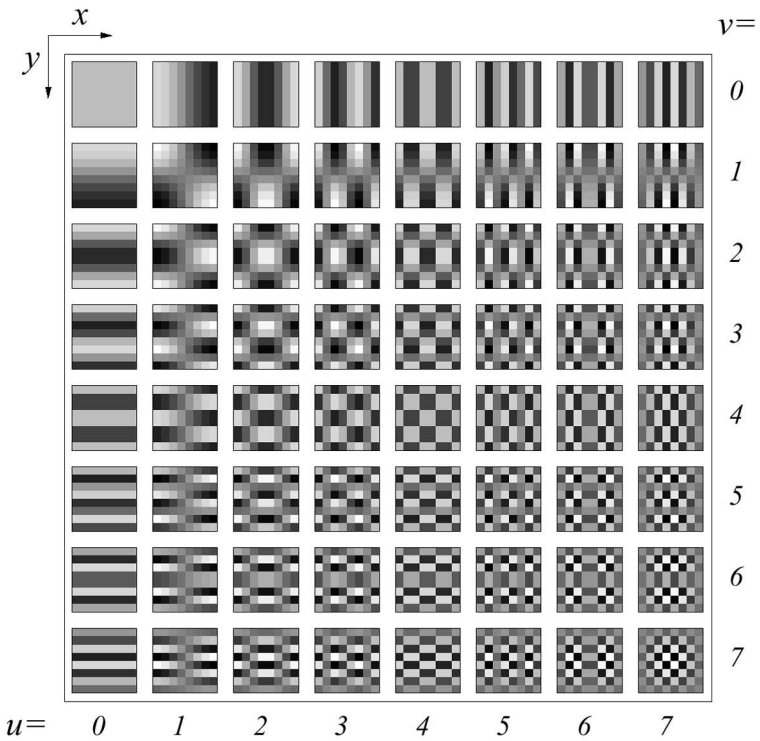


Abb. 3.1-5: DCT-Basisbilder für $M = N = 8$ (*JPEG*)

Es ist zu erkennen ist, dass

DC-Koeffizient, AC-Koeffizienten

- der Koeffizient links oben in der Ecke den *Gleichanteil* des Bildes repräsentiert („*DC-Koeffizient*“),

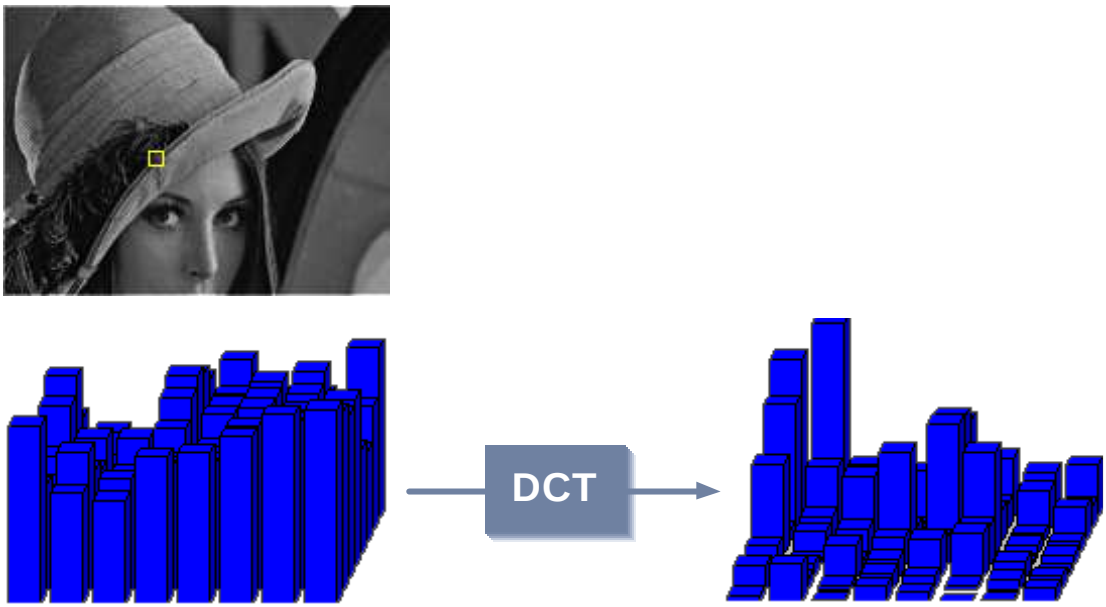
- von links nach rechts die horizontalen Frequenzanteile zunehmen,
- von oben nach unten die vertikalen Frequenzanteile zunehmen.

Die Koeffizienten, die nicht den Gleichanteil repräsentieren, werden *AC-Koeffizienten* genannt.

3.1.6 Beispiele für DCT-Blöcke

DCT-Block Beispiel: Fein strukturierter Bildinhalt

Die nachfolgende Animation 3.1-6 und Abb. 3.1-7 zeigen an Hand der Testbildvorlage *Lena*, wie verschiedene Bildblöcke zu einer unterschiedlichen Verteilung der Koeffizienten bei der DCT führen.



Animation 3.1-6: Beispiel für die DCT – fein strukturierter Bildinhalt

Beispiel DCT-Block: Homogener Bildbereich

Animation 3.1-6 repräsentiert einen 8 x 8 Pixel großen Bildblock mit fein strukturiertem Bildinhalt (links). Im korrespondierenden DCT-Block (rechts) sind viele Koeffizienten mit großem Betrag[16] vorhanden. In dem hier betrachteten Bildblock ist offensichtlich, dass die Korrelation zwischen benachbarten Bildpunkten relativ gering ist.

Ganz anders ist die Situation in Abb. 3.1-7. Hier wurde ein Bildblock mit einem recht homogenen Bildinhalt ausgewählt (links). Benachbarte Pixel korrelieren gut miteinander. Das führt erwartungsgemäß zu einer Konzentration von großen Koeffizienten im Bereich des DC-Koeffizienten.

16 Das Bild visualisiert die *absoluten Beträge* der DCT-Koeffizienten, also ohne Vorzeichen.

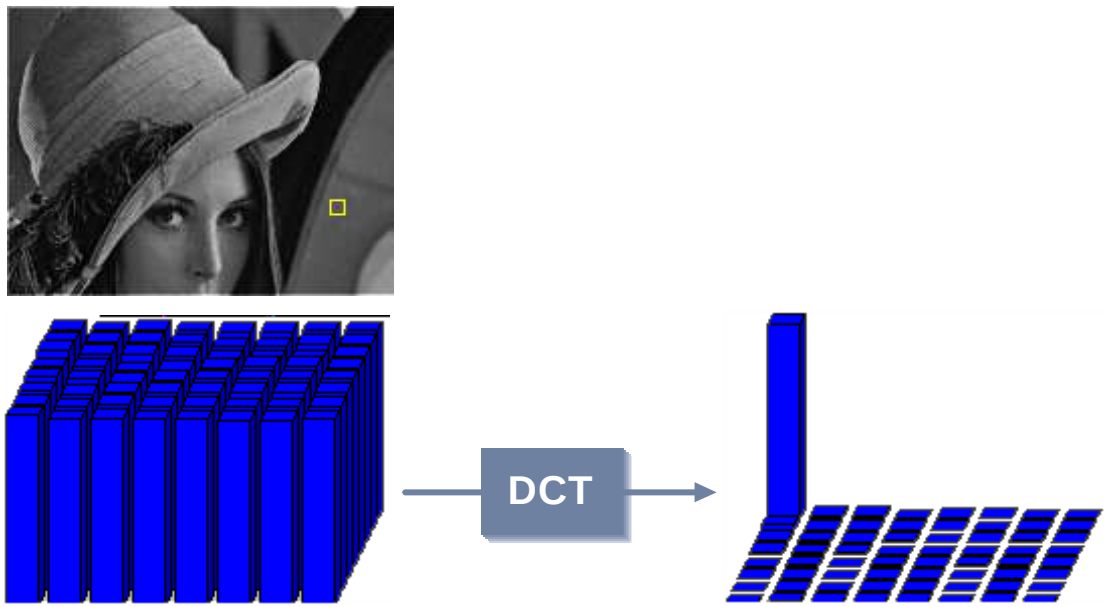


Abb. 3.1-7: Beispiel für die DCT – homogener Bildinhalt

Verschiebung des Zahlenbereiches

Bei der Berechnung der DCT subtrahiert man zur Einschränkung des Zahlenraumes für die DCT-Koeffizienten jeden Bildwert zunächst mit dem Mittelwert des Wertebereiches. Für eine 8-Bit Bilddarstellung ist dies der Wert 128. Aus Gl.3.1-6 ist überdies ersichtlich, dass die DCT-Koeffizienten keineswegs ganze bzw. nur positive Zahlenwerte annehmen müssen. Soll die Transformation exakt reversibel sein, was für den Fall einer reinen Redundanzreduktion erforderlich wäre, darf man die DCT-Koeffizienten nicht runden, sondern muss die rationalen Zahlenwerte beibehalten. Rundet man auf ganze Werte, ist dies bereits als Teil der Quantisierung der Koeffizienten zu sehen (vgl. Abschnitt 3.2).

Zahlenbeispiele DCT

Tabelle 3.1-1 und Tabelle 3.1-2 zeigen die so gewonnenen Ergebnisse für die oben aufgeführten Beispiele (Animation 3.1-6 und Abb. 3.1-7) mit einer Rundung auf ganze Werte für die DCT-Koeffizienten.

Tab. 3.1-1: *Beispielberechnung eines DCT-Blocks – fein strukturierter Bildbereich*

Bildvorlage 8 x 8 Pixel								DCT-transformierte Werte							
92	79	53	126	144	46	64	155	–101	–124	–3	30	62	38	–29	–34
140	61	66	142	124	83	124	113	–80	40	–41	3	72	16	–25	15
124	61	61	152	124	126	155	83	7	16	42	–10	75	–20	–5	22
90	106	106	142	127	142	146	95	–57	–14	16	–65	8	–65	6	–8
75	88	85	125	109	144	141	151	13	–3	–3	–18	18	–25	–1	–7
73	99	90	106	90	135	124	148	0	9	–21	–12	–25	–29	–11	–3
96	126	101	114	117	148	153	161	–16	–7	–3	11	9	3	7	–9
160	99	93	132	136	150	170	175	4	–24	1	10	6	0	–1	–9

Tab. 3.1-2: *Beispielberechnung eines DCT-Blocks – homogener Bildbereich*

Bildvorlage 8 x 8 Pixel								DCT-transformierte Werte							
160	159	160	159	158	158	159	159	238	4	4	1	2	2	1	1
161	160	160	159	156	156	153	153	8	2	2	–1	–1	0	–1	–1
163	160	160	153	155	155	156	153	–4	3	–1	–3	0	1	2	1
159	159	155	156	158	160	160	158	4	–3	–2	0	0	0	0	0
160	159	156	158	159	159	159	159	0	–3	0	3	–1	–2	–2	–1
159	156	158	158	158	158	158	158	–1	–1	0	1	–1	0	1	1
158	155	158	156	156	155	155	156	0	1	0	–1	1	0	1	0
158	155	156	155	155	154	154	155	0	2	1	–1	1	0	1	0

3.2 Quantisierung der DCT-Koeffizienten

3.2.1 Übersicht

Die im letzten Kapitel beschriebene *Diskrete Cosinus Transformation* führt alleine noch nicht zu einer Datenreduktion. Obschon sich in einem DCT-Block eine gewisse Energiekonzentration um den DC-Koeffizienten ergibt, bleiben die Werte für die übrigen Koeffizienten oft erhalten und lassen sich noch nicht direkt zusammenfassen. Um das Datenaufkommen eines DCT-Blockes weiter reduzieren zu können, werden die Koeffizienten im nachfolgenden Schritt unterschiedlich quantisiert. Bei der Auswahl einer vertretbaren Quantisierung macht man sich die visuellen Eigenschaften des menschlichen Auges zu Nutze (vgl. Abb. 1.2-7): Für Quantisierungsfehler in hochfrequenten Signalanteilen ist das Auge offensichtlich weniger empfindlich als für niederfrequente.

Errechnung der Quantisierung

Formal errechnet sich die Quantisierung als Rundung (nächstliegender ganzzahliger Wert) des Quotienten aus der DCT $F(u, v)$ und *Quantisierungsmatrix* $Q(u, v)$:

$$F_{DCT,Q}(u,v) = \frac{F(u,v)}{Q(u,v)}. \qquad 3.2\text{-}1$$

Bei der Auslegung der Quantisierungsmatrix sind verschiedene Gesichtspunkte zu berücksichtigen, die nachfolgend erörtert werden.

3.2.2 Visuelle Anpassung der Quantisierung

Gewichtet und quantisiert werden die DCT-Koeffizienten nach visuellen Gesichtspunkten. So ist die Berücksichtigung von Bildstrukturen möglich, da die einzelnen DCT-Koeffizienten unterschiedliche Strukturmerkmale repräsentieren. Da die Luminanz- und Chrominanzsignale getrennt transformiert und quantisiert werden, können überdies auch die unterschiedlichen Eigenschaften des visuellen Systems in diesem Punkt berücksichtigt werden (vgl. Abb. 1.2-6). In [3.8] wurden zahlreiche subjektive Untersuchungen vorgenommen, die zu einem Satz von Quantisierungstabellen für die Luminanz und die Chrominanz führten. Diese sind als Empfehlung in den *Anhang K* des JPEG-Standards (vgl. [3.6] u. [3.7]) eingeflossen. Abb. 3.2-1 zeigt diese beiden Tabellen.

Quantisierungsmatrizen bei JPEG

Quantisierungsmatrix - Luminanz

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	58	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

Quantisierungsmatrix - Chrominanz

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

Abb. 3.2-1: Gebräuchliche Quantisierungstabellen für Luminanz und Chrominanz

Erkennbar ist:

- Der DC-Koeffizient der Luminanz wird etwas gröber quantisiert als die benachbarten Koeffizienten. Mit dieser Maßnahme wird die leichte Bandpasscharakteristik des visuellen Systems in Bezug auf die örtliche Auflösung berücksichtigt.
- In vertikaler Richtung wird tendenziell etwas gröber quantisiert, als in horizontaler.
- DCT-Werte von diagonalen Bildstrukturen werden durch die Matrix stärker quantisiert also rein horizontale oder vertikale.
- Die Quantisierung der Chrominanz ist für mittlere und höhere Frequenzanteile gröber ausgelegt und gleichmäßiger über alle Koeffizienten verteilt. Die Matrix ist symmetrisch, das heißt horizontale und vertikale Strukturen werden gleich behandelt.

3.2.3 Einstellung des Kompressionsgrades

Quantisierung – Regler für Bildqualität

Die Qualität und damit der Kompressionsgrad ist über die Quantisierung einstellbar. Dazu könnte man beispielsweise verschiedene Quantisierungstabellen generieren und diese in den Datenstrom integrieren. Dieses Verfahren führt aber zu einem unnötigen Anstieg der Datenrate. Daher legt man nur ein Paar für die Quantisierungsmatrizen fest und führt zusätzlich einen gemeinsamen Skalierungsfaktor ein, der zwischen 1 und 100 variieren darf. Bei einem Wert von 50 nimmt die Quantisierungsmatrix die Werte gemäß Abb. 3.2-1 an. Ein Wert von 100 setzt alle Werte der Quantisierungsmatrix auf 1. Es findet also keine Quantisierung statt und dementsprechend auch keine Irrelevanzreduktion[17]. Eine maximale Quantisierung erhält man bei einem Skalierungsfaktor von 1. In diesem Fall werden alle Werte der Quantisierungsmatrix auf 255 eingestellt.

In der nachfolgenden Abb. 3.2-2 und Abb. 3.2-3 ist die Auswirkung der Quantisierung auf die beiden bekannten Bildblöcke des Testbildes *Lena* dargestellt.

17 Lediglich die in Abschnitt 3.1 beschriebene Rundung der DCT-Koeffizienten ist ggf. noch vorhanden.

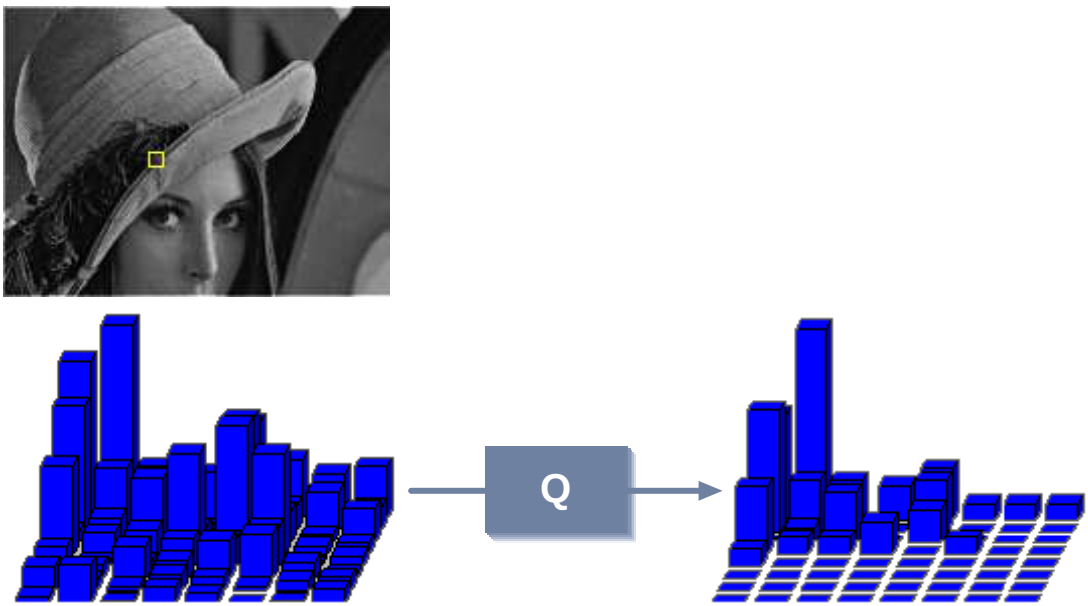


Abb. 3.2-2: Beispiel für die Quantisierung eines DCT-Blockes – fein strukturierter Bildbereich (Qualitätsstufe: 50)

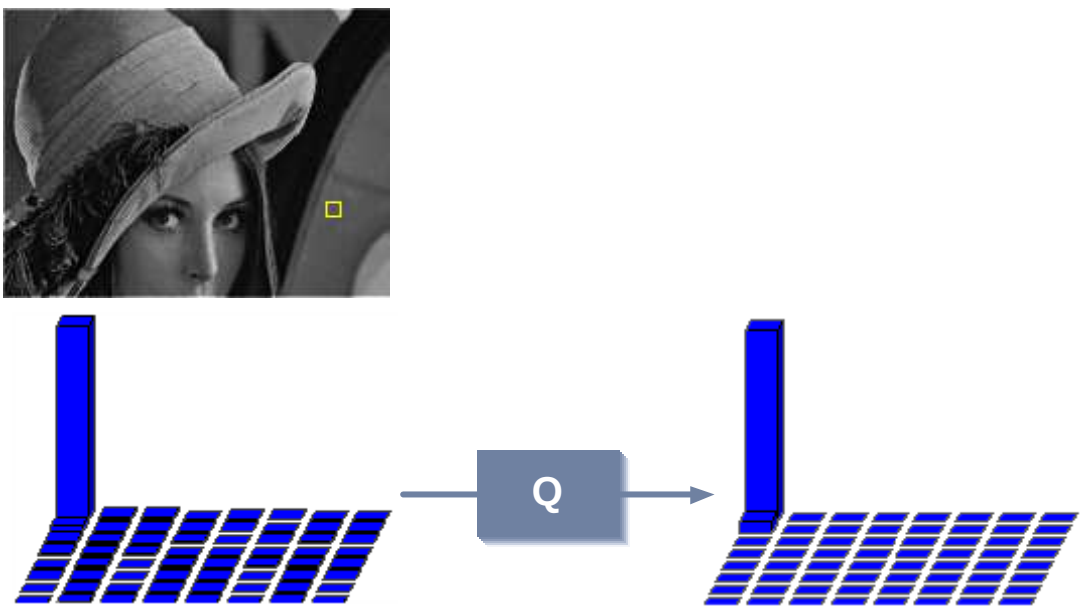


Abb. 3.2-3: Beispiel für die Quantisierung eines DCT-Blockes – homogener Bildbereich (Qualitätsstufe: 50)

In beiden Fällen kann eine große Anzahl von DCT-Koeffizienten zu Null quantisiert werden. Im zuletzt genannten Fall (Abb. 3.2-3) werden sogar alle Koeffizienten des DCT-Blockes zu Null, außer dem DC- und einem benachbarten AC-Koeffizienten.

3.2.4 Beispiel

Zum Abschluss des Kapitels soll an Hand eines eindimensionalen Rechenbeispiels die Wirkungsweise der Quantisierung verdeutlicht werden. Dazu wird die Formel für die 2D-DCT (Gl.3.1-6) umgeformt zu

Eindimensionale DCT

$$F_{1D}(u) = \frac{1}{2} C(u) \sum_{x=0}^{7} f(x) \cdot \cos\left[\frac{(2x+1)u \cdot \pi}{16}\right],$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases}. \qquad \text{3.2-2}$$

Für die Rücktransformation (*IDCT*) ergibt sich:

Eindimensionale IDCT

$$f_{1D}(x) = \frac{1}{2} \sum_{u=0}^{7} C(u) F(u) \cdot \cos\left[\frac{(2x+1)u \cdot \pi}{16}\right],$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases}. \qquad \text{3.2-3}$$

In Abb. 3.2-4 sind die Zahlenwerte für das Beispiel einer horizontalen, eindimensionalen Kante gezeigt.

8 Bildpunkte, Übergang von dunkel auf hell

70	70	70	100	150	200	200	200	digitale Werte der 8 Bildpunkte
-58	-58	-58	-28	22	72	72	72	nominiert auf Werte von -128 bis +127
13	-159	9	36	-7	-6	4	-6	ermittelt DCT-Koeffizienten (gerundet)
16	11	10	16	24	40	51	61	Quantisierungsmatrix
1	-14	1	2	0	0	0	0	nach Anwendung der Quantisierungsmatrix (gerundet)
76	68	73	105	153	190	203	200	reproduzierte digitale Werte nach IDCT

Abb. 3.2-4: Eindimensionales Rechenbeispiel zur Verdeutlichung des Quantisierungsfehlers

Es zeigt sich, dass nach Durchlaufen der ganzen Prozedur nicht identische, aber ähnliche Werte des Originalbildes erreicht werden. Nach Anwendung der Quantisierungsmatrix verbleiben nur noch zwei nicht verschwindende Koeffizienten. Die Wertefolge (1, –14, 1, 2, 0, 0, 0, 0) kann mit Hilfe einer Redundanzreduktion effizient gepackt und übertragen werden. Geeignete Verfahren zur Redundanzreduktion werden im nachfolgenden Abschnitt 3.3 vorgestellt und erläutert.

Selbsttestaufgabe 3.2-1:

Welche Aufgabe hat die Transformation bei der digitalen Bildcodierung?

Selbsttestaufgabe 3.2-2:

Was geben die DCT-Basisbilder (vgl. Abb. 3.1-5) an?

a. Oben links,

b. in der ersten Zeile ganz rechts,

c. in der ersten Spalte ganz unten?

Selbsttestaufgabe 3.2-3:

Gegeben sei der 2 x 2 Pixelblock einer Bildvorlage mit den Werten

$$f(x,y) = \begin{pmatrix} 10 & 15 \\ 12 & 9 \end{pmatrix}.$$

Geben Sie die DCT F(u, v) an!

***Hinweis**: Tragen Sie zunächst die erforderlichen Werte in eine Tabelle ein!*

Selbsttestaufgabe 3.2-4:

Wozu dient die Quantisierung der DCT-Koeffizienten?

3.3 Entropiecodierung der quantisierten Koeffizienten

3.3.1 Zick-Zack-Scan

Die quantisierten DCT-Koeffizienten sind dann gut redundant komprimierbar, wenn gleiche Werte in einer Reihefolge hintereinander liegend zusammengefasst werden können. Eine solche effiziente Codierung der Position der nicht verschwindenden Koeffizienten ist oft durch einen so genannten *Zick-Zack-Scan* der Transformationskoeffizienten möglich, da hier die festgestellte Konzentration der Energie in den verbleibenden Koeffizienten um den DC-Koeffizienten herum gut berücksichtigt ist. Der Zick-Zack-Scan ist in Abb. 3.3-1 für die beiden Beispielblöcke aus Abb. 3.2-2 und Abb. 3.2-3 dargestellt.

Zick-Zack-Scan

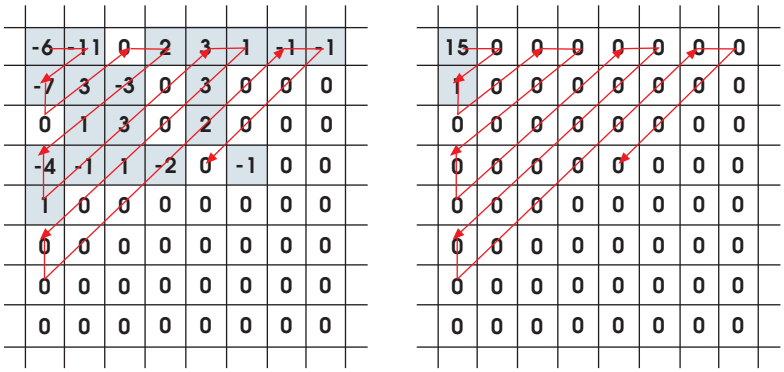


Abb. 3.3-1: Zick-Zack-Scan für Beispielblöcke: Fein strukturierter Bildbereich (links), homogener Bildbereich (rechts)

Je nach Verteilung der verbleibenden nicht verschwindenden Koeffizienten kann alternativ zum Zick-Zack-Scan auch ein normaler Zeilen- oder Spaltenscan die günstigere Koeffizientenfolge für die nachfolgende *Lauflängencodierung* (*Run Length Coding - RLC*) ergeben. In der Praxis probiert der Algorithmus von Block zu Block unterschiedliche, vorher vereinbarte Zusammenfassungsreihenfolgen aus und verwendet abschließend diejenige, die das kleinste Datenaufkommen verursacht.

3.3.2 Lauflängencodierung (*Run Length Coding – RLC*)

Run Length Coding RLC

Die Zusammenfassung der mit einem Zick-Zack- oder anderen Verfahren gewonnenen Koeffizientenfolgen geschieht in der Lauflängencodierung wie folgt:

- Der DC-Koeffizient wird direkt codiert.
- In einem Zahlenpaar werden die nachfolgenden AC-Koeffizienten abgelegt. Die erste Ziffer gibt dabei die Länge der ununterbrochenen Folge von Nullen vor dem Wert an, der selbst der zweiten Ziffer entspricht.
- Folgen im Block einem nicht verschwindenden Koeffizienten nur noch Nullen wird das Symbolwort „*End of Block*“ (*EOB*) angefügt. Der DCT-Block ist damit vollständig beschrieben.

Für die in Abb. 3.3-1 dargestellten Beispiele ergeben sich damit im Anschluss an die Lauflängencodierung folgende Wertepaare:

$$\begin{aligned} RLC_{\text{fein strukturiert}} = {} & (-6)(0,-11)(0,-7)(1,3)(1,2)(0,-3)(0,1) \\ & (0,-4)(0,1)(0,-1)(0,3)(1,3)(0,1)(0,3) \\ & (1,1)(5,-2)(0,2)(1,-1)(0,-1)(11,-1) \\ & EOB \end{aligned} \tag{3.3-1}$$

$$RLC_{\text{homogen}} = (15)(1,1)EOB \tag{3.3-1}$$

3.3.3 Variable Lauflängencodierung (*Variable Length Coding – VLC*)

Variable Lenght Coding VLC

Für die weitere Zusammenfassung der AC-Wertepaare[18] zu Codeworten wird eine so genannte *variable Lauflängencodierung* (*VLC*) herangezogen. Dazu werden mit Hilfe von fest definierten *Huffman-Tabellen* häufig vorkommende Wertepaare mit kürzeren und selten vorkommende Kombinationen mit längeren Codeworten beschrieben. Entsprechend der Bildungsvorschrift für das Wertepaar wird die erste Ziffer als *Run* bezeichnet (*Lauflänge* des zusammengefassten Abschnitts), die zweite Zahl wird in einer Tabelle nachgeschlagen und innerhalb einer so genannten Kategorie (*Category*) einem Codewort zugeordnet. Die Codeworte haben gemäß den obigen Überlegungen unterschiedliche Längen: kleine Zahlen werden mit kurzen, große Zahlen mit längeren Codeworten beschrieben. Damit wird der Tatsache Rechnung getragen, dass der genutzte Zahlenraum für die quantisierten Koeffizienten oft nur klein ist. Tabelle 3.3-1 gibt die für die JPEG-Codierung vordefinierte Tabelle für AC- und DC-Koeffizienten an. In ihr ist die Zuordnung zu den Kategorien und die verwendeten Codeworte für die zweite Zahl im Wertepaar festgelegt.

Kategorien und Wertebereiche

18 Der DC-Koeffizient selbst wird gesondert behandelt.

Tab. 3.3-1: *Festlegung der Kategorien, Wertebereiche und Codeworte für die AC- und DC-Koeffizienten*

Kategorie	Wertebereiche	Codeworte
1	–1, 1	0, 1
2	–3, –2, 2, 3	00, 01, 10, 11
3	–7, ..., –4, 4, ..., 7	000, ..., 011, 100, ..., 111
4	–15, ..., –8, 8, ..., 15	0000, ..., 0111, 1000, ..., 1111
5	–31, ..., –16, 16, ..., 31	etc.
6	–63, ..., –32, 32, ..., 63	etc.
7	–127, ..., –64, 64, ..., 127	etc.
8	–255, ..., –128, 128, ..., 255	etc.
9	–511, ..., –256, 256, ..., 511	etc.
10	–1024, ..., –512, 512, ..., 1023	etc.

In einer weiteren Tabelle sind die Codeworte für die verschiedenen Kombinationen aus *Lauflänge* und *Kategorie* zusammengefasst. Tabelle 3.3-2 zeigt Auszüge aus dieser Huffman-Tabelle für die AC-Koeffizienten.

Codeworte für AC-Koeffizienten

Tab. 3.3-2: *Festlegung der Codeworte aus der Lauflänge und der Kategorie für die AC-Koeffizienten: (0/X bis 5/X)*

Lauflänge/ Kategorie	Codewortlänge	Codeworte
EOB	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
...	...	...
1/1	4	1100
1/2	5	11011
...	...	...
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
...	...	...
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
...	...	...
4/1	6	111011
4/2	10	1111111000

Fortsetzung auf der naechsten Seite

Fortsetzung von der vorhergehenden Seite

Lauflänge/ Kategorie	Codewortlänge	Codeworte
4/3	16	1111111110010110
...	...	...
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
...	...	...

Tab. 3.3-3: *Festlegung der Codeworte aus der Lauflänge und der Kategorie für die AC-Koeffizienten: (8/X bis 15/X)*

Lauflänge/ Kategorie	Codewortlänge	Codeworte
8/1	9	111111000
8/2	15	111111111000000
8/3	16	1111111110110110
...	...	...
9/1	9	111111001
9/2	16	1111111110111110
...	...	...
10/1	9	111111010
10/2	16	1111111111000111
...	...	...
11/1	10	1111111001
11/2	16	1111111111010000
...	...	...
12/1	10	1111111010
12/2	16	1111111111011001
...	...	...
13/1	11	11111111000
13/2	16	1111111111100010
...	...	...
14/1	16	1111111111101011
...	...	...
15/0	11	11111111001
15/1	16	1111111111110101
...	...	...

Tab. 3.3-4: *Festlegung der Codeworte aus der Kategorie für den DC-Koeffizienten*

Lauflänge/ Kategorie	Codewortlänge	Codeworte
0	2	00
1	3	010
2	3	011
3	3	101
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110

Codeworte für DC-Koeffizienten

Der DC-Koeffizient wird zunächst genauso wie die AC-Koeffizienten gemäß Tabelle 3.3-1 einer Kategorie zugeordnet. Dem hierin enthaltenden Codewort wird ein zweites entsprechend der Kategorie in seiner Länge variables Codewort vorangestellt. Tabelle 3.3-3 zeigt hierfür die Zuordnung.

Der Aufstellung in Tabelle 3.3-1, Tabelle 3.3-2 und Tabelle 3.3-3 liegt ein recht komplexes Bildungsgesetz zugrunde, welches auf Ideen von *David Huffman* (1952) zurückgeht. Man spricht daher auch von einer *Huffman-Codierung*.

3.3.4 Beispiele

Beispiel VLC

Im Folgenden sollen einige Zahlenbeispiele zur richtigen Verwendung der Tabellen anleiten. Dazu wird das einfache Beispiel der RLC-Beschreibung aus Gl.3.3-1 herangezogen.

DC-Koeffizient: (15)

- Kategorie (vgl. Tabelle 3.3-1): **4**, Codewort: **1111**
- Codewort für Kategorie **4** (vgl. Tabelle 3.3-3): **101**
- Ergebnis: **101 1111**

AC-Wertepaar: (1,1)

- Kategorie (vgl. Tabelle 3.3-1): **1**, Codewort **1**
- Codewort für Lauflänge / Kategorie (**1,1**) (vgl. Tabelle 3.3-2): **1100**
- Ergebnis: **1100 1**

EOB: Codewort **1010**

Es ergibt sich damit für diesen Block ein nur 13 Bit langes Codewort:

$$VLC_{homogen} = 1011111110011010 \qquad 3.3\text{-}2$$

Vergleicht man diesen Wert mit den ursprünglich für 64 Bildpunkte notwendigen Datenaufkommen von 512 Bit so ergibt sich für diesen Block eine Kompressionsrate von etwa 39:1.

Tabelle 3.3-4 zeigt, dass die Verhältnisse für andere DCT-Blöcke im Bild sehr viel komplexer aussehen können. Der Ergebniscode für den Beispielblock mit der RLC nach Gl.3.3-1ist 117 Bits lang. Damit ist nur noch eine Kompression von etwa 4.3:1 erzielbar.

Die zuvor gemachten Überlegungen führen zu zwei wesentlichen Schlussfolgerungen:

- Die Kombination aus RLC und VLC erzeugt bildbezogen ein verschieden großes Datenaufkommen für die Zusammenfassung der einzelnen DCT-Blöcke.

Beispiel VLC-Tabelle

- Enthalten DCT-Blöcke im wesentlichen homogene Bildbereiche, dann lassen sich diese sehr viel effizienter komprimieren, als welche mit unterschiedlich fein strukturierten.

Tab. 3.3-5: *Beispiel für die Huffman-Codierung eines DCT-Blockes (fein strukturiert)*

RLC	Kategorie / Code	Huffman-Code	Ergebnis
DC: –6	3 / 001	101	101 001
(0,–11)	4 / 0100	1011	1011 0100
(0,–7)	3 / 000	100	100 000
(1,3)	2 / 11	11011	11011 11
(1,2)	2 / 10	11011	11011 10
(0,–3)	2 / 00	01	01 00
(0,1)	1 / 1	00	00 1
(0,–4)	3 / 011	100	100 011
(0,1)	1 / 1	00	00 1
(0,–1)	1 / 0	00	00 0
(0,3)	2 / 11	01	01 11
(1,3)	2 / 11	11011	11011 11
(0,1)	1 / 1	00	00 1
(0,3)	2 / 11	01	01 11
(1,1)	1 / 1	1100	1100 1
(5,–2)	2 / 01	111111110110	111111110110 01
(0,2)	2 / 10	01	01 10
(1,–1)	1 / 0	1100	1100 0
(0,–1)	1 / 0	00	00 0
(11,–1)	1 / 0	1111111001	1111111001 0
EOB			1010

Eine Verbesserung der Kompression von „schwierigen“ DCT-Blöcken kann erreicht werden, wenn statt der vordefinierten Huffman-Tabelle eine eigene Tabelle generiert wird, welche der Statistik im betroffenen Block besser angepasst wird. Dabei ist jedoch zu beachten, dass auch diese Tabelle selbst in den Code mit eingearbeitet werden muss.

DPCM des DC-Koeffizienten

Ergänzend sei noch auf eine Besonderheit bei der Codierung der DC-Koeffizienten hingewiesen. Wie bereits erwähnt kommt es häufig zu einer Energiekonzentration auf den DC-Koeffizienten. Dies führt zu einem relativ großen Zahlenraum für diesen Koeffizienten und damit verbunden zu einer ineffizienten Huffman-Codierung. Daher nutzt man zusätzlich die Korrelation von DC-Koeffizienten benachbarter DCT-Blöcke aus und codiert statt dessen die Differenz von aufeinander folgender DC-Koeffizienten. Das Ergebnis sind Differenz codierte DC-Koeffizienten, welche oftmals einen wesentlich kleineren Zahlenraum abdecken und damit besser Huffman codiert werden können.

Die hier beschriebene Nutzung der Korrelation über die Bildung von Differenzen ist in der Literatur unter dem Begriff der *Differenz Pulse Code Modulation* (*DPCM*) bekannt und ist für die Codierung von Bewegtbildern von besonderer Bedeutung. In Kapitel 4 wird hierauf noch gesondert eingegangen.

Selbsttestaufgabe 3.3-1:

Wozu dient bei der JPEG-Codierung der Zick-Zack-Scan?

Selbsttestaufgabe 3.3-2:

Beschreiben Sie kurz die Aufgabe des Run Length Coding!

Selbsttestaufgabe 3.3-3:

Welche Aufgabe hat die Variable Lauflängencodierung?

3.4 Wavelet-Codierung und JPEG 2000

3.4.1 Eigenschaften der DCT-Basisfunktionen

Der im letzten Kapitel beschriebenen Einzelbildcodierung nach dem JPEG-Verfahren haften eine Reihe von Nachteilen an. Aufgrund der relativ großen Anzahl von DCT-Blöcken[19], von denen jeder separat codiert werden muss, besitzt das Datenaufkommen eine feste untere Schranke. Diese ist dann erreicht, wenn die Quantisierung maximal ist und damit nur noch die Werte –1, 0 und +1 für einzelne Koeffizienten möglich sind. Als Ergebnis daraus verbleiben für jeden DCT-Block nur noch wenige dominante Koeffizienten ungleich Null (idealer Weise oft nur noch der DC-Koeffizient). Abb. 3.4-1 zeigt diesen Fall für die Testbildvorlage *Lena*.

Abb. 3.4-1: JPEG-Codierung bei maximaler Kompression

Grenzen von JPEG

Ohne Optimierungen in der Lauflängen- und Huffman-Codierung verbleibt für dieses Einzelbild ein Datenaufkommen von etwa 8 kByte[20]. Schaut man sich die DCT-Blöcke und das schlechte Codierergebnis an, wird deutlich, dass eine hohe Kompression durch die starke Quantisierung der DCT-Koeffizienten bei der klassischen JPEG-Codierung nicht ratsam ist. Überdies zeigt die Betrachtung der DCT-Blöcke, dass die Dekorrelation bestimmter Bildstrukturen mit Hilfe der DCT nur suboptimal funktioniert: Die Bildenergie verteilt sich weiterhin auf relativ viele DCT-Koeffizienten. Dies hängt damit zusammen, dass die DCT zwar das ursprüngliche Signal durch die gewichtete Überlagerung von Cosinus-Basisfunktionen unterschiedlicher Frequenzen annähern kann, aber keine Aussage über den Ort macht, an dem eine bestimmte Frequenzkomponente im Bild auftritt. Außerdem können Kanten („*Peaks*“), die relativ häufig in natürlichen Bildvorlagen vorhanden sind, nur

19 Bei einem Einzelbild eines ITU-R BT.601-konformen Videobildes müssen gemäß Gl. 3.1-5 insgesamt 6480 Blöcke codiert werden.

20 Mit der so genannten *progressiver Codierung* (hier nicht behandelt) und voller Optimierung bei der Huffman-Codierung ist eine Verbesserung der Kompression um den Faktor 8:1 noch möglich.

sehr schlecht mit der Superposition von wenigen DCT-Basisfunktionen angenähert werden.

Blockartefakte bei JPEG

Ersichtlich wird auch die örtliche Korrelation benachbarter DCT-Blöcke im JPEG-Verfahren nicht ausreichend berücksichtigt. Dieses als „*Blocking*“ bekannte Artefakt ist subjektiv besonders störend. Die Ursache ist in der blockweise vorgenommenen Bildfeldzerlegung zu sehen. Diese Art der Bildfeldzerlegung wiederum war jedoch erforderlich, um den örtlich nicht beschränkten Basisfunktionen (Cosinus-Funktionen) ein örtliche Bildzuordnung geben zu können. Die DCT-Koeffizienten selbst geben nämlich nur innerhalb des betrachteten Blocks Aufschluss über die zugehörige Frequenz. Wäre er klein, wäre die örtliche Zuordnung gut möglich, dann können aber nur wenige Basisfunktionen für die Beschreibung der Frequenzkomponenten gebildet werden und örtliche Korrelationen werden noch weniger gut ausgenutzt. Überdies reduziert sich der Kompressionsgewinn.

3.4.2 Blockschaltbild eines verbesserten Transformations-Encoders

Die Vorüberlegungen des letzten Abschnitts legen nahe, dass die Effizienz der Codierung mittels Transformationscodierung gesteigert werden kann, wenn statt der Beschreibung des Bildes mittels DCT-Basisfunktionen eine modifizierte Transformation gewählt wird. Eine solche Transformation stellt die so genannte *Wavelet Transformation* dar, die Thema des folgenden Abschnitt 3.4.3 ist.

Funktionsbild Wavelet-Encoder

Abb. 3.4-2 zeigt das Funktionsbild eines *Wavelet-Encoders*.

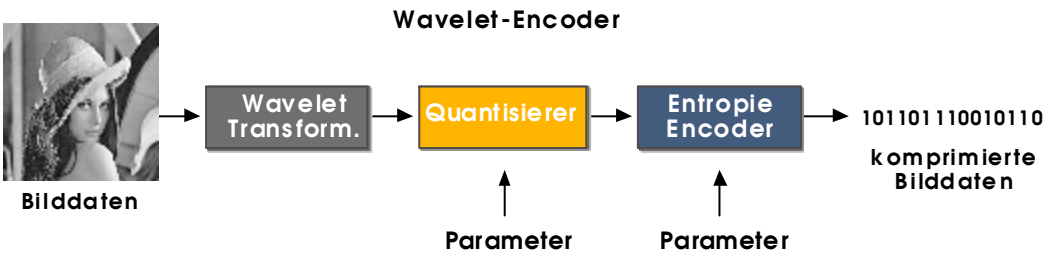


Abb. 3.4-2: Funktionsbild eines Wavelet-Encoders

3.4.3 Wavelets als Basisfunktionen

Problematik Frequenzanalyse und Auflösungsanalyse

Die Wahl der Blockgröße beim JPEG-Verfahren stellt einen Kompromiss bei dem Versuch dar, *örtliche Auflösung* und *Frequenz-Auflösung* eines Bildes gleichzeitig zu beschreiben. Mit der Lösung dieser Grundproblematik haben sich diverse Mathematiker lange Zeit beschäftigt. Hierbei kamen sie von der *Frequenzanalyse* zur *Auflösungsanalyse*. Dies bedeutet, dass man die Eigenschaften eines örtlichen Signals mit Hilfe von mathematischen Strukturen, die sich nur in ihrer Auflösung unterscheiden, untersucht. Das kann man beispielsweise erreichen, indem man eine *örtlich beschränkte Basisfunktion* entwirft, diese *verschiebt* und *skaliert*. Mit dieser verschobenen und skalierten Basisfunktion versucht man nun, das zu untersuchende Signal darzustellen. Danach wird dieselbe Basisfunktion um andere Werte verschoben und skaliert, und wiederum versucht man, das Signal auf der Basis dieser neuen Funktion darzustellen. Dieses Vorgehen kann man dann theoretisch unendlich oft wiederholen. Als Ergebnis erhält man Darstellungen des Signals bei unterschiedlichen Auflösungen. So ist es also möglich, den Ort einer auftretenden Frequenzkomponente zu ermitteln. Basisfunktionen mit den zuvor beschrieben Eigenschaften werden *Wavelet-Basisfunktionen* oder kurz *Wavelets*[21] genannt. Im Folgenden soll veranschaulicht werden, warum gerade Wavelets für die Bildkompression besonders geeignet sind. Dazu wird aus Gründen der Übersichtlichkeit nur eine eindimensionale Betrachtungsweise gewählt. Vertiefende Details zur Wavelet-Transformation lassen sich beispielsweise in [3.9] und [3.10] finden.

Wie bei der DCT zu sehen war, ist es gebräuchlich, ein Signal *f(x)* mit der Hilfe der gewichteten Summe von einfachen Grundbausteinen (*Basisfunktionen*) darzustellen:

$$f(x) = \sum_i c_i \Psi_i(x) \qquad \text{3.4-1}$$

mit $\Psi_i(x) =$ Basisfunktionen und $c_i =$ Koeffizienten (oder Gewichte).

Impuls-Funktion

Da die Basisfunktionen fest definiert sind, steckt die Information über das Signal allein in den Koeffizienten. Am einfachsten gelingt die Beschreibung eines Signals, wenn man lediglich die *Impulsfunktion* benutzt. Auf diese Weise erhält man eine Darstellung, die nur Informationen über das Verhalten des Signals als *örtlich-zeitlichen Verlauf* enthält.

Sinus-Funktion

Verwendet man *Sinussignale* als Basisfunktionen, erhält man die bekannte *Fourier-Darstellung*, die auch schon für die DCT benutzt wurde. Diese Darstellungsweise zeigt das *Frequenzverhalten* des Signals. Keine der beiden genannten Darstellungsmöglichkeiten ist für die Komprimierung von Bildinformationen ideal, da man eine

21 Die mathematische Theorie von Basisfunktionen bzw. Basisfunktionen bei unterschiedlichen Auflösungen wurde in der ersten Hälfte des 20. Jahrhunderts entwickelt. Erst in den 80er Jahren gelang es auf der Basis der digitalen Signalverarbeitung Wavelet-Basisfunktionen zu finden, die sinnvoll für die Bildcodierung eingesetzt werden konnten.

Repräsentation benötigt, die Informationen sowohl über das *örtlich-zeitliche Verhalten* als auch über das *Frequenzverhalten* des Signals in sich trägt. Gebraucht wird die Information über das Frequenzverhalten des Signals an einem bestimmten Ort, um nach der *Rücktransformation* wieder eine möglichst exakte Reproduktion des Ausgangssignals *f(x)* zu erhalten.

Frequenzaufspaltung

Es ist von besonderem Vorteil, dass in natürlichem, fotografischen Bildmaterial Bereiche mit einem Anteil hoher Frequenzen (*„Kanten“*) räumlich sehr konzentriert auftreten, während niedrige Frequenzen meist größere Flächen betreffen. Diese Beobachtung führt zu der Idee, die hochfrequenten Teile des Bildes *„herauszufiltern“* und sie getrennt von den restlichen Informationen in einer geeigneten Weise darzustellen. Mit entsprechenden Bandpassfiltern ließe sich so die Bandbreite des Signals stufenweise halbieren.

Dieses Beispiel zeigt, dass gerade die Wavelet-Basisfunktionen mit den oben beschriebenen Eigenschaften einen guten Kompromiss darstellen, da sie sich aus *Impuls-* und *Sinus-Funktionen* zusammensetzen. Günstig ist dabei die Wahl einer Gruppe von Basisfunktionen, von der jede eine unterschiedliche örtliche Ausdehnung hat. Über die unterschiedliche zeitliche Ausdehnung kann dann der Anteil der örtlichen Informationen oder der Anteil der Frequenzinformationen eingestellt werden. Eine breite Basisfunktion kann z. B. einen großen Anteil des Signals untersuchen und somit mehr Frequenzinformationen als örtliche Informationen liefern, während eine schmale Basisfunktion besser die örtlichen Eigenschaften eines schmalen Bereichs darstellen kann.

Scaling-Funktion Mutter-Wavelet

Um die Sache zu vereinfachen, soll im Folgenden angenommen werden, dass alle Basisfunktionen $\{\Psi_i\}$ skalierte und verschobene Versionen ein und derselben Prototypfunktion Ψ, der *Scaling-Funktion* oder *Mutter-Wavelet* sind. Die passende Skalierung wird durch die Multiplikation mit einem Vielfachen von 2^ν erreicht. Weil Ψ per Definition endlich ist, muss die Funktion um einen Wert *k* verschoben werden, um das ganze Signal zu erfassen:

$$\Psi_{\nu k}(x) = 2^{\frac{\nu}{2}}\Psi(2^\nu x - k),\ k \in Z. \qquad \text{3.4-2}$$

Die Multiplikation mit $2^{\frac{\nu}{2}}$ ist notwendig, um die Basisfunktionen *orthonormal* zu machen. Man erhält damit die vollständige *Wavelet-Darstellung* des eindimensionalen Signals *f(x)*:

Eindimensionale Wavelet-Darstellung

$$f(x) = \sum_{\nu\ endl.} \sum_{k\ endl.} c_{vk}\Psi_{vk}(x). \qquad \text{3.4-3}$$

Die Berechnung der Koeffizienten geschieht innerhalb der Wavelet-Transformation: Sie ergeben sich aus dem inneren Produkt des Signals mit den Basisfunktionen.

Wavelets als Teilbandcodierer

Bei der Anwendung der Wavelet-Transformation stell man fest, dass diese auch als *oktaves Bandpassfilter* angesehen und implementiert werden kann. Die Kompression mit Hilfe der Wavelet-Transformation wird daher auch oft als *Teilbandcodierer* bezeichnet. Die Betrachtungsweise der Wavelet-Transformation über Filter

ist anschaulicher und wird daher nachfolgend näher erläutert. Dazu verwendet man *Filterpaare* aus *Hoch- und Tiefpassfiltern*, so genannte *Quadrature Mirror Filter* (*QMF*), die sich rekursiv wie folgt definieren lassen:

Quadrature Mirror Filter QMF

$$T(x) = \sum_{k=0}^{M-1} c_k T(2x-k) \text{ und } H(x) = \sum_{k=0}^{M-1} (-1)^k c_{1-k} T(2x-k) \qquad 3.4\text{-}4$$

mit *T*(*x*) als *Scaling-Funktion* oder *Mutter-Wavelet* und c_k als Filterkoeffizienten.

Die Anzahl der Filterkoeffizienten bestimmt allgemein die *Ordnung des Wavelets* und somit die Anzahl der Verschiebungen über die x-Achse. Die Ordnung legt damit auch fest, wie viele Pixel in der Umgebung des aktuell zu beschreibenden Punktes betrachtet werden.

Bevor die Wavelet-Transformation an Hand eines einfachen Beispiels verdeutlicht wird, sollen nachfolgend noch einmal die wesentlichen Aussagen zusammengefasst werden:

Wavelet-Transformation Zusammenfassung

1. Die (*Diskrete*) *Wavelet Transformation* (*DWT*) ist der (*Diskreten*) *Cosinus Transformation* (*DCT*) sehr ähnlich. Die DCT benutzt allerdings *örtlich unbegrenzte Cosinusfunktionen* zur Analyse des Bildmaterials, die *DWT* hingegen benutzt *örtlich endliche Basisfunktionen*.
2. Die Basisfunktionen sind die so genannten *Scaling-Funktionen* (*Mutter-Wavelets - Tiefpass-Funktion*) und die *Wavelets* (*Hochpass-Funktion*).
3. Diese Funktionen verbinden die grundlegende Eigenschaft der *Orthogonalität*, d. h. die Vektoren der Funktionen stehen senkrecht zueinander, die eine Transformation und eine identische Rekonstruktion erst ermöglichen.
4. Die Funktionen besitzen gemäß 1. eine *endliche Ausdehnung*. Dies erlaubt die *Analyse* von Bilddaten *ohne Fenstereffekte* (*Blocking-Artefakte*), die aus der Anwendung unendlicher Funktionen bei der DCT auf endliche Bildbereiche resultieren. Während bei der DCT für alle Frequenzen die gleiche Abtastraten gelten, sind die Abtastraten bei der Wavelet Transformation abhängig von der Frequenz (*Oktavbandzerlegung* oder *Subbandcodierung*). Bei Erhöhung der Frequenz um eine Oktave wird die Abtastrate verdoppelt.

Abb. 3.4-3 veranschaulicht den Signalfluss eines Iterationsschrittes der zweidimensionalen Wavelet Transformation mit Quadraturspiegelfiltern (*QMF*). In der Praxis werden mehrere Iterationen durchgeführt, indem man den tiefpassgefilterten Teil (TT) wieder in 4 Subbänder aufspaltet und skaliert.

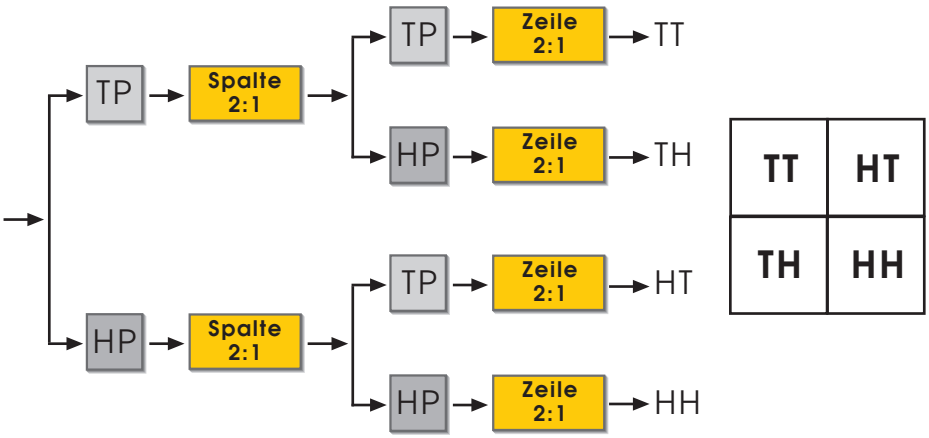


Abb. 3.4-3: Prinzip der 2D-Wavelet-Transformation (Eine Skalierungsstufe)

Skalierungsstufen

Abb. 3.4-4 zeigt die Anordnung der Frequenzbänder nach einer Skalierungsstufe (links, *4-Band-Zerlegung*) und nach 4 Skalierungsstufen (rechts, *13-Band Zerlegung*). Die Anzahl der Skalierungsstufen entscheidet über die Komplexität des Gesamtalgorithmus. So kann in der Praxis abhängig von den Anforderungen eine unterschiedliche Anzahl von Skalierungsstufen verwendet werden. Als guter Kompromiss aus Komplexität und Kompressionsergebnis können 3 Skalierungsstufen angesehen werden.

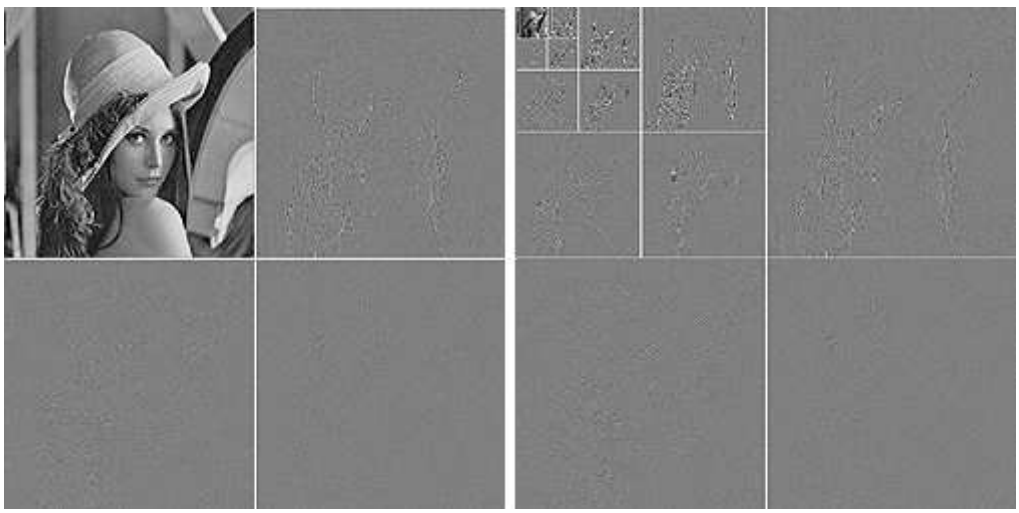


Abb. 3.4-4: Anordnung der Frequenzbänder bei einer *4-Band Zerlegung* (links, eine Skalierungsstufe) und *13-Band Zerlegung* (rechts, 4 Skalierungsstufen)

Lokale Verdichtung

Es ist zu erkennen, dass die Energie des Bildes im örtlich unterabgetasteten Tiefpass-Bereich und in den Kantenbereichen der entsprechenden Hochpass-Abschnitte konzentriert wird. Diese Konzentration der Energie des Bildes wird auch als *lokale Verdichtung* bezeichnet. Ersichtlich wird bei der Wavelet-Codierung keine Bildfeldzerlegung in Blöcken vorgenommen. Die störenden Blocking-Artefakte können demnach nicht auftreten.

Eine Rundung der Koeffizienten führt dazu, dass auch Wavelet-basierte Bildkompressionstechniken verlustbehaftet sind. Ist das Wavelet-transformierte Signal an den meisten Stellen gleich oder sehr nahe Null, so lässt sich ähnlich wie bei der DCT mit Hilfe einer geeigneten Quantisierung und einer anschließenden RLC komprimieren. Die erreichbaren Kompressionsraten bei akzeptabler Rekonstruktionsqualität sind verhältnismäßig hoch. Bei natürlichem, fotografischen Ausgangsmaterial sind Kompressionsraten von ca. 50 : 1 und darüber mit vertretbaren Qualitätseinbußen erreichbar.

3.4.4 Beispiel einer Wavelet Transformation – das Haar-Wavelet

Ein sehr einfaches Beispiel für eine Wavelet-Funktion ist das *Haar-Wavelet*, welches in Abb. 3.4-5 skizziert ist.

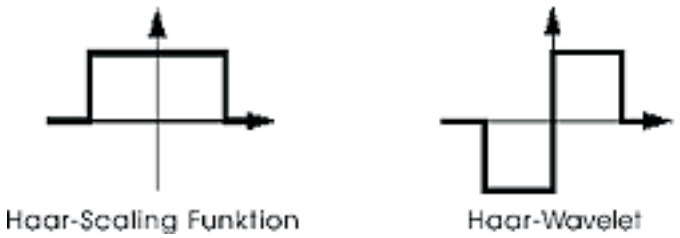


Abb. 3.4-5: *Haar-Scaling Funktion* und *alternatives Haar-Wavelet.*

Haar-Scaling Function
Haar-Wavelet

Zu Zwecken der Erläuterung ist dieses Beispiel gut geeignet, da hier das Interpolationsschema recht einfach zu verstehen ist. Es muss allerdings angemerkt werden, dass die Wavelet Transformation mit dem *Haar-Wavelet* für die Praxis der Bilddatenkompression keine Bedeutung hat.

Mittelwertsignal
Differenzsignal

Im Folgenden wird ein Haar-Wavelet der Ordnung Zwei zugrunde gelegt, d. h. bei der Faltung während der Berechnung der Koeffizienten werden jeweils nur zwei benachbarte diskretisierte Pixel des Eingangssignals betrachtet. Hier lässt sich anschaulich das Interpolationsschema erkennen. Das Signal wird zum einen in ein *Mittelwertsignal* und zum anderen in ein *Differenzsignal* gewandelt. Die zwei Ausgangssignale besitzen jeweils nur halb so viele Koeffizienten wie das Eingangssignal Werte enthielt. Abb. 3.4-6 verdeutlicht den Vorgang an einem einfachen Zahlenbeispiel.

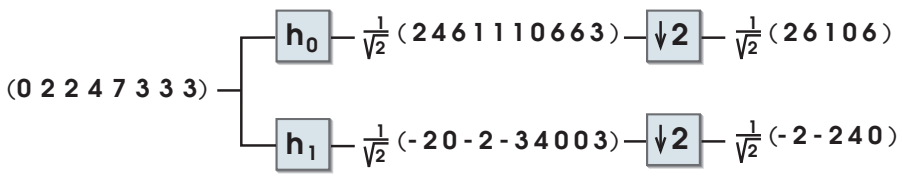


Abb. 3.4-6: Funktionsweise der *Analyse-Filterbank* des Haar-Wavelets 2. Ordnung am Beispiel einer Bildzeile: {0 2 2 4 7 3 3 3}

Das Haar-Wavelet 2. Ordnung entspricht einem QMF-Paar mit folgenden Koeffizienten:

$$h_0 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right),\ h_1 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right). \qquad 3.4\text{-}5$$

Wird keine weitere Quantisierung vorgenommen, ist eine perfekte Rekonstruktion möglich, wie Abb. 3.4-7 verdeutlicht. Hierzu werden die Frequenzanteile vorher mit einem *Upsampling-Operator* ($\uparrow 2$) wieder auf ihre volle Länge gestreckt und anschließend mit dem Synthese-Filterpaar aus $g_0 = h_0$ und $g_1 = h_1$ gefiltert[22].

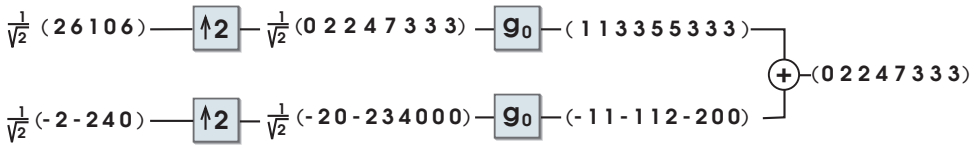


Abb. 3.4-7: Funktionsweise der *Synthese-Filterbank* des Haar-Wavelets 2. Ordnung am Beispiel aus Abb. 3.4-6

3.4.5 Vergleich verschiedener Wavelets

In der Bildverarbeitung werden normalerweise komplexere Wavelets verwendet, anhand derer eine effizientere Quantisierung und damit eine entsprechend höhere Kompression erreichbar ist. Werden andere Scaling-Funktionen und Wavelets höherer Ordnung benutzt, sind die Ergebnisse immer noch mit denen des einfachen Beispiels aus Abschnitt 3.4.4 vergleichbar. Die Ausgangssignale sind dann allerdings nicht mehr genau die *Durchschnitts-* und *Differenzsignale*, sondern können den Anforderungen der Bildkompression optimal angepasst werden. Ein wichtiger Schritt ist das Finden von Wavelet-Funktionen, die beim Hochpass gefilterten Signal viele sehr kleine Werte erzeugen (entscheidend für den anschließenden Quantisierungsschritt).

Daubechies-Wavelets

So führen Codieralgorithmen mit *Daubechies-Wavelets* oder *biorthogonale Wavelets* zu gegenüber JPEG überlegenen Resultaten. Die *Daubechies-Wavelets* wurden 1988 von *I. Daubechies* (vgl. [3.11]) als orthogonale Wavelets entwickelt, die auf dem endlichen Intervall [a; b] ungleich Null sind. Es handelt sich um ein Wavelet 8. Ordnung[23]. Sie zeichnen sich durch besonders gute Frequenzanalyse- und Kompressionseigenschaften aus. Vergleichende Tests zeigen, dass die einfache Haar-Transformation noch keine so gute lokale Verdichtung bringt wie andere Wavelets mit höherer Ordnung.

22 Es ist zu beachten, dass die *Synthese-Filterbank* nur in diesem Fall exakt gleich der *Analyse-Filterbank* ist.

23 Das entsprechende Filter hat 8 Taps.

Obwohl für die praktische Realisierung von Wavelet-Codierern inzwischen eine Reihe von guten Wavelet Transformationen bekannt sind, schreitet die Entwicklung von optimierten Wavelet-Basisfunktionen für die Bildcodierung immer noch weiter voran. Die Entwicklung von optimierten Wavelet-Basisfunktionen für natürliche Bildvorlagen führt teilweise zu recht komplexen Algorithmen, so dass oft ein vernünftiger Kompromiss zwischen Aufwand und Codiereffizienz gefunden werden muss.

3.4.6 Zusammenfassen und Codieren der Koeffizienten

Wie im Abschnitt 3.4.3 beschrieben wurde, hatte die Wavelet Transformation zunächst die Aufgabe, die Pixelwerte des Originalbildes zu dekorrelieren und so die Bildinformation in einer relativ kleinen Anzahl von Koeffizienten ungleich Null zu konzentrieren. Diese Tatsache bildet – wie auch schon beim JPEG-Algorithmus – eine gute Voraussetzung für eine Quantisierung dieser Werte.

Auch bei der Wavelet-Codierung kann man nach einer geeigneten Sortierung der Koeffizienten eine Schwellfunktion (*Threshold-Function*) wählen, oberhalb derer die Koeffizienten quantisiert übertragen werden:

$$S_q(x) = \begin{cases} 0 & wenn\,|x| < x_0, \\ x & sonst. \end{cases} \qquad \text{3.4-6}$$

Zero-Tree-Methode

Dabei muss allerdings beachtet werden, dass in den hochfrequenten Skalierungsstufen eine feinere Quantisierung erforderlich ist. So wählt man für jede Skalierungsstufe eine eigene Quantisierungstabelle, die im Idealfall an die statistische Verteilung der Koeffizienten über alle Skalierungsstufen hinweg angepasst ist. Weiterhin nutzt man die örtliche Korrelation zwischen den Koeffizienten aus verschiedenen Skalierungsstufen, indem man die so genannte *Zero-Tree-Methode* anwendet (vgl. Abb. 3.4-8) [3.12].

Abb. 3.1-2 legte bereits nahe, dass die Energie des Spektrums von natürlichen Bildvorlagen in der Regel zu höheren Frequenzen hin abnimmt. Es ist also nicht sehr wahrscheinlich, dass größere Energieanteile in hohen Frequenzen gefunden werden, wenn im gleichen örtlichen Bildabschnitt schon ein großer Energieanteil in den niedrigen Frequenzen steckt.

Da die Koeffizienten im weitesten Sinne Frequenzanteile repräsentieren, kann man sich diesen Zusammenhang auch beim Zusammenfassen der Wavelet-Koeffizienten zu Nutze machen. Man beginnt dazu mit dem Teil der Koeffizienten, die das Bild in der geringsten Auflösung darstellen (im Beispiel Abb. 3.4-8 links oben) und sucht dann der zunehmenden Auflösung folgend nach Koeffizienten, die gemäß der Schwellenfunktion Null sind. Ist das der Fall kann man mit sehr hoher Wahrscheinlichkeit davon ausgehen, dass die örtlich korrespondierenden Koeffizienten in den folgenden, höheren Auflösungen auch Null sind. Aufgrund der Skalierung korrespondieren zu einem Koeffizienten immer jeweils 4 Koeffizienten in der nächsten Skalierungsebene (siehe Pfeile in Abb. 3.4-8). Die *Wurzel* („*Root*“) eines Zero-Trees

muss nicht zwingend in dem Teil des Wavelet transformierten Bildes liegen, das die geringste Auflösung repräsentiert. Auch in höheren Auflösungen wird nach neu auftretenden Null-Koeffizienten gesucht, die dann wiederum eine neue Wurzel darstellen.

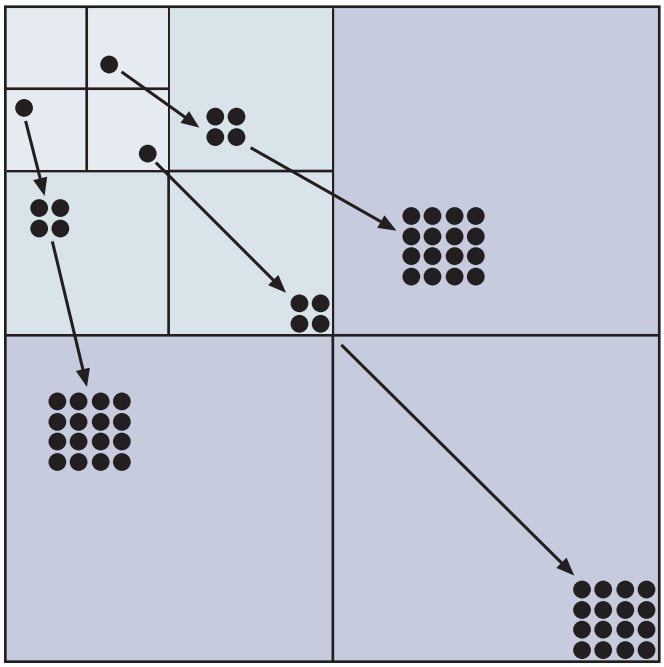


Abb. 3.4-8: Veranschaulichung der Zero-Tree-Struktur für Wavelets mit 3 Skalierungsstufen (nach [3.12], [3.13])

In der Praxis wählt man ein iteratives Verfahren, um so möglichst exakt ein bestimmtes Datenaufkommen einhalten zu können. Dazu sollten zuerst die Koeffizienten übertragen werden, welche die größte *Signifikanz des Bildes*, d. h. die größte Energie beinhalten. Dies kann erreicht werden, wenn man die Koeffizienten der Größe nach ordnet und dann in dieser Reihenfolge überträgt. In einem ersten Durchgang wählt man einen relativ großen Wert x_0 für die Schwellenfunktion $S_q(x)$. Damit wird der Wavelet-Koeffizient mit dem größten Betrag ermittelt. In den folgenden Durchgängen wird x_0 halbiert. Damit sind immer noch die meisten Koeffizienten Null. Ab diesem Schritt wird nach den Zero-Trees gemäß Abb. 3.4-8 gesucht. Die gefundenen Koeffizienten werden in vier Gruppen eingeordnet:

- *Zero-Tree Root – ZTR,*
- *Isolated Zero – IZ,*
- *Positive Value – POS,*
- *Negative Value – NEG.*

Alle gefundenen Null-Koeffizienten sind zunächst als *ZTR* klassifiziert (*Default*). Sollte sich dann bei einer *Verfeinerung* in einer weiteren Iteration mit einem kleineren Schwellwert x_0 ergeben, dass innerhalb der Baumstruktur für diese Wurzel Koeffizienten ungleich Null enthalten sind, dann wird dieser Null-Koeffizient als *IZ* klassifiziert. In den meisten Fällen tritt allerdings das ein, was eingangs für die Zero-Tree Struktur vorhergesagt wurde: Die weiteren Koeffizienten, die im Baumverlauf eines *ZTR*-Koeffizienten liegen, sind ebenfalls Null und müssen nicht mehr codiert werden. Alle von Null verschiedenen Koeffizienten werden entweder als *POS* für positive Werte oder *NEG* für negative Werte klassifiziert und werden in späteren Durchläufen nicht mehr neu durchsucht.

Die iterative Halbierung des Schwellwertes x_0 führt dazu, dass immer mehr Koeffizienten nicht mehr unter die Schwelle fallen und sich damit auch die Anzahl der Zero-Trees verringert: Das Datenaufkommen *D* für die Codierung der Koeffizienten steigt sukzessiv an. Die Iteration wird abgebrochen, wenn ein maximal gewünschtes Datenaufkommen *D*max erreicht ist.

Die verbleibenden Wavelet-Koeffizienten lassen sich aufgrund der Baumstruktur und dem iterativen Verfahren gut sortieren und effektiv mit den bereits bekannten Verfahren zur Redundanzreduktion codieren. Auch hierbei kann wieder eine Huffman-Codierung verwendet werden.

3.4.7 Der Standard JPEG 2000

JPEG 2000

JPEG 2000 ist der neu entwickelte Bildcodierstandard, der von der *Joint Photographics Expert Group* (JPEG) als Nachfolger des bewährten JPEG-Verfahrens favorisiert wird. Dieser Standard ist in mehreren Teilen als internationaler Standard *ISO/IEC 15444*[24] festgelegt worden [3.14]. Der erste Teil „*Core Coding System*“ basiert im Wesentlichen auf dem zuvor vorgestellten Wavelet-Codierer. Dieser erste Teil des Codierkonzepts ist zwar patentrechtlich geschützt, jedoch ist er lizenz- und abgabenfrei, was seine schnelle Verbreitung begünstigt.

Neben dem ersten Teil (*ISO/IEC 15444-1*: 2000) des Standards existieren noch weitere 10 Teile von *JPEG 2000*, von denen sich noch einige in der Entwicklung befinden. Obwohl sie sich nicht alle mit der Bildcodierung selber beschäftigen, sollen sie hier der Vollständigkeit halber kurz aufgelistet werden:

- Teil 2: Erweiterungen und Verfeinerungen des Codier-Kerns
- Teil 3: *Motion JPEG 2000 – JPEG 2000* als Videokompression
- Teil 4: Test von Implementierungen auf Konformität zum *JPEG 2000* Standard
- Teil 5: Referenzimplementierungen (u. a. in C und Java)
- Teil 6: Zusammengesetzte Bilder, Bildsammlungen
- Teil 7: nicht fortgeführt

24 Auch als *ITU-T T.800...810* bekannt.

- Teil 8: *JPSEC – JPEG 2000* Sicherheitsaspekte
- Teil 9: Interaktive Protokolle
- Teil 10: *JP3D – JPEG 2000* für dreidimensionale Bilder
- Teil 11: *JPWL – JPEG 2000* wireless

Neuerungen JPEG 2000

Die wichtigsten Neuerungen dieses Standards gegenüber früheren Bildcodier-Standards sind:

- Die Codiereffizienz konnte entscheidend gesteigert werden. Es sind sehr kleine Datenraten möglich.
- Verlustlose und verlustbehaftete Kompressionen können in einem Datenstrom gemeinsam integriert werden.
- Die Codierung ist nicht auf rein rechteckige Bilder beschränkt (d. h. beliebig umrandete Bilder können codiert werden).
- Ausgewählte Bildbereiche können in besserer Bildqualität codiert werden (*Enhance Coding: Regions of Interest*).

Auf weitere Details des Standards soll hier nicht eingegangen werden. Es sei jedoch darauf hingewiesen, dass für die Wavelet-Transformation ein 5/3 oder 9/7 *Daubechier-Wavelet* verwendet wird.

Das Codierergebnis ist in Abb. 3.4-9 und Abb. 3.4-10 im Vergleich zum klassischen JPEG-Algorithmus am Beispielbild *Lena* dargestellt.

Abb. 3.4-9: Codierergebnis für Testbild Lena bei 0.18 Bit/Pixel: *Klassisches JPEG* (links), *JPEG 2000* (rechts)

Abb. 3.4-10: Codierergebnis für Testbild *Lena* bei 0.07 Bit/Pixel:
Klassisches JPEG (links), *JPEG 2000* (rechts)

Der wahrnehmbare Qualitätsunterschied ist gerade bei besonders niedrigen Datenraten enorm. In der Praxis kann man davon ausgehen, dass für die gleiche Bildqualität bei *JPEG 2000* eine um den Faktor 2 bis 4 geringeres Datenaufkommen aus bei einem Standard JPEG-Verfahren benötigt wird. Es ist allerdings dabei zu beachten, dass der Rechenaufwand beim *JPEG 2000* System mindestens um den gleichen Faktor höher liegt. Trotzdem ist davon auszugehen, dass sich zunehmend *JPEG 2000* basierte Codiersysteme durchsetzen werden, insbesondere im Bereich der Bewegtbildcodierung für besonders niedrige Datenraten.

Selbsttestaufgabe 3.4-1:

Warum lassen sich mit dem JPEG-Algorithmus prinzipbedingt keine sehr kleinen Datenraten erzielen?

Selbsttestaufgabe 3.4-2:

Warum ist die Wavelet Transformation in der Transformationscodierung besser geeignet als die Diskrete Cosinus Transformation?

Selbsttestaufgabe 3.4-3:

Welche Aufgaben haben die Scaling-Funktionen, welche die Wavelets bei der Wavelet-Transformation?

Selbsttestaufgabe 3.4-4:

Welche Aufgaben haben die Scaling-Funktionen und welche die Wavelets bei der Wavelet-Transformation?

3.5 Einführung in die fraktale Bildcodierung

3.5.1 Die Idee der Selbstähnlichkeit

Die bisher betrachteten Verfahren beruhten auf der Kompression der Transformierten eines Bildes. Dabei entsprachen die Transformationen vorwiegend den Frequenzkomponenten des Bildes. Die fraktale Bildcodierung benutzt im Gegensatz hierzu nur Transformationen (d. h. *Bildmanipulationen*) im Ortsbereich. Erlaubte Operationen sind Stauchen, Verschieben, Vervielfältigen, Drehen und Änderung der Luminanzwerte von ausgewählten Bildbereichen. Die Idee besteht nun darin, die so genannte *Selbstähnlichkeit* im Bild mit diesen Transformationen zu dekorrelieren. Die fraktale Bildcodierung wurde von *M. F. Barnsley* und der Firma *Iterated Systems* 1991 patentiert und implementiert.

Obwohl die Codiereffizienz oft größer ist als beim JPEG-Algorithmus, wird dieses Verfahren derzeit noch in keinem allgemein gebräuchlichen Format eingesetzt. Man kann die fraktale Bildcodierung daher auch als den Exoten unter den Bildcodierern bezeichnen. Nachfolgend werden einige Grundideen vorgestellt. Einzelheiten dieses Verfahrens lassen sich beispielsweise in [3.15] nachlesen.

Das mehrstufige Verfahren muss folgende Schritte für die Kompression eines Bildes vornehmen:

Verfahren Fraktale Bildcodierung

- Bildfeldzerlegung (im einfachsten Fall: rechteckige Blöcke),
- Suche der Selbstähnlichkeiten dieser Blöcke innerhalb des Bildes,
- Beschreibung der Selbstähnlichkeiten durch die affine Transformation,
- Codierung der Transformation.

Die Suche nach der Selbstähnlichkeit im Bild kann sehr aufwändig und rechenintensiv sein. Das erschwert den Einsatz des Encoders in Echtzeitumgebungen. Die Rekonstruktion hingegen ist denkbar einfach: Der Trick besteht darin, bei vollständiger Beschreibung des Bildes allein aus den codierten Transformationsparametern das Ausgangsbild wieder rekonstruieren zu können. Die Beschreibung der einzelnen Bildbereiche durch andere, größere Bildbereiche wird als Funktion angesehen. Die Transformationen sind in der Regel relativ einfach, sodass die Decodierung um ein vielfaches schneller vonstatten gehen kann, als die Codierung. Man spricht hier auch von *unsymmetrischer Codierung.*

3.5.2 Rekonstruktion durch Iteration

Modifizierter Fotokopierer

Die Rücktransformationen werden iterativ auf ein beliebiges Bild von derselben Ausgangsgröße angewendet. Es entsteht eine Annäherung an das Ausgangsbild in Stufen. Abb. 3.5-1 veranschaulicht die Rekonstruktionsvorschrift an dem Modell eines *modifizierten Fotokopierers* (vgl. [3.16]). In diesem Beispiel erfährt das Ausgangsbild vier Transformationen: Einmal eine Stauchung, dann zweimal jeweils zusätzlich eine Verschiebung und eine Rotation, und schließlich eine Verschiebung und zusätzlich eine rein horizontale Stauchung. Diese Bildmanipulation kann man

nun *iterativ* wiederholen. Interessant ist nun, dass dieser iterative Kopiervorgang langsam selbst zu einem Bild hin konvergiert. Dabei ist es völlig unerheblich, welches Ausgangsbild ursprünglich verwendet wird.

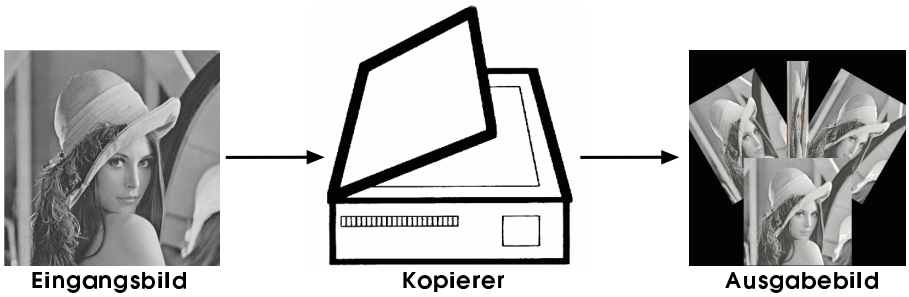


Abb. 3.5-1: Modifizierter Kopierer als Veranschaulichung der fraktalen Bildcodierung

Iteriertes Funktionen System – IFS

Dieser Zusammenhang ist in Abb. 3.5-2 dargestellt: Zwei unterschiedliche Ausgangsbilder ergeben ein und dasselbe Ergebnisbild nach einer endlichen Anzahl von Iterationen. Damit wird deutlich, dass es tatsächlich möglich ist, allein mit geeigneten Bild-Transformationsparametern ein Bild über ein so genanntes *Iteriertes Funktionen System* (*IFS*) zu beschreiben. Die zugrunde liegende Mathematik soll hier nicht weiter aufgerollt werden.

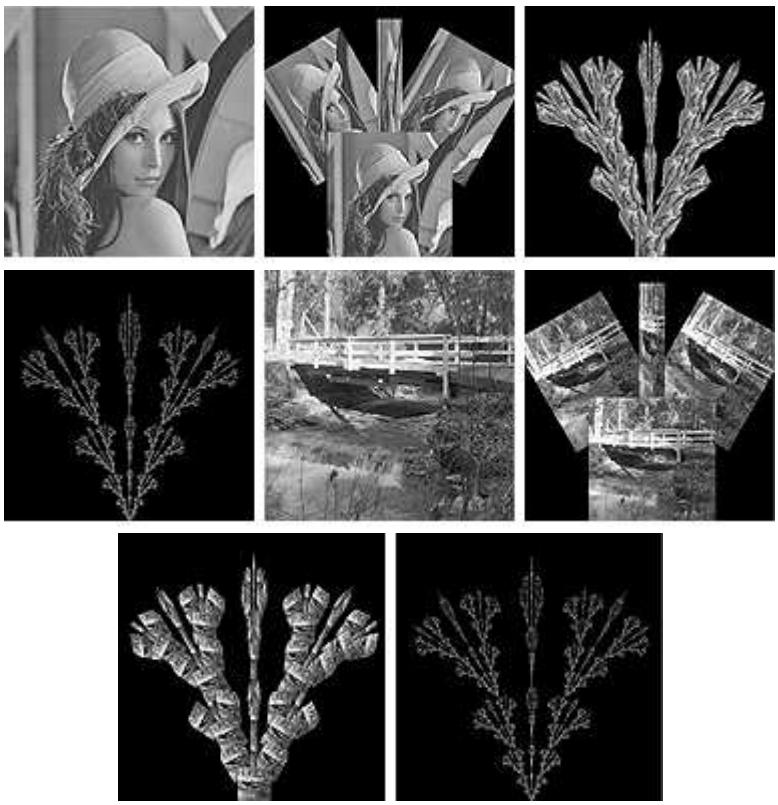


Abb. 3.5-2: Konvergenz bei der fraktalen Bildcodierung: *Blume*
Original, 1., 3. und 8. Iteration, Original, 1., 3. und 8. Iteration (v. l. o. n. r. u.)

Konvergenzkriterium

Es wird aber schon anschaulich klar, dass ein *IFS* nur dann durch die wiederholte Anwendung gegen ein Grenzwertbild konvergiert, wenn die Transformationen

bestimmte Bedingungen erfüllen. Das war der Grund, warum keine Streckungen o. ä., sondern nur *Verkleinerungen* des Bildes als Transformation erlaubt wurden. Man nennt eine solche Transformation auch *kontraktiv*. Das in Abb. 3.5-2 dargestellte Konvergenzbild *Blume* kann offensichtlich schon mit vier einfachen Transformationen vollständig beschrieben werden.

Wie sieht es allerdings aus, wenn fotografische, natürliche Bilder beschrieben werden sollen? Das Finden von Selbstähnlichkeiten in diesen Bildern ist kein einfacher Prozess. Prinzipiell gibt es unendlich viele Möglichkeiten einen endlichen Bildbereich durch andere Bildbereiche zu beschreiben. Da das Verfahren aber nur dann für die Kompression geeignet ist, wenn der Beschreibungsaufwand verringert wird, darf von vornherein nur eine eingeschränkte Anzahl von affinen Transformationen erlaubt sein. Das ist nicht zuletzt auch deshalb notwendig, um den Codieralgorithmus mit vertretbarer Laufzeit implementieren zu können.

3.5.3 Die Partitionierung (Bildfeldzerlegung)

Partitionierung in gleich große Quadrate

Der Algorithmus von *Barnsley* und *Sloan* sieht vor, das Bild gleichmäßig und ohne Überschneidungen in quadratische Blöcke aufzuteilen, die so genannten *Range-Blocks*. Diese Blöcke sollen anschließend durch andere Blöcke doppelter Größe (*Domain-Blocks*) im Bild mit Hilfe einer in ihren Parametern beschränkten affinen Transformation beschrieben werden. Das Überschneiden der *Domain-Blocks* wird zugelassen, da so leichter gut zueinander passende Blöcke zu finden sind. Abb. 3.5-3 zeigt ein Beispiel für eine ermittelte Selbstähnlichkeit in der Bildvorlage *Lena*.

Abb. 3.5-3: Links: Beispiel für eine Selbstähnlichkeit in der Bildvorlage Lena:
Range-Block links (klein); zugehöriger *Domain-Block* rechts (groß)
Rechts: Codierergebnis - IFS (ca. 0.52 Bit pro Pixel)

Affine Transformation

Die Beziehung zwischen jeden Range-Block R_i und Domain-Block D_i geschieht über die *affine Transformation*w_i, wobei diese auch Helligkeit und Kontrast als dritte Dimension *z* berücksichtigt:

$$w_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix}. \qquad 3.5\text{-}1$$

Mit den Parametern a_i, b_i, c_i, d_i, e_i und f_i lassen sich Verschiebung, Rotation, Spiegelung und Größenänderungen in *x*- und *y*-Richtung beschreiben. s_i ist für den Kontrast und o_i für die Helligkeit verantwortlich. In natürlichen Bildvorlagen ist die mathematische Ähnlichkeit zwischen *Domain*- und *Range-Blocks* oft nicht exakt erreichbar. Es wird daher eine *Distanzfunktion* definiert, welche die Güte der Übereinstimmung angibt. Durch Regelung der Schwelle kann dann die Qualität der Beschreibung und damit auch die Datenrate in gewissen Grenzen variiert werden.

Die vorgestellte Bildfeldzerlegung in gleichmäßige, quadratische Blöcke stellt die einfachste Art der Partitionierung dar. Die starke Einschränkung bezüglich der möglichen Transformationen erlaubt zwar eine relativ schnelle Berechnung, doch werden Redundanzen innerhalb des Bildes nicht optimal genutzt.

Quadtree-Partitionierung

Eine Verbesserung des Codierergebnisses kann durch das so genannte *Quadtree-Verfahren* erreicht werden. Die Bildfeldzerlegung geschieht hier adaptiv in verschiede große Quadrate. Bei den zunächst großen Quadraten findet eine weitere Aufspaltung in vier gleich große quadratische Unterblöcke dann statt, wenn zu den *Range-Blocks* kein genügend passender (Schwellenfunktion) *Domain-Block* existiert. Dieser Prozess wiederholt sich, bis zueinander passende Blöcke gefunden werden. Die Zuordnung größerer Flächen mit wenig Detailinformationen geschieht schneller und bei geringerem Speicherverbrauch. Detailreiche Bilder können aber entsprechend längere Zeiten erfordern, da sehr viele Vergleiche gemacht werden müssen. Abb. 3.5-4 zeigt die *Quadtree-Zerlegung* (links) und das Codierergebnis (rechts) bei einem Datenaufkommen von ca. 0.42 Bit pro Pixel.

Abb. 3.5-4: Quadtree-Zerlegung (links) und Codierergebnis (rechts) bei der Bildvorlage *Lena* (ca. 0.42 Bit pro Pixel)

Weitere Methoden zur Partitionierung sind die *HV-Methode* und eine auf Dreieckzerlegung basierende Methode (*Triangular*), die so zusätzlich Objektkanten erfassen kann. Mit der *Triangular-Methode* werden in der Regel die besten Ergebnisse erzielt, der benötigte Rechenaufwand ist jedoch enorm groß.

3.5.4 Messung der Codiereffizienz

Die *Codiereffizienz* der zuvor beschriebenen Codierstandards ist ein wichtiger Qualitätsparameter zur Beurteilung eines Bilddaten Kompressionscoders. Dabei wird als Verzerrungsmaß das *PSNR* in dB für einen Bildbereich *A* wie folgt definiert:

Codiereffizienz: PSNR

$$PSNR_A(s,s') = 10\log_{10}\left(\frac{255^2}{MSE_A(s,s')}\right)$$
$$\text{mit } MSE_A(s,s') = \frac{1}{|A|}\sum_{(x,y)\in A}\left[s(x,y)-s'(x,y)\right]^2, \qquad \text{3.5-2}$$

wobei s das Originalsignal und s' das decodierte Signal sind. Der *PSNR* ist zur Bildqualitätsmessung in der Bildcodierung sehr geläufig, muss aber nicht zwangsläufig der *subjektiv* empfundenen Bildqualität entsprechen. Erfahrungswerte aus der Praxis zeigen, dass bei einem *PSNR* von 35 dB eine für die meisten Betrachter annehmbare Bildqualität wiedergegeben wird.

Rate-Distortion

Trägt man die erzielte Datenrate (in Bit pro Pixel: *bpp*) gegen den *PSNR* (in dB) auf, so ergeben sich die so genannten *Rate-Distortion* Kennlinien, welche die Eigenschaften der Verfahren für verschiedene Kompressionsraten verdeutlichen. Für die Testbildvorlage Lena sind in der nachfolgenden Abb. 3.5-5 die *Rate-Distortion* Kennlinien für den *JPEG-Codieralgorithmus*, den *Wavelet-Codierer* und der *Fraktalen Bildcodierung* gegenübergestellt.

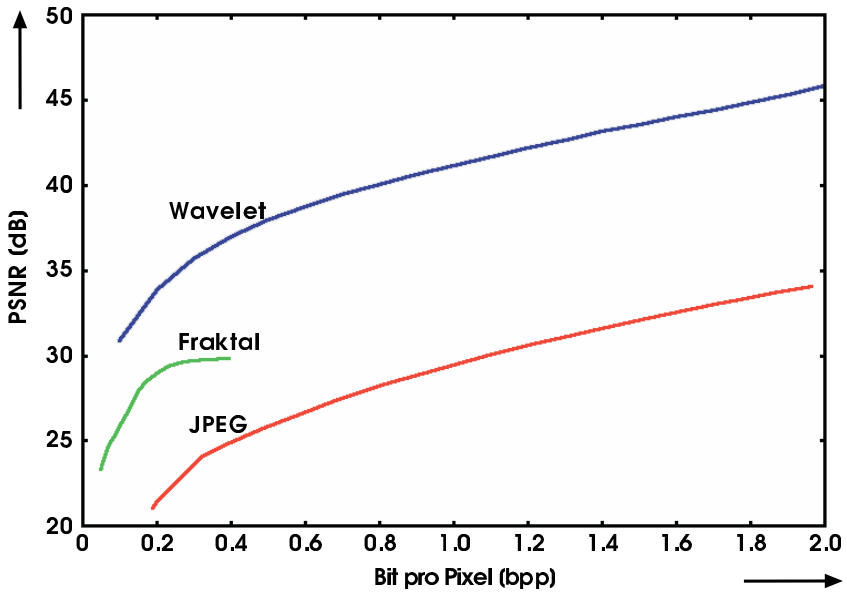


Abb. 3.5-5: *Rate-Distortion* Kennlinien bei der Bildvorlage *Lena*:
JPEG: *Baseline-Algorithmus*
Fraktal: *Quadtree-Codierung*
Wavelet: *Daubechie-Wavelet*

Vergleich

Erkennbar schneidet der *JPEG-Algorithmus* am schlechtesten ab. Es muss allerdings berücksichtigt werden, dass die zugehörigen Kennlinie um ca. 5 bis 8 dB angehoben werden muss, wenn der *progressive JPEG-Algorithmus* verwendet wird, der sich durch eine optimierte Quantisierung auszeichnet.

Die *fraktale Bildcodierung* lässt prinzipbedingt sehr geringe Datenraten zu. Doch zeigt sich auch, dass für qualitativ höherwertige Bilder der Algorithmus in dieser Form nicht geeignet ist. Ab ca. 0.2 bpp aufwärts lassen sich nur marginale Verbesserungen in der Bildqualität erzielen. Nimmt man diese Kurve im Vergleich zu einer progressiven JPEG-Codierung, dann wird deutlich, dass die fraktale Bildcodierung nur bei sehr geringem Datenraten einen Qualitätsvorteil bietet. Durch höherwertige Partitionierungen ließe sich dieses sicherlich ändern, doch ist dann der Rechenaufwand gegenüber einer JPEG-Codierung unverhältnismäßig hoch.

Die *Wavelet-Codierung* schneidet für alle Datenraten besser als ihre Konkurrenten ab. Schon bei Datenraten von 0.1 bpp liefert die Wavelet-Codierung sehr gute visuelle Ergebnisse. Mögliche sichtbaren Artefakte äußern sich durch einen lokal wirkenden Weichzeichner: In diesen Bereichen fehlen die hochfrequenten Bildanteile. Es treten keine Blockartefakte auf. Bei Datenraten größer als 1 bpp ist das codierte Bild nicht mehr vom Original zu unterscheiden (PSNR > 40 dB). Trotzdem lassen sich noch höhere PSNR-Werte erzielen. Eine verlustlose Codierung ist ab etwa 4 bpp möglich.

Die Abb. 3.5-6 und Abb. 3.5-7 zeigen vergleichend das rekonstruierte Codierergebnis der drei Verfahren an Hand der Testbildvorlage *Lena* für 0.18 bpp und 0.07 bpp.

Abb. 3.5-6: Vergleich der Codierqualität bei 0.18 bpp an Hand der Bildvorlage *Lena*: JPEG (links), Fraktal (mitte), Wavelet (rechts, JPEG 2000 kompatibel)

Abb. 3.5-7: Vergleich der Codierqualität bei 0.07 bpp an Hand der Bildvorlage *Lena*: JPEG (links), Fraktal (mitte), Wavelet (rechts, JPEG 2000 kompatibel)

Selbsttestaufgabe 3.5-1:

Nennen Sie die Transformationen, welche bei der fraktalen Bildcodierung allgemein zur Anwendung kommen.

Selbsttestaufgabe 3.5-2:

Nennen Sie zwei Verfahren zur Partitionierung des Bildes für die fraktale Bildcodierung!

Selbsttestaufgabe 3.5-3:

Erläutern Sie die Funktionsweise des „modifizierten Fotokopierers“! Was passiert, wenn man den Kopiervorgang iterativ wiederholt?

Selbsttestaufgabe 3.5-4:

Welche Voraussetzung muss die fraktale Transformation erfüllen, damit die iterative Rekonstruktion gelingt?

Selbsttestaufgabe 3.5-5:

Wie wird die Codiereffizienz allgemein gemessen?

4 Prinzipien der digitalen Quellencodierung für Bewegtbilder

Codierverfahren für Bewegtbilder lassen sich auf der Basis der bereits vorgestellten Verfahren für Einzelbilder aufbauen. Im einfachsten Fall verwendet man für jedes Bild der Bewegtbildsequenz ein bekanntes Codierverfahren für Einzelbilder. Solche Verfahren werden im nachfolgenden Abschnitt 4.1 behandelt. In Abschnitt 4.2 und Abschnitt 4.3 werden Verfahren vorgestellt, welche zusätzlich die zeitliche Korrelation zwischen den Einzelbildern der betrachteten Sequenz nutzen („*zeitliche Prädiktion*“).

4.1 Bewegtbildcodierung auf der Basis von Einzelbildcodierung

Ein einfaches Konzept zur Codierung von Bewegtbildsequenzen besteht darin, jedes Einzelbild unabhängig von den übrigen mit einem bekannten Verfahren zu beschreiben. Man spricht in diesen Fällen von einer so genannten „*Intra-Codierung*“. Wichtige Beispiele hierfür sind *Motion-JPEG* und die *DV-Codierung*, welche auch für die digitale Magnetbandaufzeichnung geeignet ist.

4.1.1 Von JPEG nach Motion-JPEG

Im Kapitel 3 wurde gezeigt, dass der JPEG-Standard ein leistungsfähiges und flexibles Verfahren zur Codierung von Einzelbildern darstellt. Es ist daher naheliegend, diesen Standard auch für die Bewegtbildcodierung zu verwenden. Um eine mittlere konstante Datenrate zu erzielen, wird beim Motion-JPEG Verfahren eine adaptive Codierung mit Hilfe eines Datenpuffers am Ausgang und einer Rückkopplung zur Quantisierungsstufe realisiert. So ist eine Anpassung der Quantisierung in Abhängigkeit vom Pufferfüllstand erreichbar.

In nichtlinearen Schnittsystemen werden heutzutage oft solche Konzepte zur Datenreduktion eingesetzt. Leider konnte sich kein einheitlicher Standard durchsetzen, so dass die existierenden Lösungen, die eine *Intra-Codierung* mit dem JPEG-Standard verwenden, nicht zueinander kompatibel sind. Oft wird diese nicht standardisierte Gruppe von Codierungen unter dem Oberbegriff *Motion-JPEG* zusammengefasst. **Motion-JPEG**

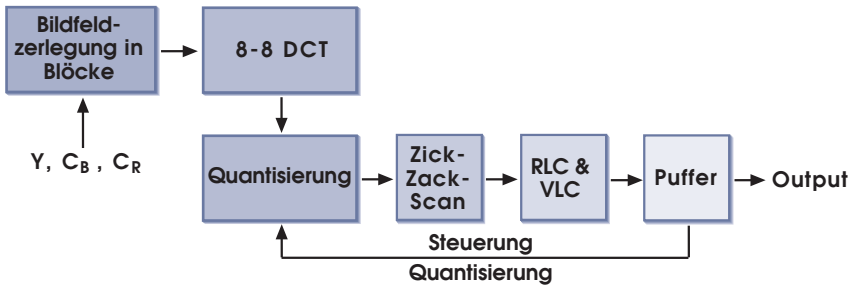


Abb. 4.1-1: Grundkonzept von *Motion-JPEG*

Abb. 4.1-1 zeigt das Grundkonzept von *Motion-JPEG*[25]. Man erkennt die von JPEG bekannten Blöcke für die Bildfeldzerlegung, DCT, Quantisierung, Zick-Zack-Scan und Redundanzreduktion (RLC und VLC).

Steuerung der Datenrate

Das Besondere an *Motion-JPEG* ist, dass in Abhängigkeit des Füllstandes im Ausgangsdatenpuffer die Quantisierung verändert wird: Bei drohendem Leerlauf des Puffers wird feiner quantisiert (Datenrate steigt), im umgekehrten Fall wird gröber quantisiert (Datenrate sinkt). Damit ist prinzipiell keine von Einzelbild zu Einzelbild konstante Bildqualität möglich. Diese Schwankungen lassen sich durch geschicktes Puffermanagement weitgehend so minimieren, dass eine Veränderung der Bildqualität unter normalen Bedingungen nicht sichtbar wird.

4.1.2 Die DV-Codierung

Ein wichtiges auf der *Motion-JPEG* Idee aufbauendes System ist die Codierung nach dem *DV-Standard* (*DV - Digital Video*), welcher für Datenraten von 25 Mbit/s (*DV25*: *DV*, *DVCAM*, *DVCPRO 25*), 50 Mbit/s (*DV50*: *DVCPRO 50*) und 100 Mbit/s (*DV100*: *DVCPRO HD*) als Standard existiert. Nachfolgend wird das Prinzip der *DV25 Codierung* erläutert (vgl. auch [4.3], [4.7], [4.11], [4.12], [4.13]).

DV25 Codierung

Ursprünglich wurde dieser Standard besonders für die Speicherung von Videodaten mit einer Nettorate von 25 Mbit/s auf Magnetband konzipiert[26]. Abb. 4.1-2 zeigt das Funktionsdiagramm.

25 Viele Anwendungen umgehen das Standardisierungsproblem mit *Motion-JPEG*, indem sie einen speziellen Modus von *MPEG*, nämlich „*MPEG-I-Frame only*“ nutzen, der *Motion-JPEG* recht ähnlich ist (vgl. Kapitel 5).

26 Die Norm IEC 61834 (*Helical-Scan Digital Video Cassette Recording System Using 6.35 mm Magnetic Tape for Consumer Use*) ist für Konsumer-Anwendungen und die SMPTE 314M (*Standard for Television – Data Structure for DV-Based Audio, Data and Compressed Video – 25 and 50 Mbit/s*) ist für den professionellen Bereich vorgesehen.

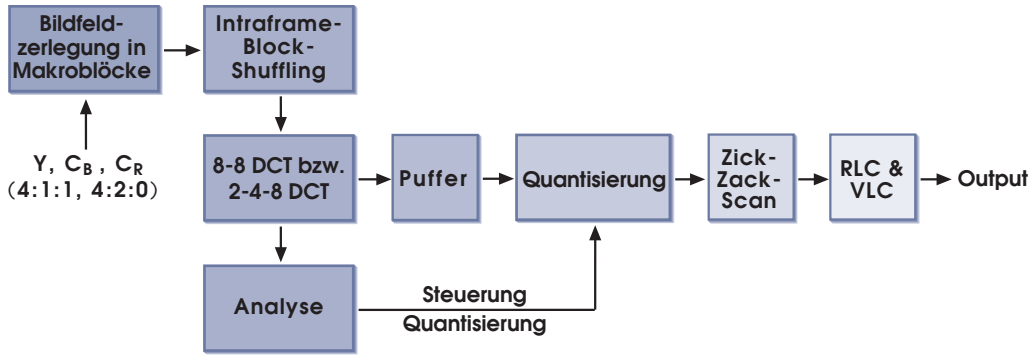


Abb. 4.1-2: Funktionsdiagramm für die *DV25 Codierung*

Feed-Forward / Feed-Backward Steuerung

Gegenüber dem in Abb. 4.1-1 dargestellten Algorithmus zur *Motion-JPEG* Codierung liegen die Unterschiede insbesondere in der Steuerung der Quantisierung. Während die Regulierung der Datenrate bei *Motion-JPEG* über den zurückgekoppelten Pufferfüllstand geschieht („*Feed-Backward*“), wird beim DV25 Algorithmus wird das zu erwartende Datenaufkommen im *vorhinein* ermittelt („*Feed-Forward*“). Nachfolgend wird auf die Besonderheiten der DV25 Codierung näher eingegangen.

Rastertransformation von 4:2:2 auf 4:1:1 bzw. 4:2:0

Der Studiostandard ITU-R BT.601 besitzt eine Farbabtastung von 4:2:2 (vgl. Abschnitt 2.2.2). Für die DV25 Codierung werden die Farbkomponenten ein weiteres Mal örtlich unterabgetastet. Gemäß den Definitionen aus Abschnitt 2.2.1 ist entweder eine Abtastung nach 4:1:1 (*Konsumer-DV* / *DVCAM* in NTSC-Ländern und *DVCPRO 25*, *D-7*) oder 4:2:0 (*Konsumer-DV* / *DVCAM* in PAL-Ländern) üblich.

Block-Shuffling

Block-Shuffling

Jedes Einzelbild des Videodatenstromes wird ähnlich wie beim JPEG-Verfahren in quadratische Bildblöcke unterteilt. Allerdings werden innerhalb von so genannten Makroblöcken (*M*) vier benachbarte 8 x 8 Pixelblöcke des Luminanzsignals und die zwei zugehörigen 8 x 8 Pixelblöcke des Chrominanzsignals (*C*B und *C*R) zusammengefasst. Ein Makroblock enthält so sechs Pixelblöcke zu je 64 Bytes, die im Anschluss jeweils getrennt DCT transformiert werden. Abb. 4.1-3 zeigt den Aufbau eines Makroblockes für eine Abtastung nach 4:1:1.

Makroblock bei 4:1:1

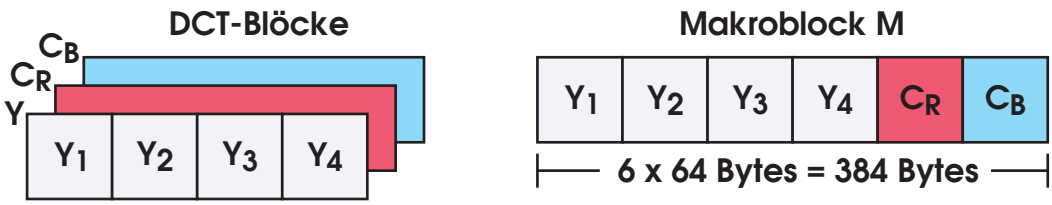


Abb. 4.1-3: Makroblockbildung bei der DV25 Codierung (4:1:1 Abtastung)

Für eine optimale DV-Magnetbandaufzeichnung sind einige Randbedingungen bei der Codierung zu beachten. So ist ein konstantes Datenaufkommen besonders wichtig, damit die Aufzeichnung der Videospuren nach einem festen Zeitraster geschehen kann. Wie in Abschnitt 3.2.3 ausgeführt wurde, führt der JPEG-Algorithmus selbst nicht zu exakt gleich großen Datenblöcken pro Bild. Würde man jeden DCT-Block einzeln so quantisieren, dass ein vorgegebenes Datenaufkommen erreicht wird, hätte man in homogenen Bildbereichen eine überdurchschnittlich gute, in fein strukturierten jedoch nur eine mäßige Bildqualität. Daher wird versucht, die Forderung nach konstantem Datenaufkommen über mehrere in der Struktur möglichst verschiedene DCT-Blöcke zu verteilen.

Videosegment vor Kompression

Die DV25 Codierung erreicht dies, indem fünf Makroblöcke aus unterschiedlichen Bildbereichen zu so genannten *Videosegmenten* zusammenfasst (5 x 384 Bytes ergeben 1920 Bytes) werden. Abb. 4.1-4 veranschaulicht die Auswahl und die Zusammenfassung der Makroblöcke für ein Videosegment vor der eigentlichen Kompression. Dieses Verfahren nennt man *Shuffling*. Innerhalb eines Videosegmentes korrespondieren benachbarte Makroblöcke also nicht mit benachbarten Bildbereichen. Die Verwürfelung führt dazu, dass jeder der fünf Makroblöcke eines Videosegments in seinem Gehalt an Redundanz und Irrelevanz sehr unterschiedlich sein kann. Bezogen auf alle Videosegmente eines einzelnen Videobildes können mit diesem Verfahren jedoch die Redundanzunterschiede minimiert werden. So kann die Forderung leichter erfüllt werden, für jedes Videosegment eine mittlere, konstante Datenkompression zu erreichen.

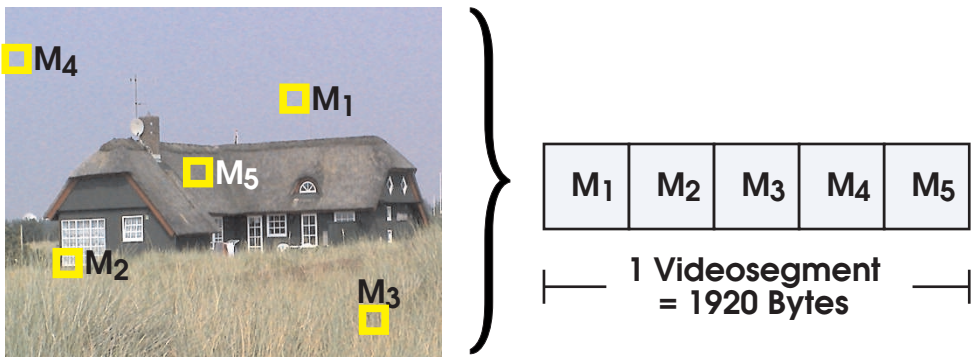


Abb. 4.1-4: Zur Bildung des Videosegments vor der Kompression

Zusammengefasst bewirkt das Block-Shuffling durch eine gleichmäßige Verteilung der Codiereffekte auf Bildbereiche von unterschiedlicher „*Komprimierfähigkeit*“ insgesamt eine Verbesserung der Bildqualität. Überdies ist eine möglichst gleichmäßige Energieverteilung im Datenstrom auch für die Fehlerschutzcodierung günstig: Nichtkorrigierbare Datenfehler bei der Übertragung äußern sich im Bild durch kleinere, örtlich verteilte Störungen (vgl. Kapitel 5).

Codierung von Halbbildern

Probleme der Zeilensprungabtastung

Aufgrund der besonderen Abtastung (Zeilensprungabtastung, vgl. Abschnitt 1.4.1) bestehen Videoeinzelbilder normalerweise aus zwei Halbbildern (*Field*), welche zeilenweise alternierend zu zwei verschiedenen Bewegungsphasen gehören. Die bei vielen Kompressionsverfahren übliche Bildfeldzerlegung in quadratische Blöcke führt hier leicht zu Problemen. So verursachen schnelle Bewegungen hohe vertikale Ortsfrequenzen in den DCT-Blöcken, was eine effiziente Kompression erschwert.

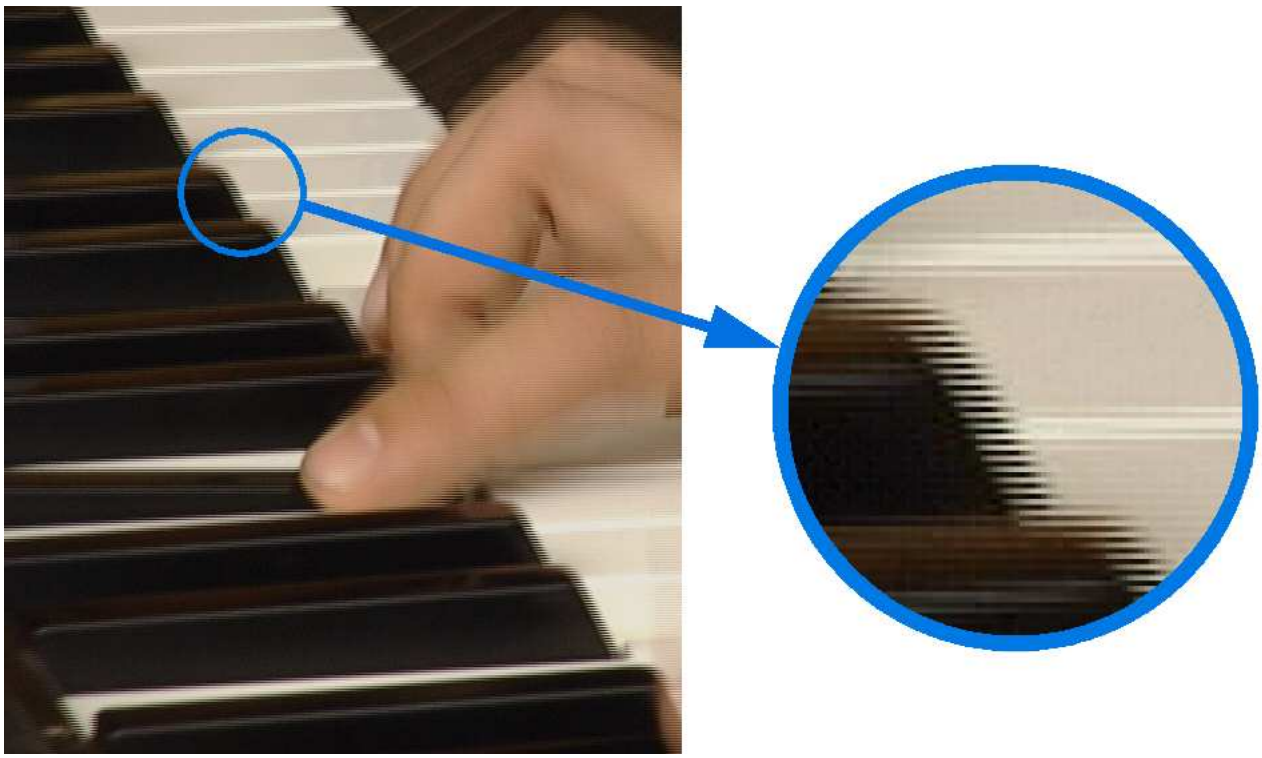


Abb. 4.1-5: Probleme der Zeilensprungabtastung bei der Bildcodierung

Abb. 4.1-5 veranschaulicht dieses oft unterschätzte Problem mit der *Zähnchenstruktur* aufgrund des Zeilensprungs. Ersichtlich darf man bei der Codierung von Zeilensprungmaterial mit viel Bewegung die Bildfeldzerlegung für die DCT-Blockbildung nur auf der Basis der Halbbilder durchführen. Allerdings ist dann eine Nutzung der vertikalen Korrelation für die Codierung nur noch eingeschränkt möglich, da lediglich die Hälfte der Zeilen herangezogen werden kann. Daher nutzt die DV25 Codierung eine adaptive Technik zur Behandlung von Halbbildern. Zunächst werden - wie beim JPEG-Verfahren - die 8 x 8 Pixel großen Bildblöcke gebildet. Dabei entstammen die Pixel des Blockes zeilenweise alternierend dem ersten bzw. zweiten Halbbild.

Field-Mode / Frame-Mode

Für jeden 8 x 8-Pixelblock wird anschließend separat entschieden, ob er im so genannten *Halbbild-Mode* (*Field-Mode*) oder im *Vollbild-Mode* (*Frame-Mode*) codiert werden soll. Im Falle des Vollbild-Modes wird die DCT normal für den 8 x 8-Pixelblock ausgeführt (*8-8 DCT*). Im Halbbild-Mode wird jeder 8 x 8-Pixelblock in zwei 4 x 8-Blöcke aufgeteilt (*2-4-8 DCT*), der erste Block (*Block A*) wird aus der paarweisen Summe der Zeilen des ersten und zweiten Halbbildes gebildet, der zweite Block (*Block B*) aus der Differenz:

$$B_{Ai} = (Z_{ai} + Z_{bi})/2 \qquad \text{4.1-1}$$

$$B_{Bi} = Z_{ai} - Z_{bi} + 128 \qquad \text{4.1-1}$$

mit $Z_{ai},\ Z_{bi}$i-te Zeile des 1. bzw. 2. Halbbildes und

$B_{Ai},\ B_{Bi}$als i-te Zeile des Blockes Abzw.$B,\ i = 1, 2, 3, 4.$

Beide Blöcke werden dann getrennt in die DCT-Ebene transformiert. Dieser Mode kommt bei bewegten, feinstrukturierten Bildbereichen zum Einsatz. In den ruhenden Bildbereichen existiert eine größere örtliche Korrelation zwischen den beiden Halbbildern. Daher bringt hier der Vollbild-Mode Vorteile. Darüber hinaus ist der Vollbild-Mode oft auch in bewegten Bildbereichen nutzbar, sofern diese nur wenig oder gar keine feine Strukturen aufweisen. Dies führt meist zu einer Qualitätsverbesserung in den Übergangsbereichen zwischen ruhenden und bewegten Bildelementen.

Die bewegungsabhängige Umsteuerung zwischen Halbbild-Mode und Vollbild-Mode wird oft mithilfe einer einfachen Messung der Kreuzkorrelation zwischen den beiden Halbbildern realisiert. Da hier lediglich eine Schwellenentscheidung vorgenommen werden muss, ist eine tatsächliche Bewegungsschätzung, wie sie in Abschnitt 4.3 beschrieben wird, nicht zwingend erforderlich. Daher kann auch ein einfacher örtlicher Vergleich der Luminanzwerte der beiden Halbbilder als Entscheidungskriterium herangezogen werden. Abb. 4.1-6 verdeutlicht die Umsteuerstrategie für den Halbbild- und Vollbild-Mode. Im Halbbild-Mode werden die Blöcke A und B nach Gl.4.1-1 und Gl.4.1-1 aus den beiden Halbbildern gebildet.

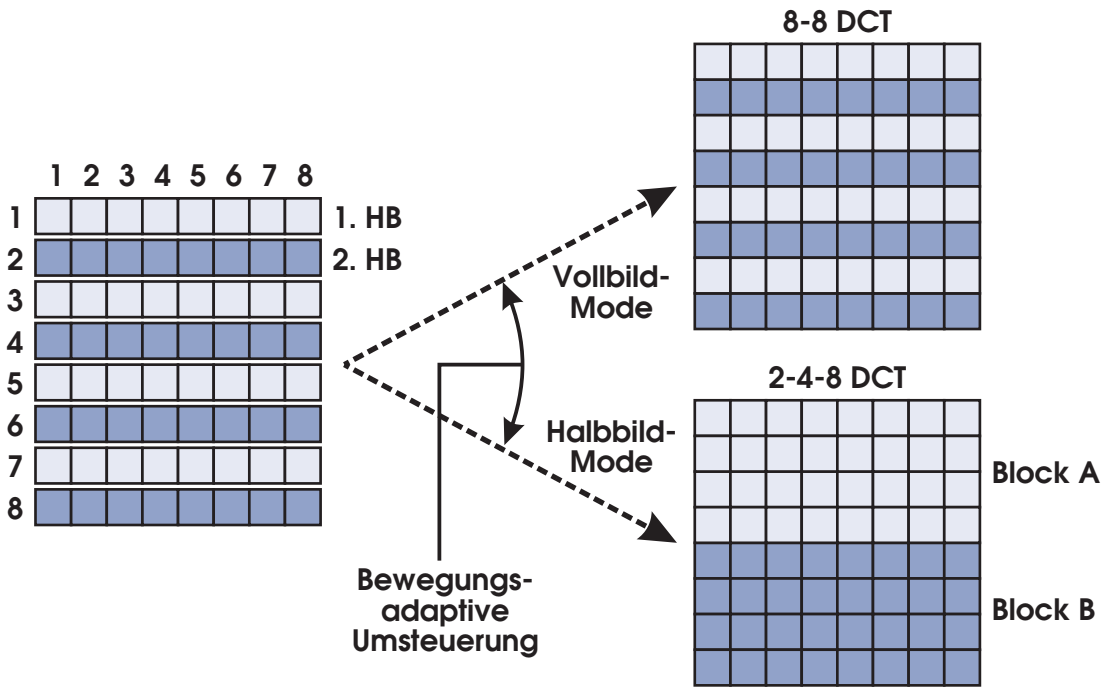


Abb. 4.1-6: Umsteuerstrategie zwischen Halbbild- und Vollbildmode bei der DV25 Codierung

Die Entscheidung für den Mode wird für jeden 8 x 8 Block getrennt getroffen. Damit optimiert sich der Codiervorgang auch für Fälle, bei denen gleichermaßen bewegte und unbewegte Bildelemente vorhanden sind. Insgesamt verbindet demnach die DV25 Codierung die Vorteile von Halbbild- und Vollbild-Mode zur Verbesserung des Kompressionsergebnisses.

Feed-Forward Steuerung der Quantisierung

Quantisierung: Stufen und Gruppen

Wie in Abb. 4.1-2 dargestellt verwendet die DV25 Codierung eine Vorabanalyse der DCT-Koeffizienten, um die Quantisierung zu steuern. Diese Vorgehensweise nennt man „*Feed-Forward*“ Steuerung. Ziel ist es, vor der eigentlichen Kompression das Datenaufkommen bei Anwendung von verschiedenen Quantisierungstabellen zu ermitteln. Kriterium für die optimale Wahl einer Tabelle ist einerseits das visuell beste Ergebnis und andererseits die Erzielung einer konstanten Datenrate je Videosegment, die kleiner, aber möglichst nahe an 385 Bytes liegt. Für die Quantisierung stehen jedem Makroblock 16 Quantisierungsstufen zur Verfügung. Die 6 DCT-Blöcke können innerhalb von 4 Gruppen unterschiedlich quantisiert werden. Die vier Gruppen charakterisieren grob die spektrale Energieverteilung im betrachteten Block. Die Zuordnung eines Blockes in eine der vier Gruppen geschieht anhand des betragsgrößten AC-DCT-Koeffizienten. Gruppe 3 ist für sehr fein strukturierten, Gruppe 0 für groben bzw. homogenen Bildinhalt optimiert. Die Gruppen 1 und 2 decken die dazwischen liegenden Fälle ab. Nach dieser Vorgruppierung wird für jeden Block mit 16 Quantisierungstabellen eine virtuelle Kompression durchgerechnet und dabei die jeweilige Anzahl der Bytes je Videosegment ermittelt. Die eigentliche Kompression geschieht dann mit der Quantisierung, die eine Kompres-

sion erzeugt, die am dichtesten an dem Grenzwert für ein Videosegment von 385 Bytes liegt, ihn aber nicht überschreitet.

Konkret wird der DC-Koeffizient in einem DCT-Block zunächst mit 9 Bits quantisiert (Wertebereich –255 bis 255). Die AC-Koeffizienten werden bei der DCT-Transformation auf 10 Bits in einen Wertebereich zwischen –511 und +511 abgebildet. Jeder DCT-Block wird gemäß der obigen Überlegung in Abhängigkeit des größten AC-Koeffizienten einer der vier Gruppen zugeordnet. Bei den AC-Koeffizienten aus der Gruppe 3 wird dann das LSB (*least significant bit*) gestrichen. Für die Gruppe 0 bis 2 sind, gemäß der Zuordnungsvorschrift für die Gruppen, die AC-Koeffizienten niemals größer als 255, sodass in diesen Gruppen das MSB (*most significant bit*) der AC-Koeffizienten ohne Informationsverlust gestrichen werden kann. Damit haben alle Koeffizienten des DCT-Blockes eine 9 Bit lange Wortbreite.

Quantisierung: Bereichsnummern

Jeder AC-Koeffizient innerhalb eines DCT-Blockes wird darüber hinaus einer *Bereichsnummer* zugeordnet, welche später in der Quantisierungstabelle über den Quantisierungsgrad entscheidet. In Abb. 4.1-7 sind diese Bereichsnummern für die 8-8 DCT und die 2-4-8 DCT aufgeführt.

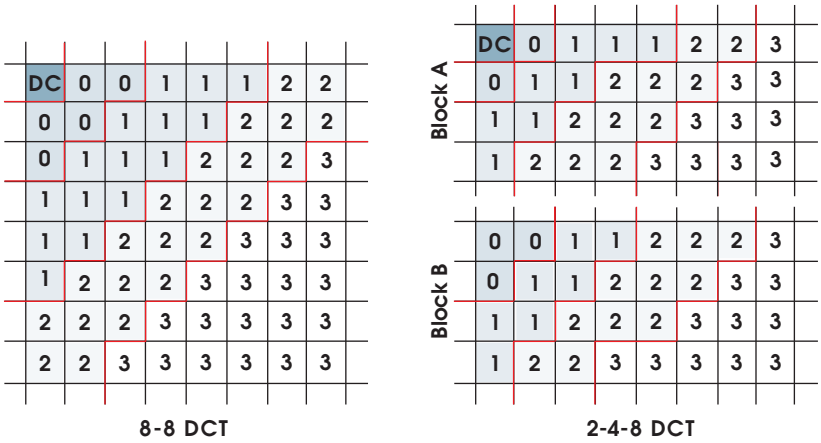


Abb. 4.1-7: Zuordnung von Bereichsnummern für die Quantisierung

Virtuelle Kompression

Mit der *Gruppennummer* und der *Bereichsnummer* können dann 16 verschiedene *Quantisierungsstufen* (*Quantization number – QNO*) für die *virtuelle Kompression* probiert und das Ergebnis in Bezug auf die erzielte Datenrate getestet werden. Tabelle 4.1-1 und Tabelle 4.1-4 zeigen den Divisionsfaktor für die AC-Koeffizienten in Abhängigkeit von der *Gruppen-* und *Bereichsnummer*.

Tab. 4.1-1: *Divisionsfaktoren für die AC-Koeffizienten – Gruppe 0*

Gruppe 0																
	Quantisierungsstufe															
Bereichs-nummer	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
1	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1	1
2	4	4	4	4	2	2	2	2	1	1	1	1	1	1	1	1
3	8	8	4	4	4	4	2	2	2	1	1	1	1	1	1	1

Tab. 4.1-2: *Divisionsfaktoren für die AC-Koeffizienten – Gruppe 1*

Gruppe 1																
	Quantisierungsstufe															
Bereichs-nummer	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	4	4	4	2	2	2	2	1	1	1	1	1	1	1	1	1
1	8	4	4	4	4	2	2	2	2	1	1	1	1	1	1	1
2	8	8	8	4	4	4	4	2	2	2	2	1	1	1	1	1
3	16	8	8	8	8	4	4	4	4	2	2	2	1	1	1	1

Tab. 4.1-3: *Divisionsfaktoren für die AC-Koeffizienten – Gruppe 2*

Gruppe 2																
	Quantisierungsstufe															
Bereichs-nummer	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	8	8	4	4	4	4	2	2	2	2	1	1	1	1	1	1
1	8	8	8	8	4	4	4	4	2	2	2	2	1	1	1	1
2	16	16	8	8	8	8	4	4	4	4	2	2	2	2	1	1
3	16	16	16	16	8	8	8	8	4	4	4	4	2	2	2	1

Tab. 4.1-4: *Divisionsfaktoren für die AC-Koeffizienten – Gruppe 3*

Gruppe 3																
	Quantisierungsstufe															
Bereichs-nummer	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	8	4	4	4	4	2	2	2	2	1	1	1	1	1	1	1
1	8	8	8	8	4	4	4	4	2	2	2	2	1	1	1	1
2	16	8	8	8	8	4	4	4	4	2	2	2	2	1	1	1
3	16	16	16	8	8	8	8	4	4	4	4	2	2	2	1	1

Die Tabellen geben den Divisionsfaktor für jeden AC-Koeffizienten an. Beispiel: Für die *Quantisierungsstufe* 6 werden in einem Gruppe-2 Block Koeffizienten des Bereiches 0 durch 2, die der Bereiche 2 und 3 durch 4 und die des Bereiches 3 durch 8 dividiert. Die gleiche *Quantisierungsstufe* führt in einem Gruppe-1 Block zu den Divisionsfaktoren 2, 2, 4 und 4. An dieser Stelle sei noch einmal angemerkt, dass alle 6 DCT-Blöcke eines Makroblockes die gleiche *Quantisierungsstufe* haben müssen, aber jeweils verschiedenen *Gruppen* zugeordnet sein können.

Struktur des komprimierten Datenstroms

Die Koeffizienten werden bei den 8-8 DCT-Blöcken mit dem bekannten Zick-Zack-Scan ausgelesen; die Zusammenfassung der Koeffizienten der 2-4-8 DCT-Blöcke geschieht nach einem leicht modifizierten Verfahren. Die sich anschließende Redundanzreduktion (*RLC* und *VLC*) ist für das DV25 Verfahren optimiert worden. Wie beim JPEG-Algorithmus wird auch hier ein 4 Bit langes *end-of-block* (EOB: 0110) Codewort eingefügt, wenn innerhalb des DCT-Blockes keine weiteren Koeffizienten ungleich null existieren. Abb. 4.1-8 zeigt noch einmal die Struktur des Videosegments nach der Komprimierung.

Videosegment nach Kompression

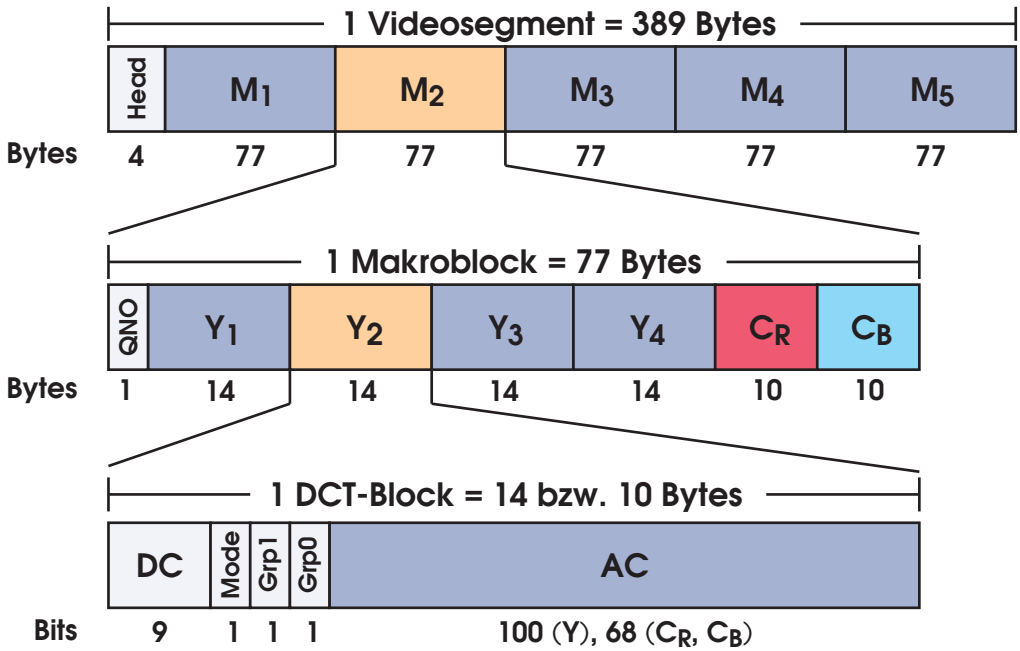


Abb. 4.1-8: Aufbau des DV25 Videosegmentes nach der Kompression ***Head***: Identifizierung des Videosegmentes, $M1$ ... $M5$: Makroblock 1 bis 5
$Y1$ ... $Y4$: DCT-Blöcke Luminanz, C_R, C_B: DCT-Blöcke Chrominanz
QNO: *Quantization number* (4 Bit) und *Error Status* (4 Bit)
DC: DC-Koeffizient, ***AC***: AC-Koeffizient, ***Mode***: Halbbild-Mode / Vollbild-Mode
Grp0, ***Grp1***: Gruppennummer für die Quantisierung auf Makroblockebene

3-Stufen-Algorithmus

Erwartungsgemäß wird der Algorithmus mit den vorgegebenen Randbedingungen nicht in allen DCT-Blöcken ein Datenaufkommen von exakt 100 bzw. 68 Bit mit der variablen Lauflängencodierung erreichen. In einigen Blöcken wird ein *EOB* Codewort bereits weit vor dem Ende des verfügbaren Speicherplatzes eingefügt werden

können, bei anderen reicht dieser möglicherweise nur für eine sehr grobe Quantisierung aus. Dieses Problem löst die DV25 Codierung mithilfe eines komplexen *3-Stufen Algorithmus*. Im ersten Durchlauf wird die Quantisierungsstufe so gewählt, dass die Summe des Datenüberhanges nach der Kompression in den Makroblöcken eines Videosegmentes gerade so groß ist wie der ungenutzte Platz in den übrigen Makroblöcken des gleichen Videosegmentes. Damit ist zunächst die Basisforderung erfüllt, dass die Summe der erforderlichen Daten nicht größer als 385 Bytes pro Videosegment wird. Im zweiten und dritten Durchlauf werden die Datenüberhänge sukzessiv in den freien Plätzen hinter den *EOB* Codeworten untergebracht.

Die zuvor beschriebene Vorausberechnung (*Virtuelle Kompression*) benötigt eine gewisse Rechenzeit, da für jede der 16 möglichen Quantisierungsstufen die Schritte Quantisierung, Auslesen der quantisierten Koeffizienten und Anwendung der RLC und VLC durchlaufen werden müssen, bevor das tatsächliche Datenaufkommen des Videosegmentes bekannt ist. Diese Verzögerung wird durch das Einfügen eines Puffers als Zwischenspeicher zwischen DCT und eigentlicher Redundanzreduktion ausgeglichen (vgl. Abb. 4.1-2).

Berechnung der Kompressionsrate

Kompressionsrate

Wird ein Studiosignal nach ITU-R BT.601 DV25 codiert, ergeben sich für ein 625/50 Hz System mit 576 aktiven Zeilen und 720 Bildpunkten je Zeile insgesamt 1620 Makroblöcke bzw. 324 Videosegmente pro Vollbild; für ein 525/60 Hz System mit 480 Zeilen und 720 Bildpunkten je Zeile errechnen sich 1350 Makroblöcke bzw. 270 Videosegmente pro Vollbild. Die Videodatenrate ist in beiden Fällen exakt gleich:

$$R_{625/50} = 324 \cdot 389 \cdot 25Hz \cdot 8Bit = 25207200\ Bit/s \qquad \text{4.1-2}$$

bzw.

$$R_{525/60} = 270 \cdot 389 \cdot 30Hz \cdot 8Bit = 25207200\ Bit/s \qquad \text{4.1-3}$$

Da die 4:1:1- bzw. 4:2:0-Farbabtastung das ITU-R BT.601-Signal auf eine Datenrate von etwa 125 Mbit/s reduziert, erzielt der DV25 Codieralgorithmus selbst eine Kompressionsrate von 5:1.

DV50 und *DV100*

DV-Codierung: Erweiterungen

Obwohl die zuvor dargestellte DV25 Codierung sehr beliebt ist, reicht sie für die Qualitätsanforderungen in der professionellen Praxis oftmals nicht aus. So stellt die vierfache Farbunterabtastung für die elektronische Tricknachbearbeitung (z. B. Blue-Box-Verfahren) ein Problem dar. Daher existiert neben dem DV25 Standard

auch eine Norm, welche mit der doppelten Datenrate, also 50 Mbit/s arbeitet (Beispiel: *DVCPRO50*, *D-9*). Mit der originalen Farbabtastung des ITU-R BT.601 Standards von 4:2:2 wird die Ausgangsdatenrate von 158 Mbit/s auf 50 Mbit/s reduziert. Dies entspricht einer Kompressionsrate von 3.2:1.

Der Makroblock für eine 4:2:2 DV-Codierung hat nur 4 Blöcke: Y_1, Y_2, C_R, C_B. Bei der Quantisierung ist für die Koeffizienten erheblich mehr Datenplatz vorhanden. Das führt zu einer Steigerung der Bildqualität. Die weiteren Verarbeitungsschritte decken sich im wesentlichen mit den oben beschriebenen Verfahren für die DV25 Codierung. In der Praxis werden für die Realisierung der DV50 Codierung einfach zwei DV25 Codierschaltkreise „parallel“ geschaltet. Jeder verarbeitet dann einen 2:1:1 Datenstrom. Das führt zu der Notwendigkeit, kleinere Änderungen in der Syntax des 4:2:2 50 Mbit/s Datenstroms vorzunehmen.

Ebenfalls auf 50 Mbit/s wird progressiv abgetastetes Bildmaterial (50 Hz, 576 Zeilen, 720 Pixel pro Zeile, 4:2:0) komprimiert (*DVCPRO P*). Die Kompressionsrate ist hier wie bei der DV25 Codierung 5:1.

Für hochaufgelöstes Video wurden ebenfalls mehrere DV-basierte Codiervorschläge entwickelt (*Konsumer DV-HD* und *Professional DV-HD*). So wird beispielsweise beim *DVCPRO HD* (*D-12*) Standard ein Videosignal (60 Hz, 720 Zeilen progressiv, 1920 Pixel pro Zeile, 4:2:2) auf 100 Mbit/s komprimiert. Das entspricht einer Kompressionsrate von 1:6.7. Weitere ausführliche Details zur DV-Codierung lassen sich beispielsweise in [4.4] und [4.14] finden.

Selbsttestaufgabe 4.1-1:

Warum ist die Regelung des komprimierten Datenstroms mit Hilfe einer FeedBackward-Steuerung für die DV25 Codierung nicht geeignet?

Selbsttestaufgabe 4.1-2:

Wie sieht die Makroblockbildung für die DV25 Codierung nach der 4:2:0 Abtastung aus?

Selbsttestaufgabe 4.1-3:

Welches Ziel verfolgt das Block-Shuffling bei der DV25 Codierung?

Selbsttestaufgabe 4.1-4:

Wozu dienen bei der DV25 Quantisierung die Quantisierungsstufen, wozu die Gruppen?

4.2 Prädiktive Bildcodierung mit DPCM

Intraframe-Codierung
Interframe-Codierung

Die in den vorigen Kapiteln vorgestellten Codierverfahren nutzen für die Kompression im wesentlichen die örtliche Redundanz innerhalb der Einzelbilder aus. Bei dieser so genannten *Intraframe-Codierung* spielen die zeitlich-örtlichen Ähnlichkeiten zwischen aufeinanderfolgenden Einzelbildern eines Videodatenstromes keine Rolle. Wenn man sich jedoch verdeutlicht, wie viele Bildbereiche in aufeinanderfolgenden Bildern konstant oder zumindest ähnlich bleiben, wird klar, warum die zeitliche Redundanzreduktion für die digitalen Bildcodierung so bedeutend ist. Damit können neben örtlich redundanten auch zeitlich redundante Bildanteile für die Codierung eliminiert werden. Diese Codierung wird im Unterschied zu den bisher vorgestellten Verfahren *Interframe-Codierung* genannt. Der Realisierungsaufwand ist für Coder und Decoder größer. So müssen beispielsweise bei der Decodierung die zeitlich redundanten Anteile aus einem Bildspeicher heraus rekonstruiert werden.

Prädiktion

Verallgemeinert wird bei dieser Art der Codierung nur der Teil der Bildinformation übertragen, der am Decoder nicht durch eine geeignete Strategie aus der bisher vorhandenen Bildinformation vorhersagt werden kann. Eine solche Strategie nennt man *Prädiktion*.

4.2.1 Feed-Forward DPCM - D*PCM

D*PCM / DPCM

In der Bildverarbeitung haben sich Strategien bei der Codierung bewährt, die auf der so genannten *Differential Pulse Code Modulation – DPCM* aufbauen. Bei den Grundschaltungen zur DPCM unterscheidet man zwischen der *D*PCM* (auch *Feed-Forward DPCM*) und der *DPCM* (auch *Feed-Backward DPCM*).

Die *PCM* selbst ist ein einfaches Übertragungsverfahren, bei der die Codierung des Bildsignals durch eine einfache Quantisierung realisiert wird. Die *D*PCM* erweitert die PCM-Übertragungstrecke, in dem vom Eingangssignal $s(n)$ ein Prädiktionssignal $\hat{s}_1(n)$ subtrahiert wird. Mit dieser Differenzbildung wird die gewünschte Dekorrelation des Bildsignals vorgenommen. Am Decoder erzeugt ein möglichst gleicher Prädiktor das Prädiktionssignal $\hat{s}_2(n)$ wieder hinzu, das die vorhersagbaren Bildsignalanteile repräsentiert. Abb. 4.2-1 zeigt das Prinzipschaltbild der *D*PCM*.

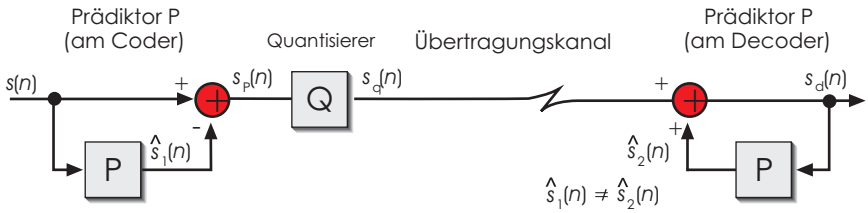


Abb. 4.2-1: Prinzipschaltbild der D*PCM

4.2.2 Übergang zur Feed-Backward DPCM

Man kann zeigen, dass die erreichbare Codiereffizienz des D*PCM-Konzeptes nicht viel größer ist, als die einer einfachen PCM. Problematisch kann jedoch der Quantisierungsfehler *q*(*n*) sein, da er sich nach und nach beim Decoder aufaddiert und zu einer Fehlerfortpflanzung führt. Der Grund liegt darin, dass Coder und Decoder nicht das gleiche Prädiktionssignal verwenden. Am Coder wird der Prädiktor mit *s*(*n*) gespeist. Am Decoder summiert sich stets der Quantisierungsfehler *q*(*n*) hinzu. Als Ergebnis können die beiden Prädiktoren mit ihren zeitlichen Vorhersagen innerhalb kürzester Zeit auseinander laufen. Daher muss man als Verbesserung den beiden Prädiktoren die gleichen Eingangssignale anbieten. Die dazu notwendigen Umbauschritte sind in Abb. 4.2-2 (a – c) dargestellt.

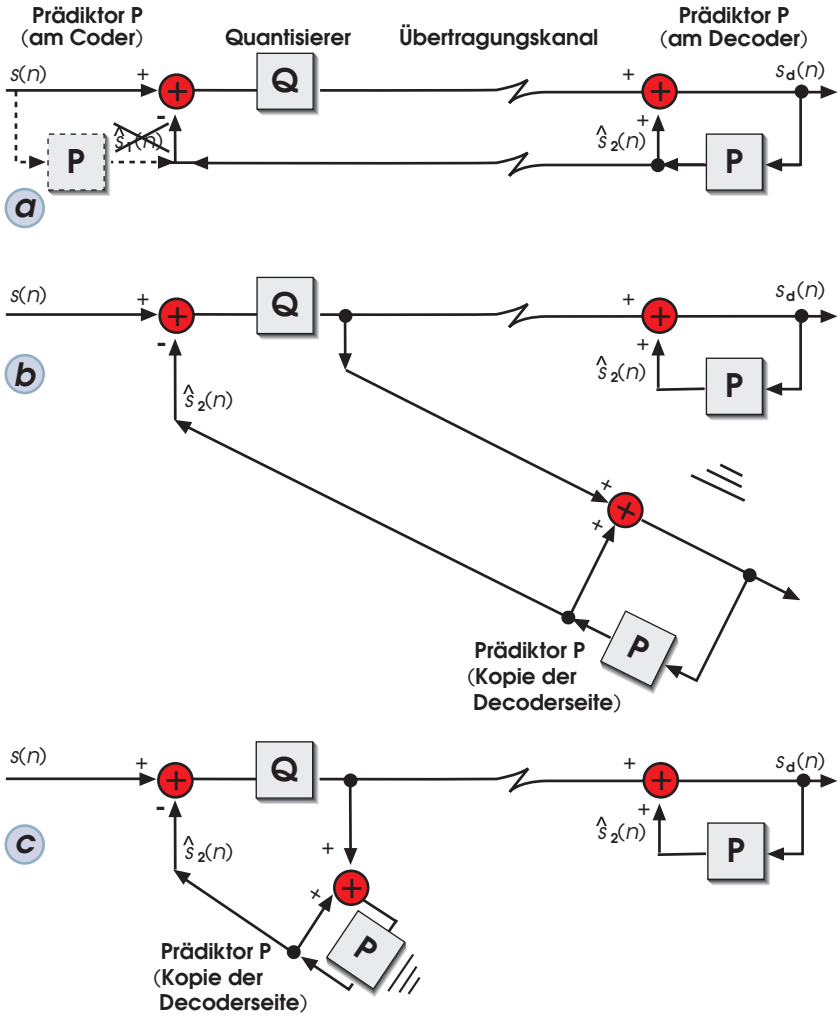


Abb. 4.2-2: Übergang von der D*PCM zur DPCM (nach [4.10])

Zunächst wird das Signal des Prädiktors am Decoder $\hat{s}_2(n)$ für die Differenzbildung zum Coder zurückgeführt (vgl. Abb. 4.2-2 (a)). Damit befindet sich der Quantisierer ebenfalls in der Schleife zur Berechnung des Differenzsignals auf der Coderseite. Im nächsten Schritt wird der Decoder dupliziert (Abb. 4.2-2) und in den Coder geklappt (Abb. 4.2-2), damit wieder nur ein Signal den Übertragungskanal passieren muss. Man erkennt, dass Coder und Decoder trotz Quantisierung das gleiche Prädiktionssignal $\hat{s}_2(n)$ erzeugen, da beide das gleiche Signal

$$s_d(n) = s(n) + q(n) \qquad 4.2\text{-}1$$

am Eingang des Prädiktors verwenden. Man muss allerdings berücksichtigen, dass im Unterschied zur D*PCM auch der Prädiktor auf der Coderseite ein brauchbares Prädiktorsignal aus dem quantisierten Signal erzeugen können muss. Man kann theoretisch zeigen, dass die DPCM für die spektrale Energieverteilung von natürlichen Bildvorlagen tatsächlich einen Vorteil bringt. Bei entsprechender Wahl des DPCM-Decoders kann zudem gewährleistet werden, dass alle Übertragungsfehler asymptotisch abklingen. Damit ist ein wichtiges Stabilitätskriterium erfüllt.

Vergleich D*PCM und DPCM

Abb. 4.2-3 zeigt das fertige Blockschaltbild des DPCM-Übertragungssystems, bei dem noch die Redundanzreduktion nach der Quantisierung hinzugefügt wurde.

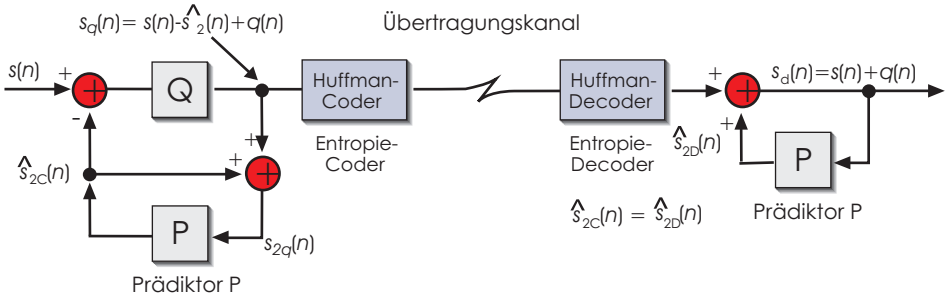


Abb. 4.2-3: Blockschaltbild der DPCM mit Redundanzreduktion

4.2.3 Feed-Backward DPCM mit Bildspeicher als Prädiktor

DPCM mit Bildspeicher

Für die digitale Bewegtbildcodierung stellt sich die Frage, wie der Prädiktor *P* ausgelegt werden muss, um die Dekorrelation für ein Folge beliebiger, photografischer Einzelbilder zu maximieren. Als ersten Ansatz für den Prädiktor kann man einen Bildspeicher einsetzen, der für geringe Bildveränderungen eine gute Prädiktion liefert.

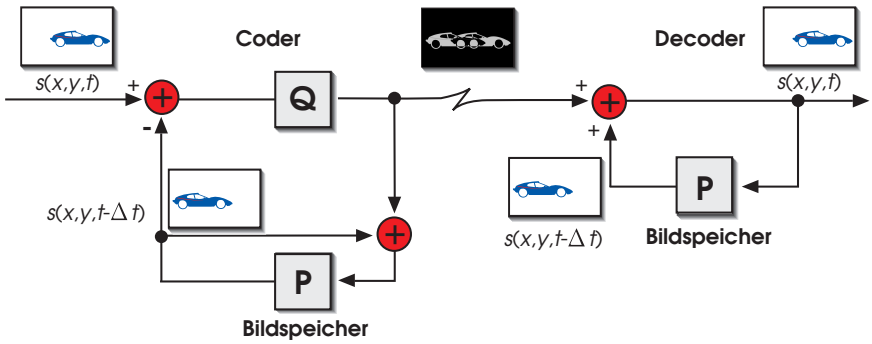


Abb. 4.2-4: DPCM mit Bildspeicher als Prädiktor

Die maximale Dekorrelation des Bildsignals wird erreicht, wenn zeitlich aufeinanderfolgende Einzelbilder oder zumindest große Bildbereiche ähnlich sind. Abb. 4.2-4 veranschaulicht den Sachverhalt an einem einfachen Beispiel.

Die Bildspeicher am Coder und Decoder halten jeweils das letzte Einzelbild (Zeitpunkt $t - \Delta t$) vor, um es mit dem aktuellen Bild zu dekorrelieren. Wenn die Quantisierung nicht zu grob eingestellt ist, ergibt sich eine recht kleine Differenzbildleistung, die effektiv mit einer Redundanzreduktion codiert und übertragen werden kann.

Selbsttestaufgabe 4.2-1:

Beschreiben Sie die wesentlichen Unterschiede zwischen Feed-Forward und Feed-Backward DPCM!

Selbsttestaufgabe 4.2-2:

Für welche Fälle eignet sich der Bildspeicher als Prädiktor in einem DPCM-System besonders gut? Beschreiben Sie eine einfache Beispielszene, bei der dieser Prädiktor besonders schlecht geeignet ist!

4.3 Bewegungsgesteuerte, prädiktive Bildcodierung

Aufeinanderfolgende Einzelbilder eines Videos unterscheiden sich in erster Linie durch die veränderten Bewegungsphasen der im Bild vorhandenen Objekte. Handelt es sich bei den bewegten Objekten um natürliche, massebehaftete Körper, so kann davon ausgegangen werden, dass sich die Bewegungsrichtung nicht spontan ändert, sondern über mehrere Einzelbilder annähernd konstant bleibt. Diese Eigenschaft führt zu der Idee, den in Abschnitt 4.2 vorgestellten Prädiktor zusätzlich mit einer Bewegungsvorhersage zu versehen.

4.3.1 Das Prinzip der Bewegungskompensation

In Abschnitt 4.2 wurde die *Interframe-DPCM* auf Basis eines einfachen Bildspeichers vorgestellt. Es wurde deutlich, dass die erzielbare Datenkompression entscheidend von der Vorhersagegüte des Prädiktors abhängt. Im Fall des einfachen Bildspeichers ist die Vorhersage nur für unveränderte Bildinhalte gut, bewegte Bildbereiche führen zu schlechter Prädiktion.

Bewegungskompensation
Verschiebungskompensation

Will man auch für die bewegten Bildreiche eine gute Prädiktion erzielen, müssen die Bewegungen dieser Bildbereiche für die aktuelle Bewegungsphase geeignet zurechtgerückt werden. Dieser Vorgang wird in der Bildcodierung „*Bewegungskompensation*“ genannt. Bildsequenzen entstehen durch die zeitliche Abtastung von zeitkontinuierlichen Bildsignalen. Die Bewegung wird dabei ebenfalls zeitlich diskretisiert. Daher ist der Begriff *Verschiebungskompensation* treffender. Da sich jedoch der Begriff *Bewegungskompensation* in der Literatur eingebürgert hat, werden nachfolgend beide Begriffe synonym verwendet. Um eine *Bewegungskompensation* oder *Verschiebungskompensation* durchführen zu können, ist eine örtliche Lokalisierung der bewegten Objekte und die Schätzung der Verschiebung erforderlich.

Bewegungskompensierte DPCM

Abb. 4.3-1 zeigt das Prinzip der bewegungskompensierten DPCM.

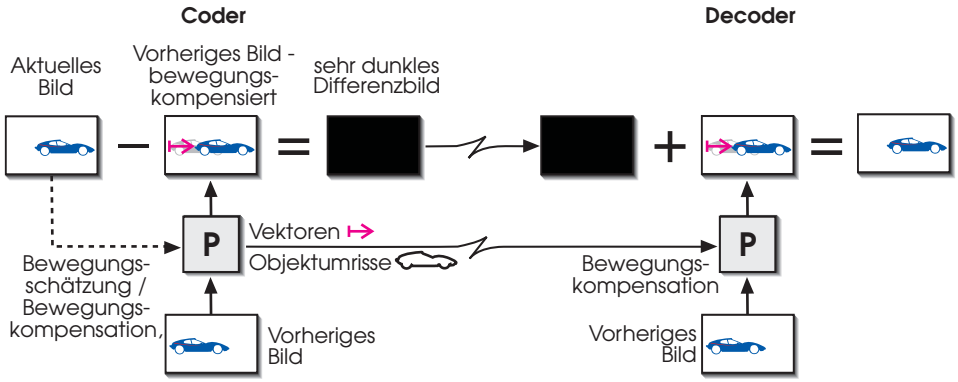


Abb. 4.3-1: Prinzip der DPCM ***mit*** Bewegungskompensation als Prädiktor

Obwohl beide Prädiktoren – wie bei der DPCM üblich – die gleichen Signale verarbeiten, ist die Komplexität zwischen Coder und Decoder unterschiedlich. Die für die Prädiktion erforderlichen *Bewegungsinformation* (*Verschiebungsvektoren* und die zugehörige *örtliche Objektzuordnung*) wird nämlich ausschließlich am Coder erzeugt und zusammen mit dem Differenzbild zum Decoder übertragen. Dieser kann verhältnismäßig einfach mit einem vorherigen Einzelbild und der Bewegungskompensation das aktuelle Einzelbild errechnen.

Die Abb. 4.3-2 und Abb. 4.3-3 verdeutlichen noch einmal die genaue Funktionsweise der beiden Prädiktoren im Coder und Decoder.

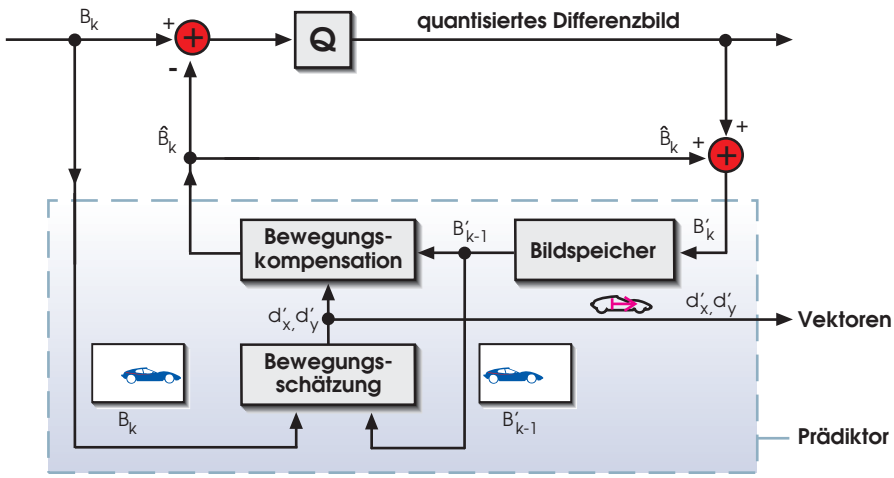


Abb. 4.3-2: Blockschaltbild DPCM mit Bewegungskompensation: Coder

Der Prädiktor am Coder (Abb. 4.3-2) beinhaltet die drei Blöcke *Bewegungsschätzung*, *Bildspeicher* und *Bewegungskompensation*. Die Bewegungsschätzung errechnet aus dem aktuellen Einzelbild B_K und dem im Bildspeicher vorgehaltenen und quantisierten Einzelbild der letzten Bewegungsphase B'_{K-1} die Objektverschiebung in *x*-Richtung d'_x und *y*-Richtung d'_y. Die Bewegungsinformation wird, wie bereits beschrieben, zusätzlich zum quantisierten Differenzbild übertragen. Weiterhin wird mit ihr die Bewegungskompensation am Coder versorgt, die aus dem Signal des Einzelbildes B'_{K-1} und der Verschiebungsvektoren das Signal $\hat{B}_K$ errechnet. Das

errechnete Signal $\hat{B}_K$ sollte eine möglichst gute Prädiktion des aktuellen Einzelbildes B_K sein. Alle Bildbereiche, die der Prädiktor nicht mit diesem Verschiebungsmodell vorhersagen kann, verbleiben als Anteile im Differenzbild. Bei der Errechnung und Codierung der Verschiebungsinformation muss darauf geachtet werden, dass die Codiereffizienz durch das zusätzliche Datenaufkommen nicht zu sehr verschlechtert wird.

Der Decoder nach Abb. 4.3-3 benötigt nur den Bildspeicher und die Bewegungskompensation. Da die Bewegungsinformation bei dieser Codierung mit im Datenstrom übertragen wird, ist aufwändige Ermittlung im Decoder nicht erforderlich. Mit Hilfe dieser Daten kann der Decoder aus dem Signal B'_{K-1} direkt die Prädiktion $\hat{B}_K$ errechnen.

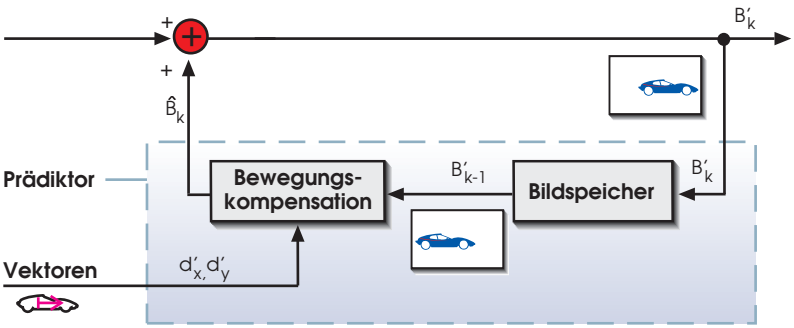


Abb. 4.3-3: Blockschaltbild DPCM mit Bewegungskompensation: Decoder

Wie leicht einzusehen ist, hängt die Qualität der Prädiktion von der Güte der Bewegungsschätzung ab. Im folgenden Abschnitt werden einige Verfahren vorgestellt, mit denen die Ermittlung von Verschiebungsvektoren prinzipiell möglich ist.

4.3.2 Verfahren zur Bewegungsschätzung

Die zur Messung von Verschiebungen entwickelten Verfahren können in drei Gruppen aufgeteilt werden:

- Differenzielle Verfahren,
- Phasenkorrelation,
- Matchingverfahren.

Dabei sind verschiedene Varianten möglich, die auf der *rekursiven* oder *hierarchischen* Anwendung der Algorithmen beruhen. Bei den Matchingverfahren können verschiedene Suchstrategien angewendet werden (vgl. Abschnitt 4.3.3).

Differenzielle Verfahren

Die *differenziellen Verfahren* basieren auf einer Methode, die 1976 von *Cafforio* und *Rocca* vorgeschlagen wurde (vgl. [4.15]). Der Algorithmus setzt auf die Beziehung zwischen örtlicher und zeitlicher Ableitung auf. Eine Verschiebung kann dabei durch den folgenden Ausdruck bestimmt werden:

$$\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} \frac{\int\limits_B (s_k(x,y) - s_{k-1}(x,y)) \cdot \frac{\partial}{\partial x} s_k(x,y) dB}{\int\limits_B \left(\frac{\partial}{\partial x} s_k(x,y)\right)^2 dB} \\ \frac{\int\limits_B (s_k(x,y) - s_{k-1}(x,y)) \cdot \frac{\partial}{\partial y} s_k(x,y) dB}{\int\limits_B \left(\frac{\partial}{\partial y} s_k(x,y)\right)^2 dB} \end{bmatrix}. \qquad 4.3\text{-}1$$

Dabei bezeichnet $s_k(x, y)$ das Bildsignal zum Zeitpunkt *kT* und *B* den betrachteten Teilbereich der Bildebene. Dieser Ansatz funktioniert allerdings nur, wenn für die Helligkeitsübergänge, z. B. an Kanten, ein linearer Verlauf angenommen wird[27]. Für kleine Verschiebungsmessungen funktionieren differenzielle Verfahren recht gut. In natürlichen Bildvorlagen lassen sich die zuvor genannten Einschränkungen jedoch oft nicht finden. Daher sind sie für die Betrachtung bei der Bildcodierung nur von geringem Interesse.

Phasenkorrelation

Die Methode der *Phasenkorrelation* verwendet den Umweg über den Frequenzbereich. Um die Verschiebung mit Hilfe der Phasenkorrelation zu bestimmen, werden zwei zeitlich aufeinanderfolgende Bildblöcken normiert, in den Frequenzbereich transformiert, multipliziert und das Produkt zurücktransformiert. Das Maximum der so berechneten Kreuzkorrelation liefert die gewünschte Verschiebung. Das Problem bei der Phasenkorrelation ist, dass die örtliche Zuordnung der gefundenen Verschiebung nicht exakt vorgenommen werden kann. Dies gilt insbesondere dann, wenn größere Bildbereiche für die Phasenkorrelation herangezogen werden, was zumindest zu einer höheren Betragsgenauigkeit des Verschiebungsvektors führt. Daher kann die Phasenkorrelation zwar für die Messung von Globalverschiebungen (z. B. Kameraschwenk) gut verwendet werden, nicht jedoch für eine Verschiebungsmessung von kleinen Objekten.

Matchingverfahren

In diesem Punkt sind die so genannten *Matchingverfahren* im Vorteil. Bei ihnen wird korrespondierender Bildinhalt in aufeinanderfolgenden Bildern gesucht. Als Maß für die Ähnlichkeit zweier aufeinanderfolgender Einzelbilder bzw. Bildausschnitte wird allgemein die so genannte *Displaced Frame Difference* (*DFD*) an einer Position (*x*, *y*) und für eine Verschiebung (d_x, d_y) definiert:

Displaced Frame Difference – DFD

$$DFD(x, y, d_x, d_y) = s_k(x, y) - s_{k-1}(x - d_x, y - d_y). \qquad 4.3\text{-}2$$

Mean Square Error – MSE

In der Praxis werden diese Matchingverfahren im Ortsbereich auf $m \cdot n$ große Blöcke

27 Diese im allgemeinen nicht gerechtfertigte Annahme führt zu einem Schätzfehler, weshalb oft ein rekursives Vorgehen vorgeschlagen wird, das „pelrekursive differentielle Schätzung" genannt wird.

angewendet. Dazu wird mit Hilfe der DFD ein mittlerer quadratischer Fehler (*Mean Square Error - MSE*) in Bezug auf den Bildblock (*Blockmatching*) berechnet:

$$MSE(d_x, d_y) = \frac{1}{m \cdot n} \int_{-\frac{m}{2}}^{\frac{m}{2}} \int_{-\frac{n}{2}}^{\frac{n}{2}} DFD^2(x, y, d_x, d_y) dx\, dy. \qquad 4.3\text{-}3$$

Mean Absolute Difference – MAD

Alternativ kann auch die so genannte *Mean Absolute Difference* (*MAD*) als Zielfunktion herangezogen werden:

$$MAD(d_x, d_y) = \frac{1}{m \cdot n} \int_{-\frac{m}{2}}^{\frac{m}{2}} \int_{-\frac{n}{2}}^{\frac{n}{2}} |DFD(x, y, d_x, d_y)| dx\, dy. \qquad 4.3\text{-}4$$

Es lässt sich allerdings zeigen, dass die MAD nur zu suboptimalen Ergebnissen führt (vgl. [4.17]). Daher wird sie nur in Hardwarelösungen eingesetzt, die aufgrund der Randbedingungen nur einen geringen Aufwand erlauben.

4.3.3 Der Blockmatching Algorithmus im Detail

Die korrekte Messung einer Verschiebung mit dem Blockmatching Verfahren ist so lange möglich, wie sich die Helligkeit eines Objektes zwischen zwei Einzelbildern nicht ändert. Diese Annahme kann nicht immer erfüllt werden, da *Ein-*, *Aus-* und *Überblendungen*, *Beleuchtungsänderungen* (Licht und Schatten) und verändernde Oberflächenreflexionen eine Verschiebungsmessung mit Hilfe des Blockmatching Verfahrens erschweren.

Prinzip Blockmatching

Abb. 4.3-4 veranschaulicht das Blockmatching an einem einfachen Beispiel.

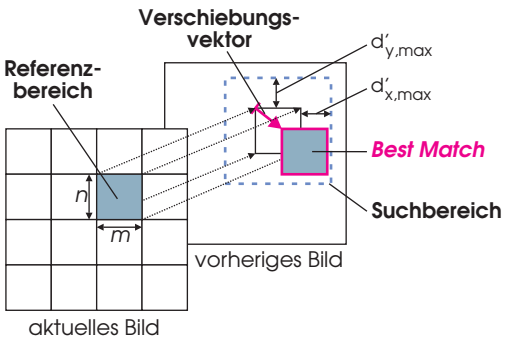


Abb. 4.3-4: Blockmatching Verfahren: Festlegung von Referenz- und Suchbereich

Es werden folgende Schritte durchgeführt:

1. Das aktuelle Bild wird in gleichmäßige, gleichgroße Blöcke der Größe *m* x *n* zerlegt. Für eine einfache Berechnung wird oft der Fall *m* = *n* (quadratische Blöcke) angenommen.

Referenzblock / Suchbereich

2. Für jeden Block (*Referenzblock*) des aktuellen Bildes werden mehrere Verschiebungstests (*Matches*) im vorherigen Bild vorgenommen. Dazu wird der *MSE* bzw. *MAD* für jede mögliche Verschiebung des Referenzblockes innerhalb eines definierten *Suchbereich* im vorherigen Bild berechnet. Aufgrund der örtlich diskreten Darstellung wird nur eine beschränkte Anzahl von Verschiebungen an diskreten Positionen getestet und die zugehörige *MSE* bzw. *MAD* ermittelt. Für die Berechnung der *diskreten MSE* ergibt sich

$$MSE_{\perp}(d_x, d_y) = \frac{1}{m \cdot n} \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} \left(s_k(x,y) - s_{k-1}(x-i-d_x, y-j-d_y)\right)^2. \qquad 4.3\text{-}5$$

Best Match

3. Als *Best Match* wird die Verschiebung bezeichnet, die innerhalb des Suchbereiches eine minimale *MSE* bzw. *MAD* erzeugt. Die Verschiebungswerte d'_x und d'_y sind gefunden.

Bei der Anwendung des Blockmatching Verfahrens sind einige Randbedingungen zu beachten. So muss beispielsweise die Größe des Referenzblockes gut überlegt werden. Kleine Blöcke führen in der Regel zwar zu einer Zielfunktion mit kleinen Werten. Es besteht aber auch die Gefahr von Mehrdeutigkeiten bei der Suche nach dem absoluten Minimum. Extremfall: Ein einzelnes Pixel lässt sich möglicherweise an vielen Positionen im Bild mit gleicher *MSE*[28] wiederfinden (konstante Helligkeit). Insgesamt führen kleine Referenzblöcke zu einer großen Anzahl von Bewegungsvektoren (viele möglicherweise sehr ähnlich), die codiert und übertragen werden müssen. Eine Steigerung der Codiereffizienz ist damit erschwert. Werden auf der anderen Seite die Referenzblöcke zu groß gewählt, ist das Auffinden des örtlich exakten Minimums ebenfalls problematisch, da das so genannte *MSE-Gebirge*[29] relativ flach wird und nicht mehr ausreichend charakteristische Minima besitzt. Zudem bleiben die *MSE*- bzw. *MAD*-Werte recht groß, da die Korrelation zwischen Referenzblock und den tatsächlich im Bild bewegten Objekten gering wird.

Überlappung von Blöcken

Das zuvor skizzierte Verfahren führt dazu, dass sich Referenzblöcke bei der Suche nach dem minimalen *MSE* im vorherigen Bild im Ergebnis überlappen. Abb. 4.3-5 verdeutlicht dies.

28 Für diesen Spezialfall geht die *MSE* in die *DFD* über.

29 3D-Darstellung der Zielfunktionswerte über *x*- und *y-Achse*.

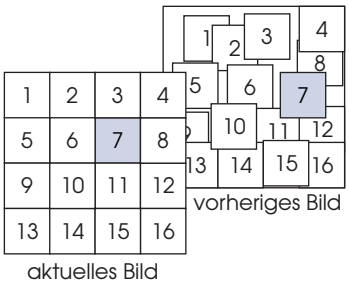


Abb. 4.3-5: Mögliche Überlappung von Blöcken beim Blockmatching Verfahren

Aufgrund von Auf- und Verdeckungseffekten ist eine optimale Zuordnung von Referenzblöcken im Suchbereich des vorherigen Bildes möglicherweise nicht möglich. Als Konsequenz hieraus muss festgehalten werden, dass eine exakte Messung der Verschiebung manchmal gar nicht möglich ist. Oft ergeben sich an Hand des *MSE-Gebirges* mehrere Minima, die nicht zwangsläufig mit der tatsächlichen Bewegung korrelieren müssen. Abb. 4.3-6 zeigt zwei Beispiele für mögliche *MSE-Gebirge.*

MSE-Gebirge

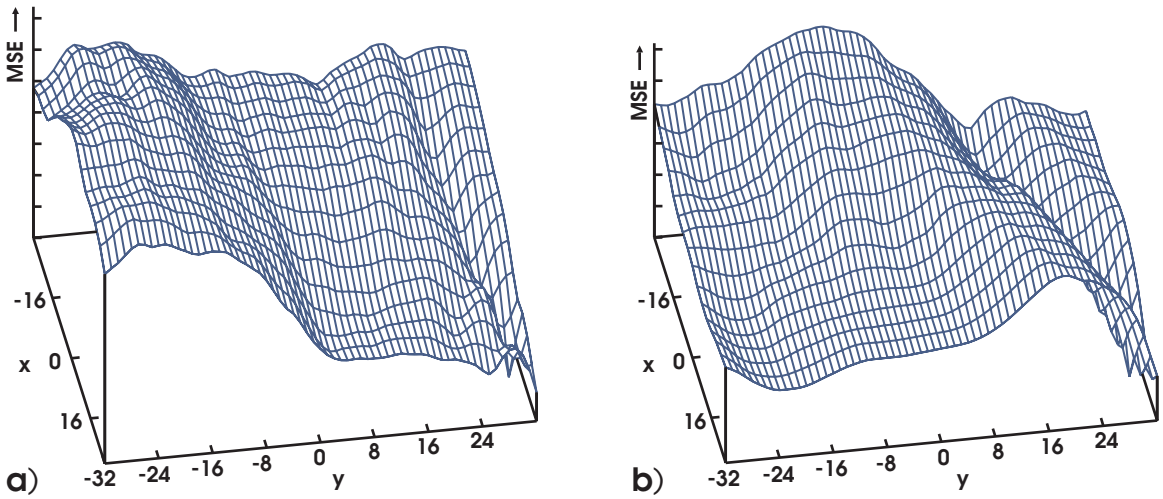


Abb. 4.3-6: *MSE-Gebirge* a) ausgeprägter Abfall zu einer Seite hin b) Abfall zu zwei Seiten hin

Abb. 4.3-6 macht deutlich, dass das ermittelte Minimum möglicherweise direkt an der Grenze des Suchbereiches liegt, oder mehrere gleichwertige Minima im Suchbereich gefunden werden können. Darüber kann man erahnen, dass die Anzahl der *MSE*-Berechnungen N_{FS} für die Ermittlung des Minimums (d_x, d_y) sehr groß werden kann, insbesondere wenn große Such- und Referenzbereiche vorgegeben sind. Allgemein errechnet sich N_{FS} (*Full-Search*) zu:

Full-Search

$$N_{FS} = (2\,d_{x,\max} + 1) \cdot (2\,d_{y,\max} + 1) \qquad 4.3\text{-}6$$

Für den Fall $d_{x,\max} = d_{y,\max}$ hängt der Rechenaufwand quadratisch von der maximalen Verschiebung ab.

4.3.4 Minimierung des Rechenaufwandes beim Blockmatching Algorithmus

Der Rechenaufwand beim Blockmatching kann verringert werden, wenn statt der vollständigen Suche (*Full-Search*) nur eine bestimmte Anzahl von Testpunkten zur Ermittlung der minimalen *MSE* im Suchbereich herangezogen werden. Da nicht alle Positionen abgesucht werden, besteht bei so einem Verfahren natürlich die Gefahr, dass nur ein lokales Minimum gefunden wird, und damit nicht die optimale Codiereffizienz erzielt werden kann. In vielen Fällen sind die Ergebnisse aber fast gleich gut, obwohl nur etwa 5% bis 10% des Rechenaufwandes getrieben werden muss. Es kommt in der Praxis also sehr darauf an, einen guten Kompromiss zwischen Aufwand (*Rechenzeit*) und Nutzen (*Codiereffizienz*) zu finden.

Logarithmische Suche

Nachfolgend werden drei Beispiele für solche rechenoptimierten Suchalgorithmen vorgestellt, die ihre Schrittweiten *logarithmisch* in Abhängigkeit der besten Suchrichtung verfeinern. Zunächst wird der *MSE* im Zentrum (keine Verschiebung) errechnet. Für die nächste Berechnung wird eine Schrittweite *p* gewählt, die anfangs den Wert

$$p = \frac{d_{xy,\max}}{2} \qquad \text{4.3-7}$$

besitzt. Es folgen Berechnungen des *MSE*, die je nach Verfahren horizontal, vertikal und / oder diagonal um die Schrittweite *p* vom Ausgangspunkt verschoben sind. Das Minimum dieses *MSE*-Wertes bestimmt den Startpunkt für die weitere Berechnung des Verschiebungsvektors. In der nächsten Iterationsstufe wird die Schrittweite *p* wieder halbiert, so dass der Verschiebungsvektor immer mehr angenähert wird. Ist die gewünschte Auflösung erreicht, wird die Iteration abgebrochen.

In Abhängigkeit der maximalen Suchweite $d_{x,\max} = d_{y,\max} = d_{max}$ ergeben sich für diese Algorithmen die Anzahl der Suchpunkte zu

$$\begin{aligned} N_{TSS} &= 1 + 8 \cdot \log_2(d_{\max}) \\ N_{TDL} &= N_{CSA} = 5 + 4 \cdot \log_2(d_{\max}). \end{aligned} \qquad \text{4.3-8}$$

Three-Step Search, 2D Logarithmic Search, Cross-Search Algorithms

Abb. 4.3-7 visualisiert die Suchpunkte für den *Three-Step Search* (*TSS*), den *2D Logarithmic Search* (*TDL*) und den (*Greek*) *Cross-Search Algorithms* (*CSA*) für $d_{\max}$ = 8.

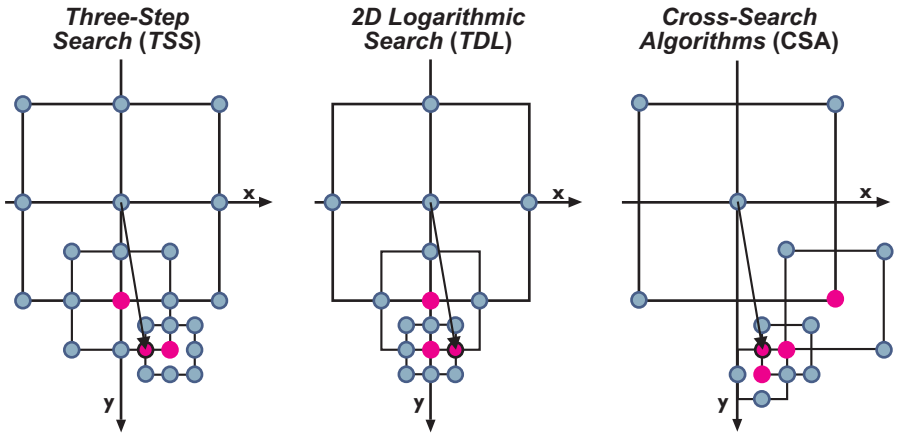


Abb. 4.3-7: Blockmatching mit logarithmischer Suchstrategie

4.3.5 Rekursiv hierarchisches Blockmatching

Subpixel-Matching

In der Praxis zeigt sich, dass die Genauigkeit der vorgestellten Blockmatching Verfahren in vielen Fällen nicht ausreicht. Wünschenswert ist oft eine Verschiebungsmessung, die auch Verschiebungen mit der Genauigkeit unterhalb eines Pixels erfassen kann. Man spricht hier auch von *Subpixel-Genauigkeit*. Die Genauigkeit kann bei vielen Bildsequenzen weit unter die Pixelgrenze gedrückt werden, wenn man rekursiv die Verschiebungsmessung in unterschiedlich fein abgetasteten Bildbereichen vornimmt. Die Subpixel-Genauigkeit wird erreicht, wenn die letzte Stufe der Verschiebungsmessung in einem örtlich überabgetasteten Bildbereich durchgeführt wird. Abb. 4.3-8 zeigt so ein *rekursiv hierarchisches Blockmatching*.

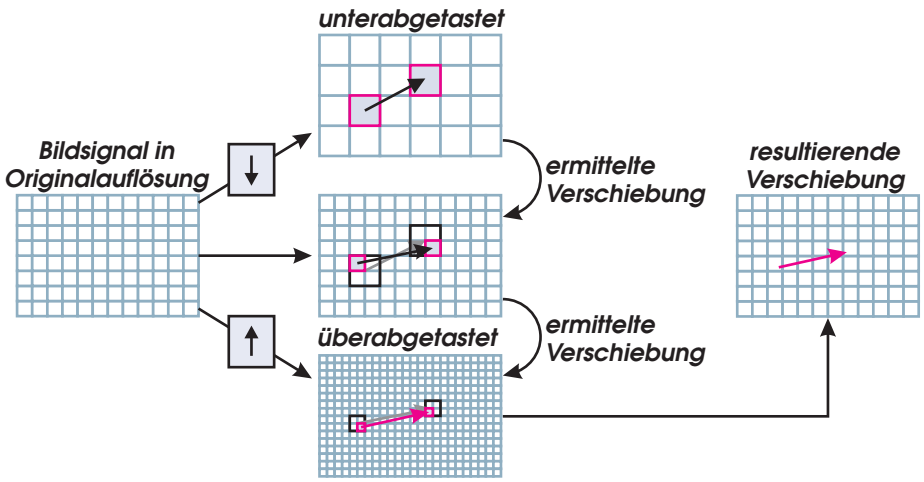


Abb. 4.3-8: Prinzip des rekursiv hierarchischen Blockmatching

4.3.6 Anwendung des Blockmatching Algorithmus mit der DPCM

Blockmatching – Minimierung der Differenzbildleistung

In Abb. 4.3-2 wurde ein Coder für ein Codierverfahren vorgestellt, bei der der Prädiktor an Hand von Verschiebungsinformation eine Prädiktion für die DPCM vornimmt. Der Blockmatching Algorithmus ist für diese Anwendung besonders gut geeignet, da er im Suchverfahren genau die Differenzbildung von aufeinanderfolgenden Bildbereichen nutzt. Wie bereits erwähnt ist der hier vorgestellte Blockmatching Algorithmus im Zusammenhang mit der DPCM gar nicht zur exakten Messung der Bewegung in den einzelnen Bildbereichen geeignet. Vielmehr wird die *Minimierung des Prädiktionsfehlers* und der *Differenzbildleistung* angestrebt. Diese Minimierung führt letztlich zu einer Reduktion der erforderlichen Datenrate für das quantisierte Differenzbild. Die hier gemachte Unterscheidung zwischen der *Minimierung der Differenzbildleistung* und einer *tatsächlichen Verschiebungsmessung* ist wichtig, um zu verstehen, dass es mit einem normalen Blockmatching Verfahren nicht möglich ist, den exakten Bewegungsverlauf von Objekten zu bestimmen und so die verschiedenen bewegten Objekte zu trennen, wie es beispielsweise für eine objektbasierte Bewegtbildcodierung nach MPEG-4 (vgl. Kapitel 5) wünschenswert wäre.

4.3.7 Hybride DCT als Kombination aus DPCM und DCT

Hybride DCT

In Kapitel 3 wurde gezeigt, dass der *DCT*-Algorithmus besonders für die *Intraframe-Codierung* von Einzelbildern geeignet ist. Die Diskussion zum *DPCM*-Verfahren zeigte, dass dieser Algorithmus besonders für die *Interframe-Codierung* in Bewegtbildsequenzen vorteilhaft eingesetzt werden kann. Es stellt sich nun die Frage, ob sich die Effizienz der Datenreduktion noch dadurch weiter erhöhen lässt, indem man beide Verfahren kombiniert und in einem Encoder-Decoder-System integriert. Ein solches System wird *hybride DCT* genannt und bildet die Grundlage für alle modernen Codierverfahren der Bildcodierung. So basiert z. B. auch der *MPEG-Standard* (vgl. Kapitel 5) auf einer hybriden DCT.

Ohne Zweifel lassen sich mit der *hybriden DCT* bei Bewegtbildsequenzen höhere Kompressionsfaktoren erreichen als mit der *Intraframe-DCT* oder der *Interframe-DPCM* alleine. Der besondere Ansatz bei der hybriden DCT besteht darin, dass nicht das Bildsignal, sondern das bewegungskompensierte Differenzbildsignal transformationscodiert wird.

Hybride DCT-Encoder

Abb. 4.3-9 zeigt das relativ komplexe Blockschaltbild des Encoders der *hybriden DCT*.

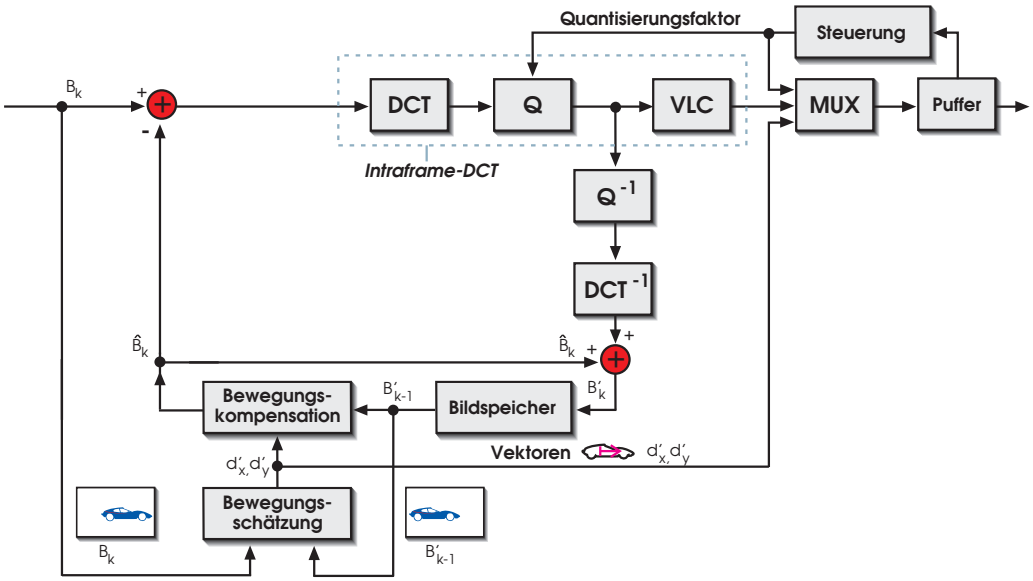


Abb. 4.3-9: Vereinfachtes Blockschaltbild der *Hybriden DCT*-Encoder

Deutlich zu erkennen sind die bekannten Komponenten der *Interframe-DPCM*, die durch die *Intraframe-DCT* (gestrichelt umrandet) ergänzt wird. In der DPCM-Schleife sind die inverse Quantisierung (Q^{-1}) und die inverse DCT (DCT^{-1}) hinzugefügt worden, damit der Quantisierungsfehler im Bildbereich mit dem bewegungskompensierten Bild $\hat{B}_K$ zusammengefügt werden kann.

Neu ist auch der Multiplexer (*MUX*), der das DCT-transformierte, bewegungskompensierte und quantisierte Differenzbild mit der zugehörigen Bewegungsinformation und der aktuellen Quantisierungstabelle verknüpft. Anders als bei der DV-Codierung (Abschnitt 4.1.2) erfolgt die Steuerung der Quantisierung über eine Feed-Backward Steuerung. Die Puffersteuerung am Ausgang des Coders sorgt dafür, dass im Mittel eine gleichmäßige Datenrate am Ausgang abgenommen werden kann.

Hybride DCT-Decoder

Abb. 4.3-10 zeigt das relativ einfache Blockschaltbild des *hybriden Decoders*. Der Eingangspuffer dient dazu, ein variables Datenaufkommen abfangen zu können. Der Demultiplexer (*DeMUX*) trennt Bilddaten, Quantisierungstabelle und Bewegungsinformation voneinander. Die übrigen Blöcke sind von der DCT- und DPCM-Decodierung her bekannt.

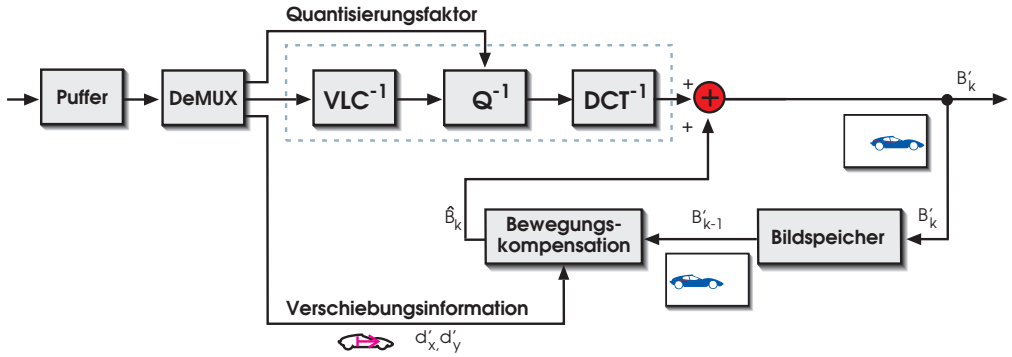


Abb. 4.3-10: Vereinfachtes Blockschaltbild der *Hybriden DCT*-Decoder

Selbsttestaufgabe 4.3-1:

Geben Sie die Formel für die diskrete MAD an!

Selbsttestaufgabe 4.3-2:

Warum berechnet ein Blockmatching Verfahren nicht in jedem Fall die tatsächliche Verschiebung von Objekten?

Selbsttestaufgabe 4.3-3:

Berechnen Sie die Anzahl der Suchpunkte für $d_{\max}$ = 8 für die Verfahren Full Search, Three-Step Search, Cross-Search Algorithms und 2D Logarithmic Search!

5 Standards der digitalen Videocodierung

Aus den in Kapitel 4 vorgestellten Systemkonzepten zur digitalen Videocodierung haben sich eine Reihe von internationalen Standards der *International Organisation for Standardisation* (*ISO*), *International Electrotechnical Commission* (*IEC*) und der *International Telecomunication Union* (*ITU*) entwickelt. Nachfolgend wird zunächst ein kurzer Überblick über die wichtigsten dieser Standards gegeben, bevor im Anschluss auf Details eingegangen wird.

Entwicklung Videocodierstandards

Abb. 5-1 veranschaulicht grob die Entwicklung einiger wichtiger Standards für unterschiedliche Anwendungsbereiche im Zeitraum von 1988 bis 2004.

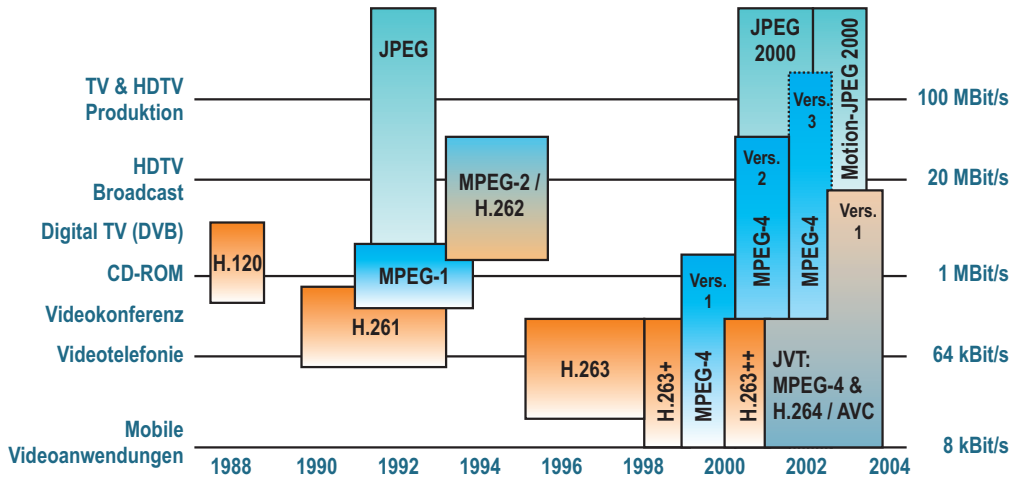


Abb. 5-1: Entwicklung der Videocodierstandards in der *ISO* und *ITU*:
orange: *ITU* (vormals *CCITT*)
hell-blau und cyan: *ISO/IEC* bzw. *ITU*

Es ist zu erkennen, dass schon vor der Entwicklung des Intraframe basierten JPEG-Verfahrens für die Einzelbildcodierung (*ITU-T T.81* bzw. *ISO/IEC 10918*; vgl. Kapitel 3) einige Interframe-Codierer für die Bewegtbildcodierung in der *ITU* definiert wurden. Zielanwendungen waren hier im wesentlichen die Bildtelefonie mit der schmalbandigen Übertragung von Bewegtbildern über ISDN-Kommunikationsleitungen (z. B. *ITU-T H.261*). Die hieraus entwickelten Weiterentwicklungen *MPEG-1* (*ISO/IEC 11172*) und *MPEG-2* (*ISO/IEC 13818*) sind für die Distribution von Bewegtbildvideo auf CD (*MPEG-1*) bzw. *DVD*[30] und für das digitale Fernsehen *DVB*[31] (*MPEG-2*) besonders geeignet. Die besondere Bedeutung dieser Anwendungen hat dafür gesorgt, dass sich *MPEG-1* und *MPEG-2* hier fest etabliert haben. Neuere und effizientere Codiersysteme, wie *ITU-T H.263* oder *MPEG-4 (ISO/IEC 14496*), setzen sich daher zunächst einmal nur in neuen

30 *DVD – Digital Versatile Disc*

31 *DVB – Digital Video Broadcasting*

Anwendungsbereichen, beispielsweise in der mobilen Bildkommunikation (Stichwort *UMTS*) oder in Systemen mit kleinen Speichermedien (*Media-Cards*) durch.

Viele der wichtigen Standards sind in identischer Form in beiden Normierungsgremien festgelegt worden. Neuere Standards, wie beispielsweise *ITU-T H.264/AVC* (identisch mit *ISO/IEC 14496-10 - MPEG-4, Part 10*) werden von so genannten *Joint-Meeting-Teams* ausgearbeitet (vgl. Abschnitt 7.3).

In den hier skizzierten 15 Jahren der Entwicklung von digitalen Videocodierverfahren hat die Codiereffizienz von Systemgeneration zu Systemgeneration erheblich zugenommen. Dieser Fortschritt wurde zwar erkauft mit einem ebenfalls deutlichen Anstieg bei der Komplexität der Codiersysteme. Auf der anderen Seite begünstigt immer schnellere und preiswertere Hardware die Beherrschbarkeit von komplexen Bildverarbeitungsalgorithmen bei der zügigen Einführung neuer Codiersysteme. Dieses führt dazu, dass einige Codierverfahren, deren Entwicklung mehr als 10 Jahre zurückliegt, nach und nach an Bedeutung verlieren werden.

5.1 Einfache Videocodierkonzepte der ITU-T – H.120 und H.261

Die beiden ersten standardisierten Videocodierkonzepte, die *Intraframe*- und *Interframe*-Verfahren miteinander kombinieren, sind die *ITU-Recommendations*[32] *H.120* und *H.261*. Die Systeme ermöglichen dabei in der Regel folgende Codieroptionen auf mehr oder weniger großen Teilbereichen (meist Makroblöcke) des Bildes:

Allgemeine Codieroptionen

1. Intraframe-Codierung,
2. Skip-Codierung,
3. Interframe-Codierung.

Diese beiden Standards sind von ihrer Leistungsfähigkeit heutzutage von Nachfolgestandards in vielen Anwendungsbereichen abgelöst worden.

5.1.1 H.120

ITU-T H.120

Die Empfehlung *ITU-T H.120*[5.1] stammt in seiner ersten Version aus dem Jahre 1984. Sie beschreibt den ersten internationalen Standard für die digitale Videocodierung. Die angestrebten Datenraten liegen zwischen 1544 kbit/s und 2048 kbit/s. Dieser Codec ermöglicht in seiner ersten Version nur eine Umschaltung zwischen der *Skip*- und der *Intraframe-Codierung*. Diese einfachste Methode zur Ausnutzung von Bild-zu-Bild Abhängigkeiten wird *Conditional Replenishment* (*CR*) genannt. Dazu wird eine Steuerinformation für jeden Bildblock generiert, die dem Decoder signalisiert, ob der Inhalt des letzten Bildblockes beibehalten werden soll (*Skip-Codierung*), weil sich der Bildinhalt nicht oder nur wenig geändert hat, oder

32 Das Standardisierungsgremium hieß damals noch *Consultative Committee for International Telegraph and Telephone* (*CCITT*).

ob eine herkömmliche *Intraframe-Codierung* stattfinden soll. Bei der *Intraframe-Codierung* selbst wird das *DPCM-Verfahren* und nicht eine Transformationscodierung (vgl. Kapitel 4) verwendet.

Prädiktionsfehlersignal

In Videosequenzen mit wenig Bewegung kann dieses Verfahren recht erfolgreich angewendet werden. Allerdings kann das *Prädiktionsfehlersignal*, verstanden als Differenz aus Original und Prädiktion, nicht weiter reduziert werden. Eine verbesserte Version des Standards aus dem Jahr 1988 codiert das *Prädiktionsfehlersignal* wiederum als Bildsignal. Eingeführt wurde die Bewegungskompensation und eine Hintergrundprädiktion, bei der die Schätzung des Szenenhintergrunds eine Verbesserung bringt, wenn eine Vordergrundbewegung den Hintergrund aufdeckt.

Insgesamt ist aber der Standard H.120 von recht geringer praktischer Bedeutung geblieben.

5.1.2 H.261

Blockschaltbild ITU-T H.261

Sehr viel erfolgreicher als *ITU-T H.120* konnte sich der Standard *ITU-T H.261*[5.2] in der Bildkommunikation (Bildtelefonie mit ISDN-Bandbreite bzw. n-facher ISDN-Bandbreite) durchsetzen. Der Encoder ist so ausgelegt, dass für den Videodatenstrom alleine Datenraten zwischen 64 kbit/s und 1920 kbit/s vorgesehen sind.

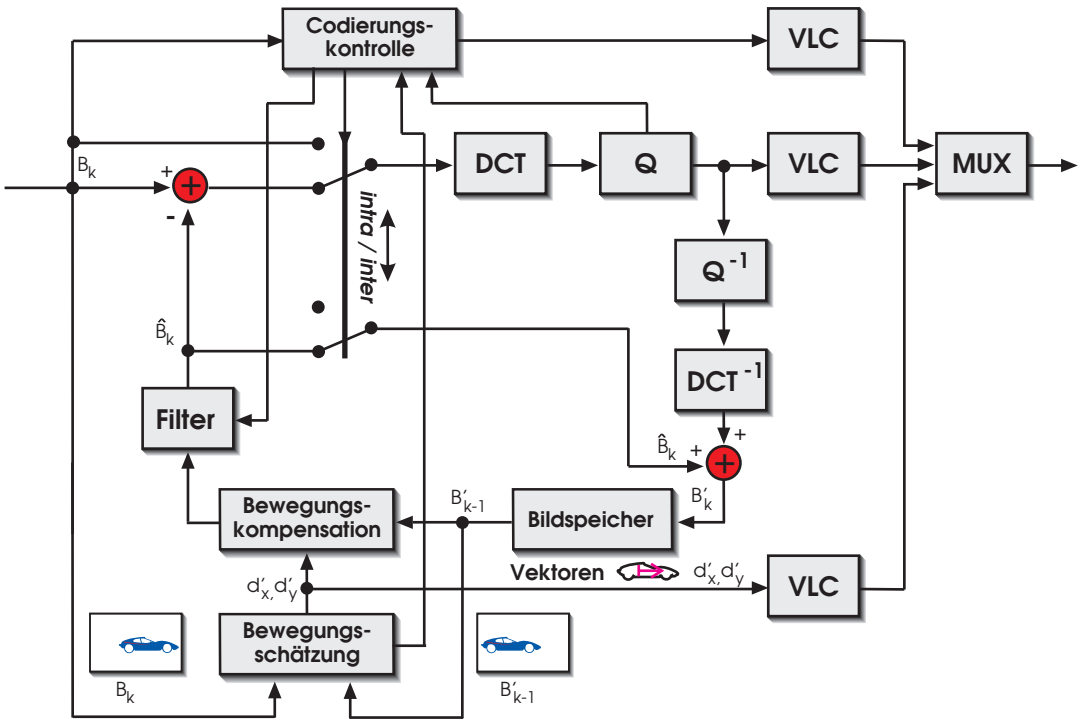


Abb. 5.1-1: Funktionsschaltbild des Codierers nach *ITU-T H.261* ohne Berücksichtigung des Ausgangspuffers zur Regelung der Datenrate

H.261 Codiereigenschaften

Der Standard wurde 1991 von der *ITU* in einer ersten Version verabschiedet und

verarbeitet *QCIF*- und *CIF*-Auflösungen[33] bei Bildfolgefrequenzen zwischen etwa 8 1/3 und 30 Hz. *ITU-T H.261* arbeitet als *Hybrider DCT-Coder*, wie er im Prinzip in Abb. 4.3-9 eingeführt wurde. Abb. 5.1-1 zeigt das hier verwendete Blockschaltbild. Die Bildaufteilung geschieht in Makroblöcken von 16 x 16 Bildpunkten (Luminanz) und 8 x 8 Bildpunkten (Chrominanz). Die Farbunterabtastung ist damit 4:2:0 (Abb. 2.2-1). Jeder Makroblock kann *Intraframe*-, *Skip*- oder *Interframe*-codiert werden. Die Genauigkeit der verwendeten Bewegungsvektoren für die Bewegungskompensation liegt je horizontal und vertikal bei einem Pixel (maximal +/- 16 Pixel). Die variable Lauflängencodierung (*VLC*) der mit Zick-Zack-Zusammenfassung gewonnenen und linear quantisierten Koeffizienten geschieht auf der Basis jedes einzelnen DCT-Blockes der Größe 8 x 8 Pixel. Die Bewegungsparameter werden in einer Differenzcodierung verlustfrei übertragen.

Das Tiefpassfilter (*Schleifenfilter*) hinter der Bewegungskompensation gleicht in gewissen Grenzen den Mangel einer Pixel-genauen Bewegungskompensation aus und kann so die Codiereffizienz eventuell erhöhen. Gesteuert wird das Filter durch die zentrale Codierungskontrolle, welches aus den Daten der Bewegungsmessung und der Quantisierung gespeist wird.

Eine konstante Bitrate wird bei diesem Verfahren durch Zwischenpufferung und Adaption der Quantisierungsstufenhöhe erreicht. Der Bitstrom folgt einer *hierarchischen Aufteilung* in mehreren Ebenen, die in Abb. 5.1-2 dargestellt sind.

H.261 Hierarchieebenen

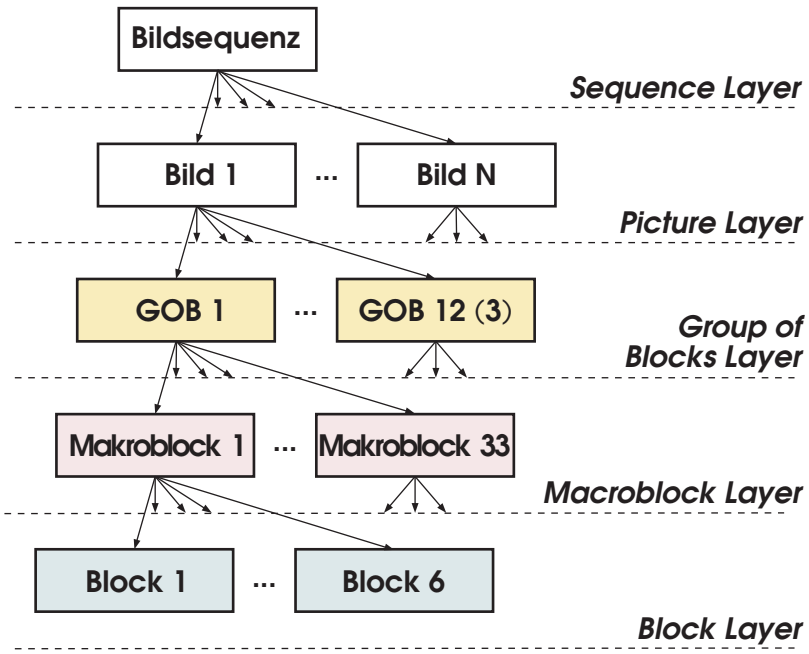


Abb. 5.1-2: Hierarchie des Codierkonzepts (links) nach *ITU-T H.261*

33 *CIF*: 352 x 288 Pixel bzw. 384 x 288 Pixel (1/4 PAL) bzw. 320 x 240 Pixel (1/4 NTSC) *QCIF*: *Quarter-CIF* – 1/4 Auflösung von *CIF*

Im *Picture Layer* werden die Einzelbilder einer Bildsequenz je nach Bildauflösung in 12 bzw. 3 Blockgruppen, *Group Of Blocks – GOB* zerlegt (CIF bzw. QCIF-Format, siehe Abb. 5.1-3). Jede GOB belegt somit eine Fläche von 176 x 48 Bildpunkten im Luminanzbild. Jede GOB des *Group of Block Layers* wird in 33 Makroblöcke unterteilt. Die Größe eines Makroblockes ist wie bei beim DV-Verfahren (vgl. Kapitel 4) 16 x 16 Pixel bezogen auf das Luminanzbild. Ein einzelner Makroblock (*Makroblock-Layer*) beinhaltet gemäß der 4:2:0-Farbbildfeldzerlegung 4 Luminanz- und 2 Chrominanzblöcke mit je 8 x 8 Bildpunkten. Jeder Block (*Block-Layer*) besteht schließlich aus den 64 DCT-Transformationskoeffizienten.

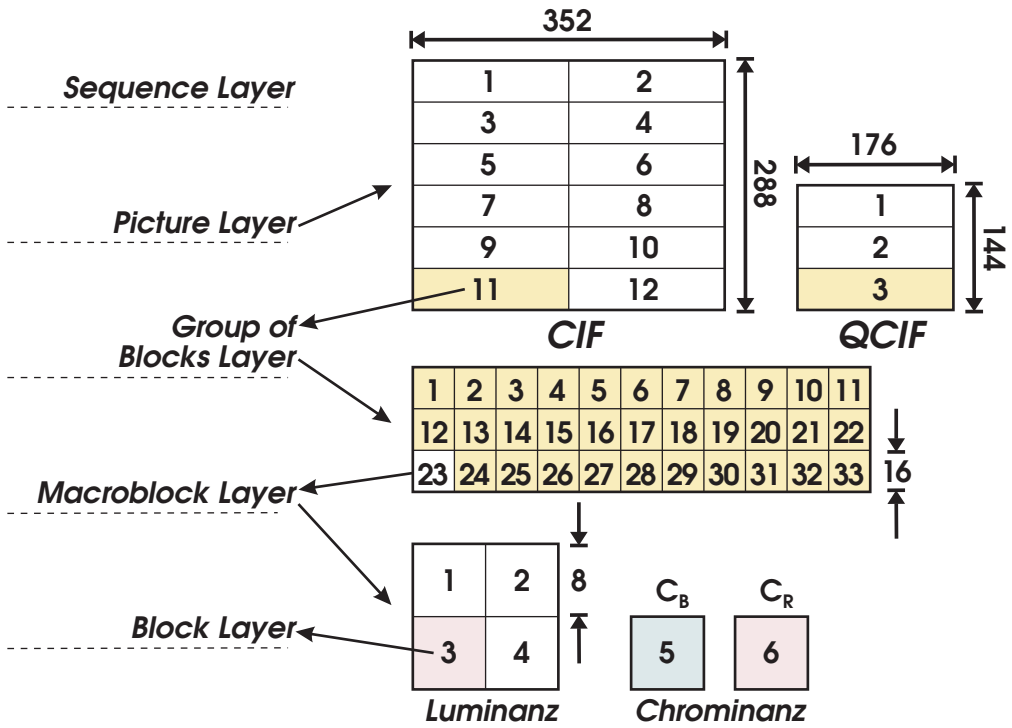


Abb. 5.1-3: Hierarchische Aufteilung des Bildes in *GOBs*, Makroblöcke und 4:2:0-Farbunterabtastung nach *ITU-T H.261*

Schleifenfilter

Auf der Makroblock-Ebene wird nach verschiedenen Makroblock-Typen unterschieden. Hier wird die oben angegebene *Intraframe-*, *Interframe-* oder *Skip-Codierung* festgelegt. Bei der Interframe-Codierung ist optional das Zuschalten des so genannten *Schleifenfilters* möglich.

In den Ebenen *Group of Blocks Layer*, *Macroblock Layer* und *Block Layer* sind eine Reihe von Parametern definiert, um eine fehlerfreie und synchrone Bildübertragung zu gewährleisten. In der *Group of Blocks Layer* existieren Resynchronisationsmechanismen und Angaben zu Makroblockadressen[34].

34 Dies ist wichtig, um bei wenig Bildänderung eine optimale Skip-Codierung durchführen zu können.

Auf der Ebene der Makroblöcke sind recht viele Parameter definiert. Doch bei weitem nicht jeder Parameter muss bei jedem Makroblock übertragen werden. Der so genannte *Makroblocktyp* gibt an, welcher Parameter im jeweiligen Makroblock definiert ist. Tabelle 5.1-1 zeigt die VLC-Tabelle für die Makroblocktypen.

VLC-Tabelle Makroblöcke

Tab. 5.1-1: *VLC-Codierung der Makroblocktypen bei ITU-T H.261*

Intra	*Filter*	*Quant.*	*MC*	*CBP*	*TC*	*VLC*
x					x	0001
x		x			x	0000011
				x	x	1
		x		x	x	00001
			x			000000001
			x	x	x	00000001
		x	x	x	x	0000000001
	x		x			001
	x		x	x	x	01
	x	x	x	x	x	000001

Intra kennzeichnet die Umschaltung zwischen Intraframe- und Interframe-Codierung. Die Kombination mit dem Schleifenfilter (*Filter*) macht nur im Interframe-Modus Sinn. Mit dem Parameter *Quant.* wird signalisiert, dass eine neue Quantisierungsstufenhöhe übertragen wird. Der gesetzte Parameter *MC* (*Motion Compensation* - Bewegungskompensation), gibt an, dass eine Bewegungskompensation durchgeführt werden soll. Mit dem Parameter *CBP – Coded Block Pattern*, lässt sich festlegen, aus welchen der 6 Koeffizientenblöcken DCT-Koeffizienten übertragen werden sollen. Die $2^6 = 64$ verschiedenen Kombinationen sind in einer eigenen VLC-Tabelle abgelegt, wobei die Sonderfälle „*kein Koeffizientenblock*“ und „*alle 6 Koeffizientenblöcke*“ besonders kurze Codeworte besitzen. Der Fall, dass keine Transformationskoeffizienten übertragen werden sollen (*Skip-Codierung*), lässt sich auch leicht über den Parameter *TC – Transformation Coefficient* anwählen. Der Coder wird diesen Fall immer dann signalisieren, wenn der Prädiktionsfehler hinreichend klein ist.

H.261 Anwendung Bildtelefonie

Aus der Tabelle ist ersichtlich, dass der Fall der prädiktiven Codierung ohne Bewegungskompensation am häufigsten auftreten soll (*VLC* ist „1“). In der Praxis treten solche Fälle allerdings nur auf, wenn sich der Bildinhalt selbst nur selten oder sehr geringfügig ändert. Ein stationärer Hintergrund mit nur geringer Vordergrundbewegung ist zwingend erforderlich. Ersichtlich ist so eine Situation in erster Linie bei der Anwendung *Bildtelefonie* möglich, bei dem sich das Kopf-Schulterbild mit nur wenigen Bewegungen vor einem stationärer Hintergrund bewegt (kein Zoom- oder Kamerabewegungen).

5.2 MPEG-1

5.2.1 Einführung

Moving Picture Experts Group

Die *Moving Picture Experts Group* (*MPEG*) wurde 1988 aus dem *Joint Technical Committee* (*JTC1/SC29/WG11*) der *ISO/IEC* gegründet, um einen digitalen Codierstandard für Video und Audio zu entwickeln, der für die Speicherung auf digitalen Medien bis zu einer Datenrate von etwa 1.5 Mbit/s geeignet ist. 1992 wurden die Arbeiten dieser Arbeitsgruppe als *ISO/IEC 11172*[5.3] standardisiert. Als Kurzname für diesen Standard hat sich die Bezeichnung *MPEG-1* durchgesetzt. Der Standard ist in fünf Teile unterteilt: *Systems*, *Video*, *Audio*, *Compliance Testing* und *Simulation Software*. Nachfolgend wird vorwiegend der Videoteil erläutert, ergänzt durch die Beschreibung der wichtigsten Datenstrukturen, die im System-Teil festgelegt sind.

In den Standard sind eine Reihe von Erfahrungen aus der *ITU-T H.261* Norm eingeflossen. So wird in *MPEG-1* ebenfalls eine Intraframe-Codierung auf der Basis einer DCT-Transformation vorgenommen. Weiterhin wird die *Interframe-Codierung* mit einer bewegungskompensierten Prädiktionsfehlerscodierung realisiert.

Um eine digitale Speicherung von komprimierten Videosignalen beispielsweise auf CD oder DAT zu ermöglichen, sind einige neue Forderungen zu stellen, die vom *ITU-T H.261* Standard nicht erfüllt werden können:

Forderungen an MPEG-1

- Ermöglichen eines wahlfreien Zugriffs und Editierens (Stichworte *Schnittfähigkeit*, *Berücksichtigung eines Szenenwechsels*),
- Abspielbarkeit der Videosequenz vorwärts und rückwärts, auch in Zeitrafferdarstellung,
- Verarbeitung von mehr und schnelleren Bewegungen im Bild (Stichworte *Zoom*, *Schwenk*),
- Verbesserung der Codiereffizienz durch Erhöhung der Genauigkeit der Bewegungskompensation (Halbpixel-Genauigkeit).

Beschränkung: Anzahl Makroblöcke

Obwohl theoretisch auch Bildformate in TV-Auflösung (bis 768 x 576 Pixel) zugelassen sind, wird nur das CIF-Format (352 x 288 Pixel für 50 Hz Systeme und 352 x 240 Pixel für 60 Hz System) mit einer Farbunterabtastung von 4:2:0 verwendet, womit identische Makroblockstrukturen entstehen, wie sie bereits von *ITU-T H.261* her bekannt sind (vgl. Abb. 5.1-3). Der Grund hierfür liegt in der Beschränkung der maximalen Anzahl von 396 Makroblöcken pro Bild und 9900 Makroblöcke pro Sekunde. Für 625/50 Hz-Systeme ergeben sich für das CIF-Format

$$Z_{MB,25\,Hz} = \frac{352}{16} \cdot \frac{288}{16} \cdot 25 = 396 \cdot 25 = 9900, \qquad 5.2\text{-}1$$

also genau die maximal bei MPEG-1 erlaubte Anzahl an Makroblöcken. Die gleiche Anzahl von Makroblöcken pro Sekunde ergibt überdies auch für 525/30 Hz-Systeme:

$$Z_{MB,30\,Hz} = \frac{352}{16} \cdot \frac{240}{16} \cdot 30 = 330 \cdot 30 = 9900. \qquad 5.2\text{-}2$$

Die maximal bei *MPEG-1* erlaubte Datenrate (Video und Audio) beträgt

$$R_{MPEG-1,\max} = 1856000\ Bit/s. \qquad 5.2\text{-}3$$

Mit dieser Datenrate ist das Format prädestiniert für die Speicherung von Audiovisuellen Daten auf CD-ROM Medien.

Vorverarbeitung von TV-Material

Um TV-Bildmaterial mit der höheren Bildauflösung von 720 x 576 Pixel (*ITU-R BT.601*, 50 Hz System) verwenden zu können ist eine einfache Formatkonversion notwendig. Diese sorgt auch gleichzeitig für die Progressiv-Umwandlung des Zeilensprungbildes, das in MPEG-1 selbst nicht codiert werden kann. Abb. 5.2-1 zeigt die entsprechend notwendige Vorverarbeitung.

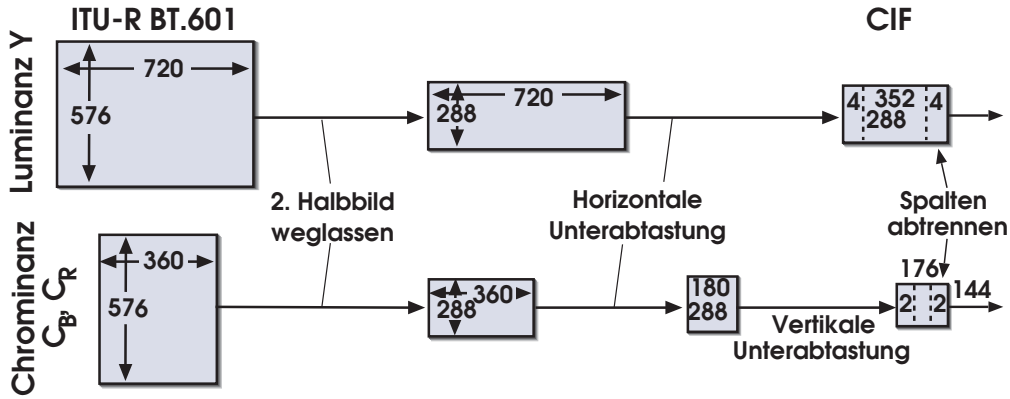


Abb. 5.2-1: Vorverarbeitung von TV-Bildmaterial für die MPEG-1 Codierung

Hinweis auf MPEG Audio

Um eine Vorstellung zu bekommen, wie groß die Datenrate für das reine *MPEG-1* Videosignal werden darf, sei an dieser Stelle noch kurz auf die festgelegten Datenrate bei der *MPEG-1* Audiocodierung eingegangen. Weitere Einzelheiten zu den verschiedenen MPEG Audiocodierungen sind beispielsweise in [5.4] nachzulesen.

Für *MPEG-1* sind insgesamt drei verschiedene *Audio Layer* definiert. Die minimale Datenrate für Audio liegt bei 32 kbit/s, maximal sind 448 kbit/s (*MPEG-1 Layer I*), 384 kbit/s (*MPEG-1 Layer II*) und 320 kbit/s (*MPEG-1 Layer III*[35]) möglich.

MPEG Generischer Standard

Die *MPEG-1* Norm definiert im wesentlichen nur die Syntax eines gültigen *MPEG-1* Datenstromes. Man spricht daher von einem *generischen Standard*. Dazu werden verschiedene Tools beschrieben, welche eine Kompression des Eingangsdatenstromes ermöglichen (z. B. DCT-Intraframecodierung, Interframe-Codierung mit

35 Audio *MPEG-1 Layer III* ist wegen seiner effizienten Codierung unter der Abkürzung *MP3* populär geworden und wird daher vielfach außerhalb eines MPEG-1 Multiplexformats als reines Audioformat genutzt.

Bewegungskompensation etc.). Bewusst nicht definiert wird, wie die Implementierung des Encoders im Detail vorgenommen werden muss, um eine gültigen *MPEG-1* Datenstruktur zu erzeugen. Damit wird erreicht, dass für die Implementierung des Encoders eine maximale Freiheit besteht. Der Encoder kann selber entscheiden, welche zusammengestellten Codierformen für das jeweilige Bildmaterial die besten Ergebnisse bezüglich Aufwand, Kompressionsrate und Bildqualität liefern. So kann bei gleichem Bildmaterial und gleicher Zieldatenrate eine recht unterschiedliche Bildqualität erreicht werden. Insbesondere der Aufwand, den ein Encoder betreibt, um eine möglichst optimale Interframe-Codierung mit Bewegungskompensation zu erzielen, entscheidet über die Bildqualität am Decoder. Dieser Ansatz führt, wie bereits in Kapitel 4 erwähnt, zu einer unsymmetrischen Komplexität von Encoder und Decoder.

5.2.2 Hierarchieebenen

Auch bei *MPEG-1* unterscheidet man zwischen verschiedenen Hierarchieebenen (Abb. 5.2-2).

MPEG-1 Hierarchieebenen

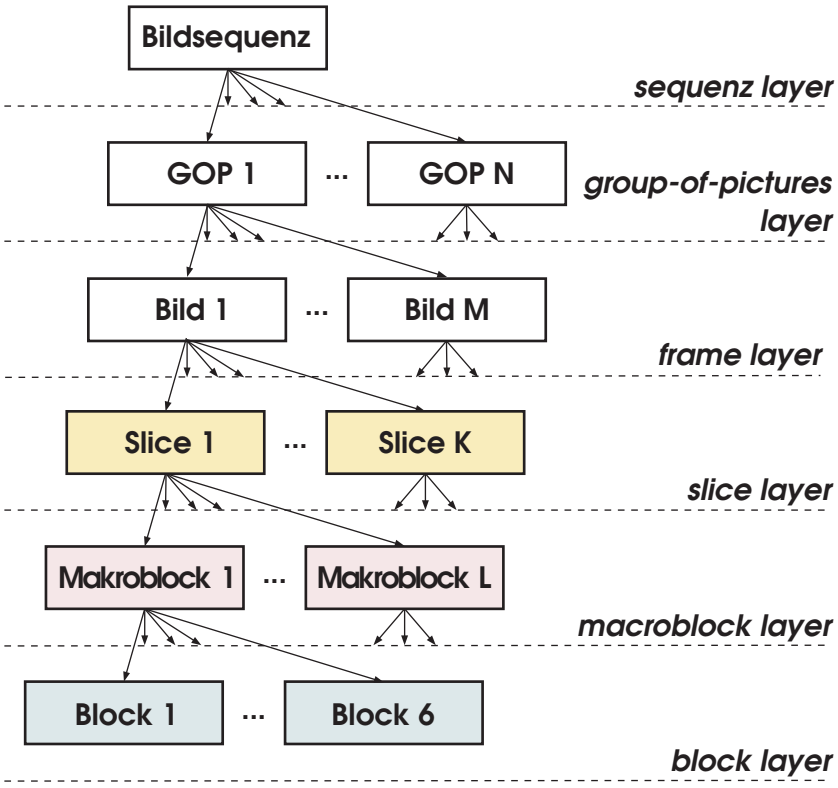


Abb. 5.2-2: Hierarchieebenen bei MPEG-1

Im Gegensatz zu *ITU-T H.261* (Abb. 5.1-2) ist die Hierarchie um eine Ebene erweitert worden. Die *GOB*-Struktur entfällt. Stattdessen werden die Makroblöcke zeilenweise in so genannten *slices* angeordnet (vgl. Abschnitt 5.2.4). Zusätzlich werden auf *Frame-Ebene* mehrere aufeinanderfolgende Bilder zu einer Bild-Gruppe (*group-of-pictures – GOP*) zusammengefasst. Der detaillierte Aufbau der *GOP* wird im nachfolgenden Abschnitt behandelt.

5.2.3 MPEG *Group-of-Pictures* – Übersicht

GOP

Der Videodatenstrom wird bei MPEG-1 zeitlich in so genannte *Group-of-Pictures – GOP* aufgeteilt. Bei einer typischen MPEG-Sequenz wiederholt sich die Struktur der *GOP* in der Bewegtbildsequenz (Abb. 5.2-4). Jede dieser GOP kann bis zu drei verschiedene *Bildtypen* enthalten und ist normalerweise zwischen 9 und 30 Bilder lang sein:

Bildtypen

- *Intraframe-codierte* Bilder – ***I-Frame*** oder ***I-Bild*** (DCT-codiert) – *Intra Coded Frames*,
- *Interframe-codierte* Bilder, die bewegungskompensierte Blöcke enthalten, die aus der Vorhersage *eines* vorherigen *I-* oder *P-Bildes* hervorgehen – ***P-Frame*** oder ***P-Bild*** (codiert mit hybrider DCT) – *Predictive Coded Frames*,
- *Interframe-codierte* Bilder, die bewegungskompensierte Blöcke enthalten, die aus der Vorhersage *eines* vorherigen *und eines* nachfolgenden *I-* und / oder *P-Bildes* hervorgehen (*bidirektionale* Prädiktion) – ***B-Frame*** oder ***B-Bild*** (codiert mit hybrider DCT) – *Bidirectionally Predictive Coded Frames*,
- Sonderfall ***B-Frame*** oder ***B-Bild***: *Interframe-codierte* Bilder, die bewegungskompensierte Blöcke enthalten, die aus der Vorhersage *eines* nachfolgenden *I-* und / oder *P-Bildes* hervorgehen.

GOP-Struktur

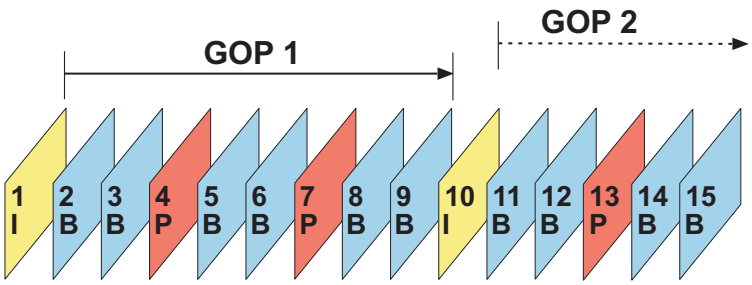


Abb. 5.2-3: Aufbau der GOP-Struktur am Beispiel einer 9er-GOP

Der nachfolgende Abschnitt beschreibt die einzelnen Bildtypen (*I*, *P*, und *B*) im Detail.

5.2.4 Intraframe-Codierung – *I-Blöcke* und *I-Bilder*

I-Bilder

Eine gültige *MPEG-GOP* beginnt mit einem *I-Bild*. Diese Bilder dienen als Referenzbilder innerhalb der *GOP* und sind wichtig für das Setzen von Schnittmarken oder für die Realisierung eines schnellen Vor- oder Rücklaufs. Damit dieses möglich ist, dürfen die Makroblöcke eines *I-Bild* keine Verweise (*Prädiktionen*) auf zurückliegende oder zukünftige Bilder beinhalten. Daher werden alle Makroblöcke eines *I-Bildes* ausschließlich *Intraframe-codiert*.

Als Codierverfahren kommt auch hier wieder die DCT-basierte Transformation (Blockgröße 8 x 8 Bildpunkte) mit anschließender Quantisierung der Transformationskoeffizienten zum Einsatz. *I-Bilder* werden damit ähnlich wie JPEG-Bilder codiert. Aufgrund der fehlenden Nutzung der Korrelation von zeitlich aufeinanderfolgenden Bildern erzeugt ein codiertes I-Bild ein höheres Datenaufkommen als ein Interframe-codiertes *P*- oder *B-Bild* (s. u.).

Slice-Struktur

Die einzelnen Bilder einer *GOP* bestehen jeweils aus der Zusammenfassung von zeilenweise zusammenhängenden Makroblöcken, den so genannten *Slices*. Diese örtliche Zusammenfassung von Makroblöcken führt insbesondere in homogenen Bildbereichen zu einer weiteren Datenreduktion. Zudem dient ein *Slice* dazu, die Fehlertoleranz des Systems zu erhöhen und die Synchronisierung zu erreichen. Dazu wird zu Beginn einer *Slice* ein 32 Bit langes Synchronwort hinzugefügt. Die minimale *Slicelänge* beträgt bei MPEG-1 ein Makroblock, maximal kann ein *Slice* alle Makroblöcke des Bildes beinhalten. Abb. 5.2-4 veranschaulicht eine mögliche Slice-Verteilung eines Bildes. Aus Gründen der Übersichtlichkeit wurde hier ein – für *MPEG-1* unübliches – Bildformat von nur 192 x 144 Bildpunkten verwendet. Außerdem wurden nur Slices definiert, die nicht versetzt über mehrere Zeile gehen. Dies ist eine Beschränkung, die erst für die *MPEG-2* Norm eingeführt wurde (vgl. Abschnitt 5.3).

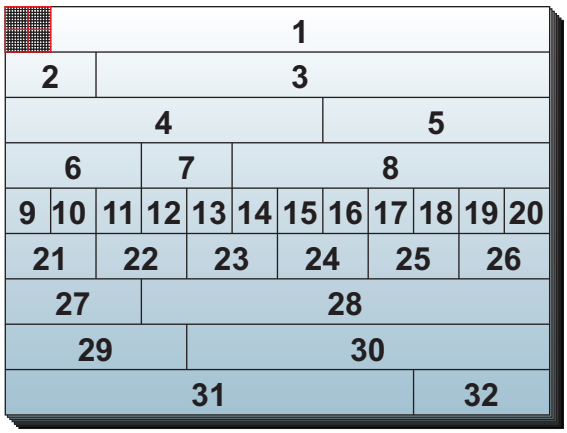


Abb. 5.2-4: Beispiel für die *Slice-Aufteilung* eines Bildes
(Block oben links: Verdeutlichung eines Makroblock des *Slice 1* mit 16 x 16 Bildpunkten, entsprechend vier Luminanz-Blöcken)

Wie bei der Codierung nach *ITU-T H.261* besteht ein Makroblock selbst aus 6 Blöcken (vgl. Abb. 5.2-5), vier 8 x 8 Blöcke für die Luminanz und je ein 8 x 8 Block für

die Farbdifferenzkomponenten C_R und C_B. So wird die auch für *MPEG-1* übliche Farbbildzerlegung 4:2:0 realisiert.

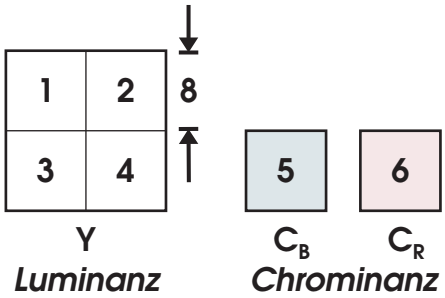


Abb. 5.2-5: Makroblockstruktur für MPEG-1, bestehend aus 6 Bildblöcken mit jeweils 8 x 8 Bildpunkten

Quantisierung

Nach der DCT-Tansformation der 8 x 8 Bildblöcke wird die Quantisierung durchgeführt, wie sie im Abschnitt 3.2.1 und Abschnitt 3.2.2 beschrieben wurde. Es werden allerdings modifizierte Quantisierungsmatrizen für die Luminanz- und die Chrominanzblöcke benutzt. Abb. 5.2-6 zeigt die Quantisierungsmatrix für die Luminanzblöcke, die für Intraframe-codierten Blöcke gilt. Für die Chrominanz wird eine Matrix verwendet, in der alle Positionen den Wert 16 annehmen.

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

Abb. 5.2-6: Standard Quantisierungsmatrix für intraframe codierte Blöcke

Die quantisierten Koeffizienten werden mit dem bekannten *Zick-Zack-Scan* zusammengefasst (vgl. Abb. 3.3-1). *RLC* und *VLC* werden vom Prinzip her so durchgeführt, wie im Abschnitt 3.3.1 und Abschnitt 3.3.2 vorgestellt wurde. Es werden allerdings auch hier andere Tabellen für die variable Lauflängencodierung verwendet. Diese lassen sich beispielsweise in [5.3] oder [5.6] nachschlagen.

5.2.5 Interframe-Codierung – *P-Blöcke* und *P-Bilder*

P-Bilder

Im Unterschied zu *I-Bildern* darf es in *P-Bildern* Makroblöcke geben, die aus der bewegungskompensierten Prädiktion eines Makroblockes aus einem vorherigen *I-* oder *P-Bild* hervorgehen. Die Bewegungsschätzung hierfür geschieht auf der Basis der 16 x 16 Pixel großen Makroblöcke. Die Genauigkeit der Bewegungsmessung ist gegenüber *ITU-T H.261* auf ein halbes Pixel erhöht worden. Je *P-codiertem* Makroblock wird ein Verschiebungsvektor zugelassen. Dieser wird, entsprechend skaliert, bei der Decodierung auch auf die verbleibenden Blöcke der Chrominanz angewandt. Abb. 5.2-7 skizziert die Prädiktionsbeziehungen zwischen *P-* und *I-Bildern* innerhalb einer 9er-GOP.

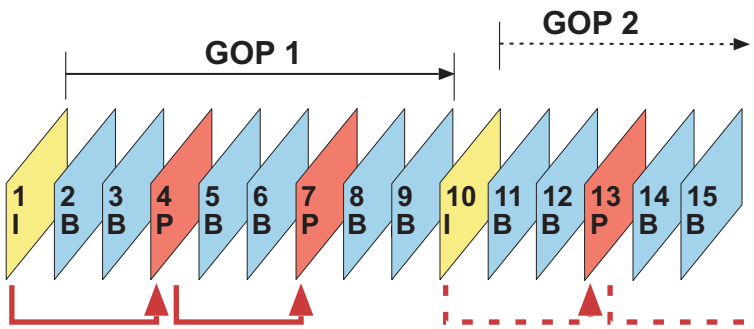


Abb. 5.2-7: Prädiktionsbeziehungen zwischen *P-* und *I-Bildern* innerhalb einer 9er-GOP

P-Bilder Prädiktionsbeziehungen in der GOP
P-Prädiktion

Abb. 5.2-8 zeigt die bewegungskompensierte Interframe-Prädiktion auf Makroblockebene für *P-Blöcke*. *P-Bilder* dürfen aber auch zwei weitere Typen von Makroblöcke besitzen: *Skipped-Makroblöcke* und *Intraframe-codierte*Makroblöcke.

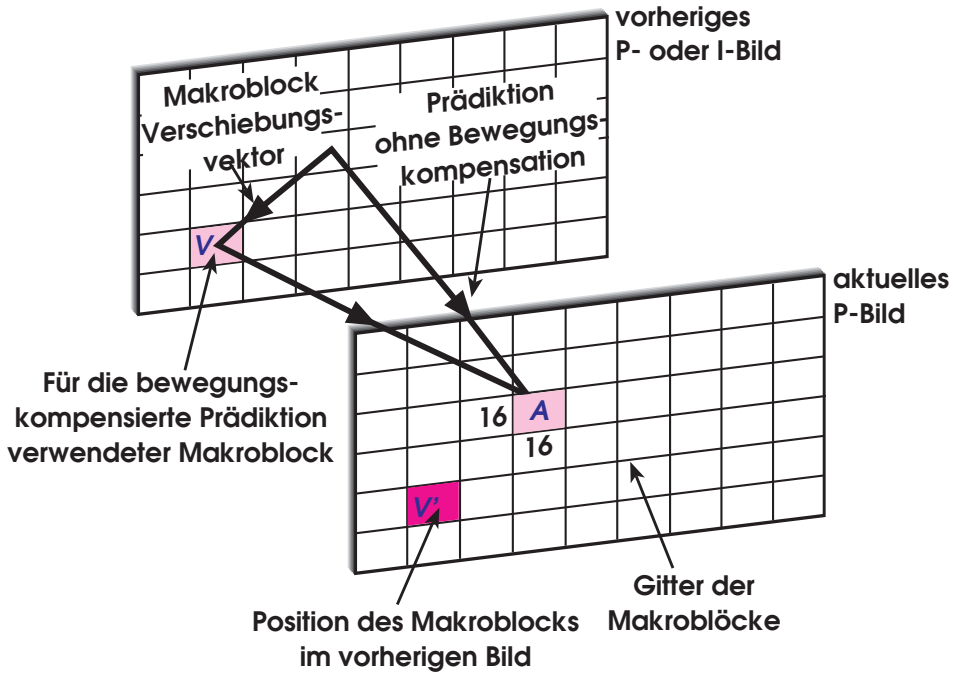


Abb. 5.2-8: Bewegungskompensierte Interframe Codierung bei P-Bildern

Hat sich der Bildinhalt eines Makroblockes nicht oder nur sehr wenig verändert, kann für diesen ein *Skip-Code* eingetragen werden. Weitere Daten sind dann für die Übertragung dieses Makroblockes nicht notwendig. Verändert sich der Bildinhalt eines Makroblockes, dann überprüft der Encoder die beiden möglichen Codierfälle in Bezug auf das erforderliche Datenaufkommen. Entweder wird der Prädiktionsfehler aus der bewegungskompensierten Prädiktion zusammen mit dem zugehörigen Verschiebungsvektor[36] codiert oder es wird eine klassische Intraframe-Codierung vorgenommen und dafür die quantisierten DCT-Koeffizienten übertragen (*I-Makroblock*).

Die DCT-Koeffizienten des Prädiktionsfehlers aus der bewegungskompensierten Prädiktion werden ebenfalls mit dem Zick-Zack-Scan zusammengefasst und quantisiert. Bei Interframe-codierten Blöcken wird eine *gleichmäßige* Quantisierungsmatrix (alle Positionen der Matrix besitzen den Wert 16) für Luminanz und Chrominanz verwendet.

Bei der Lauflängencodierung ist zu beachten, dass die DC-Koeffizienten nicht differenziell mit DC-Koeffizienten aus zeitlich zurückliegenden Blöcken verrechnet werden, sondern so wie die AC-Koeffizienten gemäß einer VLC-Tabelle direkt zusammengefasst werden. Diese Vorgehensweise ist einsichtig, da eine Subtraktion bereits bei der bewegungskompensierten Prädiktion durchgeführt wurde, und damit die Wahrscheinlichkeit gering ist, eine weitere Redundanzreduktion durch die besondere Behandlung der DC-Koeffizienten zu erreichen. Bei normal bewegten fotografischen Bildvorlagen ist das durchschnittliche Datenaufkommen eines codierten *P-Bildes* etwa halb so groß wie das eines rein Intraframe-codierten *I-Bildes*.

5.2.6 Bidirektionale Interframe-Codierung – *B-Blöcke* und *B-Bilder*

Prädiktionsbeziehungen und GOP-Typen

B-Bilder

Von besonderer Bedeutung bei der *MPEG* Codierung sind die so genannten *B-Bilder*, bei denen Makroblöcke sich aus der Prädiktion eines früheren *und* eines später folgenden *I-* oder *P-Bildes* errechnen lassen. Abb. 5.2-9 skizziert die Prädiktionsbeziehungen zwischen *B-* und *I-* bzw. *P-Bildern* innerhalb einer 9er-GOP.

P-Bilder Prädiktionsbeziehungen in der GOP

36 Für die Codierung des Verschiebungsvektors hat sich bei MPEG eine differenzielle Codierung durchgesetzt.

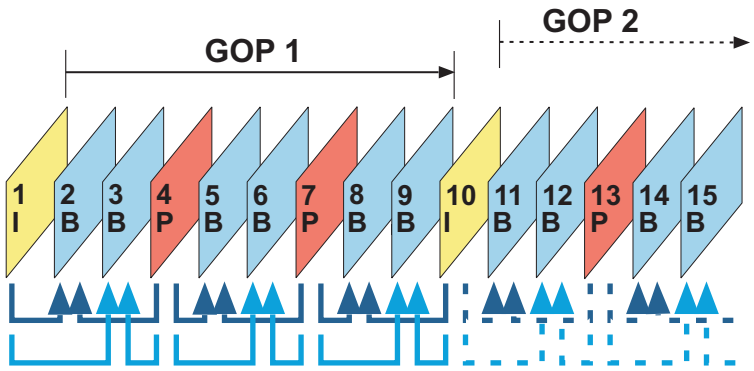


Abb. 5.2-9: Prädiktionsbeziehungen zwischen B- und I- bzw. P-Bildern innerhalb einer 9er-GOP

closed GOP

Im Gegensatz zu den Prädiktionen für *P-Bilder* gibt es hier auch Prädiktionen von Bildern, die außerhalb der Ausgangs-GOP liegen. Im Beispiel von Abb. 5.2-9 sind dies die Prädiktionen vom *I-Bild* 10 auf die *B-Bilder* 8 und 9. In den meisten Fällen stellt das kein Problem dar. Wird jedoch die Schnittfähigkeit der Bewegtbildsequenz gewünscht, sollte man so genannte *closed GOPs* verwenden. Diese unterscheiden sich von der bisher dargestellten GOP darin, dass sie immer mit einem *P-Bild* und nicht mit einem *B-Bild* endet (vgl. Abb. 5.2-10)

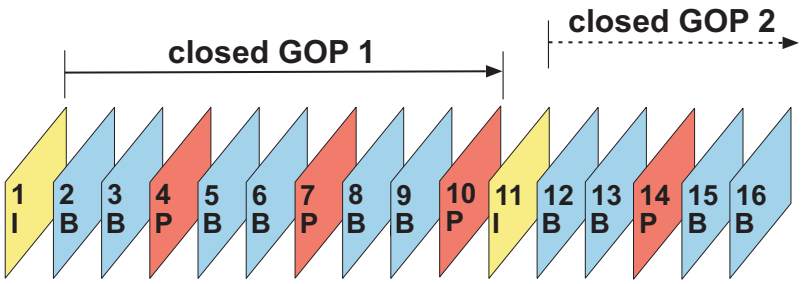


Abb. 5.2-10: Darstellung einer *closed* 10er-GOP

Die Benutzung von *closed GOPs* führt in der Regel zu einer etwas höheren Datenrate, da *P-Bilder* in der Regel mehr Speicherplatz benötigen als B-Bilder. Hier ist das entsprechende Anwendungsziel mit zu berücksichtigen. *Reguläre* GOPs werden charakterisiert durch zwei Parameter *M* und *N*. *M* gibt die Entfernung zwischen *I-Bildern* an, *N* die zwischen *P-Bildern* bzw. bis zum nächsten *I-Bild*. Es lassen sich auch *irreguläre* GOPs aufbauen, bei denen die tatsächliche Lage der Bildtypen nicht mit der Beschreibung der beiden Parameter übereinstimmt. Solche Fälle sollten dringend vermieden werden.

Bild-Umsortierung

Weiterhin ist zu beachten, dass *B-Bilder* nur decodiert werden können, wenn das vorherige und das nachfolgende Bezugsbild (*I-* oder *P-Bild*) bereits am Decoder vorliegt. Um dies zu ermöglichen, werden für die Übertragung die *P-Bilder* vor die entsprechenden *B-Bilder* positioniert (*Bild-Umsortierung*). Abb. 5.2-11 zeigt die Umsortierung für das Beispiel einer 9er-GOP aus Abb. 5.2-3.

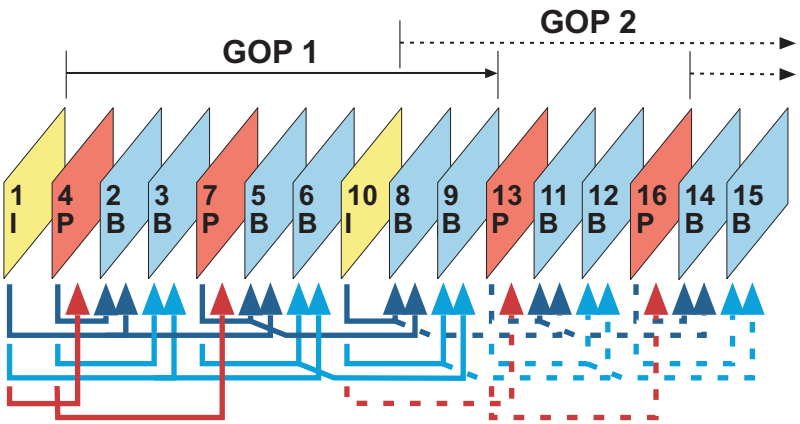


Abb. 5.2-11: Bild-Umsortierung am Beispiel einer 9er-GOP mit Prädiktionsbeziehungen zwischen *I-*, *P-* und *B-Bildtypen*

Es ist erkennbar, dass der zeitliche Ablauf der GOPs nun ineinander verschachtelt ist.

Bidirektionale Prädiktion auf Makroblock-Ebene

B-Prädiktion

Abb. 5.2-12 veranschaulicht die bewegungskompensierte bidirektionale Prädiktion auf Makroblock-Ebene.

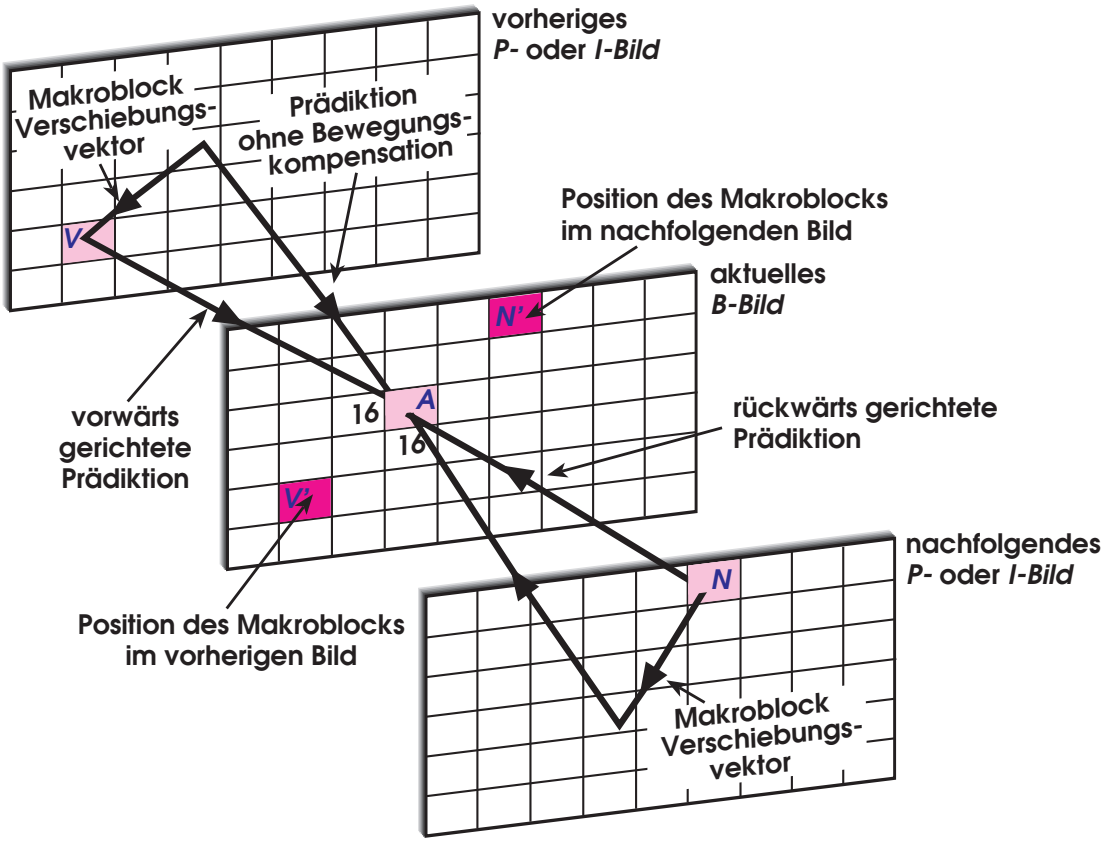


Abb. 5.2-12: Bewegungskompensierte *bidirektionale* Prädiktion

Codiermethoden bei B-Bildern

Der Rechenaufwand für die Codierung von Makroblöcken in *B-Bildern* ist in der

Regel höher als in *P-Bildern*. Wie in *I-* bzw. *P*-Bildern wird zunächst das Datenaufkommen für eine Intraframe-Codierung (*I-Makroblock*) und eine *vorwärts gerichteten* Interframe-Codierung (*P-Makroblock*) ermittelt. In *B-Makroblöcken* ist überdies auch eine *rückwärts gerichtete* Interframe-Codierung möglich. Für die Berechnung des Datenaufkommens für diesen Fall wird die gleiche Prozedur wie bei den *P-Makroblöcken* angewandt.

Interpolierte Prädiktion

Als vierte Codiermethode ist in *B-Bildern* die *bidirektionale* Interframe-Codierung erlaubt. Man nennt diese Codierung auch *interpolierte Prädiktion*, da sie aus einer Interpolation der *vorwärts gerichteten* und der *rückwärts gerichteten* Prädiktion hervorgeht. Bei der normalen Prädiktion wurde der Makroblock als Prädiktor gewählt, auf den der zugehörige Verschiebungsvektor zeigte. Dieser Makroblock wurde vom aktuell zu codierenden Makroblock abgezogen, um den verbleibenden Prädiktionsfehler zu erhalten. Bei der interpolierten Prädiktion wird zunächst der Mittelwert (Bildpunkt für Bildpunkt) aus dem Makroblockes der rückwärts gerichteten Prädiktion und dem Makroblock der vorwärts gerichteten Prädiktion berechnet. Dieser wird dann vom aktuellen Makroblock abgezogen.

Formal lässt sich dieser Vorgang folgendermaßen beschreiben. *A* sei der zu codierende Makroblock aus dem aktuellen Bild, *V* sei der Makroblock, der aus dem *vorwärts gerichteten* Verschiebungsvektor hervorgeht, *N* der aus der dem *rückwärts gerichteten* Verschiebungsvektor. *P* als verbleibenden Prädiktionsfehler lässt sich dann wie folgt errechnen:

$$P_{i,j} = A_{i,j} - \left(\frac{V_{i,j} + N_{i,j}}{2} \right), \qquad \text{5.2-4}$$

wobei die Laufvariablen *i* und *j* Werte von 1 bis 16 annehmen, um alle 256 Bildpunkte des Makroblockes zu berücksichtigen. Es muss beachtet werden, dass es sich hier um eine ungewichtete Mittelwertbildung handelt, ungeachtet der tatsächlichen zeitlichen Abstände zu den *I-* oder *P-Referenzbildern*. Die zeitlichen Abstände eines B-Bildes zu seinen benachbarten Referenzbildern wird bereits bei der Größe des zugehörigen Verschiebungsvektors berücksichtigt.

Eigenschaften der B-Prädiktion

Die hier beschriebenen verschiedenen Möglichkeiten der Prädiktion in *B-Bildern* und die Mittelwertbildung sind sehr flexibel einsetzbar und führen auch bei verrauschten Bildern zu recht guten Ergebnissen. Selbst Auf- und Abblendungen werden mit erfasst, obwohl das Verschiebungsvektor-Modell diese eigentlich nicht beschreiben kann.

Makroblöcke Steuerinformationen

Die flexible Struktur der Makroblöcke macht es erforderlich, dass ähnlich wie bei *ITU-T H.261* mit den Makroblöcken zahlreiche zusätzliche Steuerinformationen übertragen werden müssen. Die wichtigsten sind: *Makroblockadresse*, *Typ des Makroblocks* (*I*, *P*, *B* oder *skipped*), Typ der DCT, *Quantisierungsskalierung* (ggf. für Luminanz und Chromimanz unterschiedlich), *codierte Verschiebungsvektoren* (*I*: kein, *P*: ein Vektor, *B*: ein oder zwei Vektoren).

5.2.7 Regelung der Datenrate

Codierkontrolle

Ähnlich wie bei anderen Konzepten für die Bewegtbildcodierung ist auch hier ein Teilsystem erforderlich, das die Datenrate am Ausgang regelt. Ein Block „*Codierkontrolle*“ übernimmt die Aufgabe der Steuerung hierfür. Wie in Abschnitt 4.3 ausgeführt, kann eine konstante Datenrate durch die Beeinflussung der Quantisierung erzielt werden. Dazu wird in Abhängigkeit des Füllstandes des Ausgangspuffers die Quantisierungsmatrix mit einem gemeinsamen Faktor variiert, der Werte zwischen 1 und 31 annehmen kann. Erst wenn die gröbste Quantisierung nicht mehr ausreicht, um das Überlaufen des Puffers zu verhindern, werden *skipped* Makroblöcke generiert, d. h. Makroblöcke werden ohne Datenaufkommen übersprungen. Ein Decoder *friert* das Bild eines übersprungenen Makroblockes solange ein, bis wieder Daten für diesen Block übertragen werden. Wenn für den umgekehrten Fall selbst mit der feinsten Quantisierung ein Leerlaufen des Puffers nicht erreicht werden kann, müssen so genannte *Stopfbits* eingefügt werden. Mit Hilfe einer „*Multipass*“-Technik kann die Planung des Datenaufkommens optimiert werden und so die mittlere Bildqualität erhöht werden. Der Aufwand am Encoder ist dann aber ungleich höher und für die Echtzeitcodierung in der Regel ungeeignet.

CBR und VBR VBV

Insgesamt ist das Schwanken eines *MPEG*-Datenstromes jedoch kaum zu verhindern, da *I*-, *P*- und *B-Bilder* vom Prinzip her ein recht unterschiedliches Datenaufkommen besitzen. Es hängt im wesentlichen von der Anwendung ab, ob am Ausgang eine konstante Datenrate erzwungen werden muss, oder ob kleinere Schwankungen in der Datenrate erlaubt sind. Daher werden im MPEG-Datenstrom weitere Informationen zur Datenraten-Kontrolle untergebracht. Es sind dies der Typ des Datenstroms *constant bit rate* (*CBR*) oder *variable bit rate* (*VBR*) sowie die Größe des erforderlichen Datenpuffer (*MPEG-1*: maximal 320 kByte). Obwohl der Mode *VBR* bei *MPEG-1* möglich ist, wird er überwiegend erst im Standard *MPEG-2* (s. u.) angewendet. Nicht verwechselt werden darf der Mode des *VBR* mit dem, was hinter der Abkürzung *VBV* (*Video Buffering Verifier*) steckt. Unter *VBV* wird der Eingangs- bzw. Ausgangspuffer verstanden, der für die Regelung der Datenrate innerhalb einer GOP notwendig ist. Diese Regelung findet auch im *CBR* Mode statt. Abb. 5.2-13 zeigt den typischen Verlauf eines Datenpufferfüllstandes, der mit einem optimierten *VBV* erreicht wird.

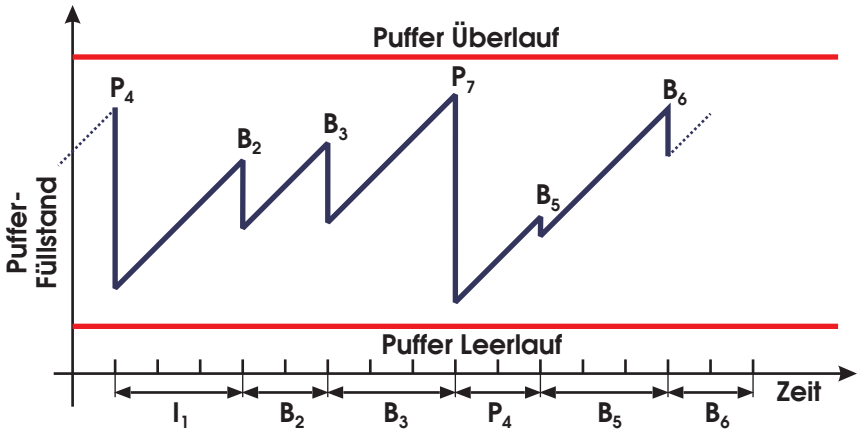


Abb. 5.2-13: Beispiel eines typischen Verlaufes des Datenpufferfüllstandes

Regelmechanismus TM5

Als ein Beispiel für den Regelmechanismus der Datenrate soll hier das *Test Model 5 rate control – TM5* [5.5] kurz beschrieben werden. Dieses Verfahren strebt ein konstantes Datenaufkommen auf GOP-Ebene an. Dabei wird im ersten Ansatz davon ausgegangen, dass die Bildtypen *I*, *P* und *B* ein Datenverbrauch im Verhältnis von

$$D_I : D_P : D_B = 140 : 52 : 36 \qquad 5.2\text{-}5$$

besitzen. Für dieses Verhältnis werden im ersten Durchgang die entsprechenden Quantisierungen eingestellt und rekursiv mit den tatsächlichen Datenraten korrigiert. Das Verfahren ermöglicht es auch, sich nach und nach an das tatsächliche Datenverhältnis der Bildtypen *I*, *P* und *B* anzupassen und damit relativ wenige Iterationen für die Bereitstellung einer konstanten Datenrate auf GOP-Basis benötigt. *TM5* ist ein relativ einfach arbeitender Algorithmus für die Regelung der Datenrate. In guten *MPEG-Encodern* arbeiten jedoch ausgefeiltere Verfahren.

In einer Reihe von Übertragungssystemen werden zudem auch die bereits oben erwähnten variablen Datenraten (*VBR*) verwendet. Hier ist die optimale Regelung der mittleren Datenrate ein sehr wichtiger Faktor zur Erzielung einer hohen Bildqualität. Mit komplexen Systemen wird in mehreren Durchläufen das Eingangsbildmaterial zunächst analysiert und bewertet. Wenig und viel bewegtes Bildmaterial kann sich über den zeitlichen Verlauf hin in der erforderlichen Datenrate ausgleichen, vorausgesetzt es steht ein genügend großer Puffer zur Verfügung. Weiterhin werden GOPs gezielt an erkannten Szenenschnitten ausgerichtet.

5.2.8 Blockschaltbild des MPEG-1 Encoders

Da bei *MPEG-1* auch eine hybride Codierung angewendet wird, weist das Blockschaltbild eines *MPEG-1* Encoders Ähnlichkeiten zu einem *ITU-T H.261* Encoder auf (vgl. Abb. 5.1-1, Abb. 5.3-2). Entscheidende Erweiterungen sind aber für die Umschaltung zwischen *Intraframe-Codierung* (*I-Bild*), *Vorwärts-Prädiktion* (*P-Bild*), *Rückwärts-Prädiktion* (*B-Bild*) und *Bidirektionale Prädiktion* (*B-Bild*) notwendig. Die bidirektionale Prädiktion erfordert zudem eine zweite Einheit, bestehend aus Bildspeicher, Bewegungskompensation und Mittelwertbildung. Abb. 5.2-14 zeigt das so gewonnene Blockschaltbild des *MPEG-1* Encoders. Aus Grün-

den der Übersichtlichkeit wurde die Regelung mit dem Ausgangpuffer nicht berücksichtigt.

MPEG-1 Blockschaltbild

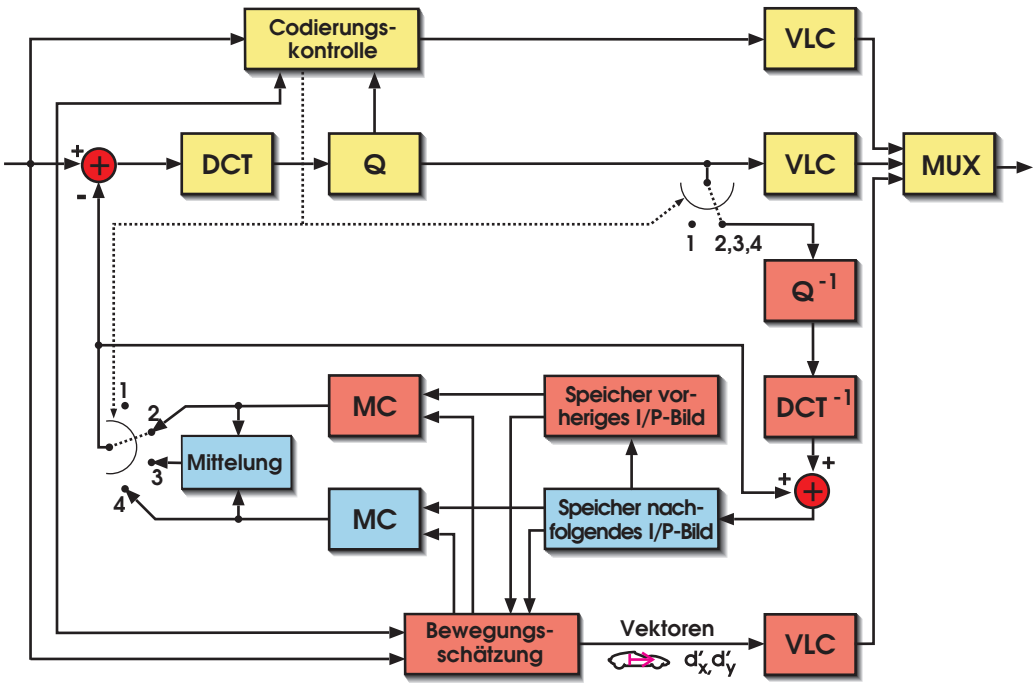


Abb. 5.2-14: MPEG-1 Encoders ohne Berücksichtigung des Ausgangspuffers
Farben-Schaltblöcke, die zusätzlich gebraucht werden:
I-Bild (gelb), *P-Bild* (rot), *B-Bild* (blau)
Schalterstellungen:
1 intraframe (*I-Bild*);
2 vorwärts gerichtete Prädiktion (*P-Bild*);
3 bidirektionale Prädiktion (*B-Bild*);
4 rückwärts gerichtete Prädiktion (*B-Bild*)
MC – *Motion Compensation* (*Bewegungskompensation*)

Selbsttestaufgabe 5.2-1:

Was versteht man unter dem Prädiktionsfehlersignal?

Selbsttestaufgabe 5.2-2:

Errechnen Sie die Anzahl der Makroblöcke pro Sekunde, welche sich nach der MPEG-1 Codierung für ein ITU-R BT.601 Signal (50 Hz Standard) prinzipiell ergeben?

Selbsttestaufgabe 5.2-3:

Skizzieren Sie die Bild-Umsortierung einer closed 10er-GOP mit den zugehörigen Prädiktionsbeziehungen zwischen I-, P- und B-Bildtypen!

5.3 MPEG-2 / H.262

5.3.1 Einführung

Nach der erfolgreichen Einführung des *MPEG-1* Standards gab es schon bald Ansätze, *MPEG-1* Erweiterungen zu entwickeln, die auch für TV-Anwendungen nutzbar sind. Diese Bemühungen führten zur Festlegung des heute am weitesten verbreiteten zweiten Standards der *Moving Picture Expert Group – MPEG-2* „*Generic Coding of Moving Pictures and Associated Audio Information*“ (*ISO/IEC 13818*[5.8]), der in seiner ursprünglichen Form bestehend aus drei Teilen 1994/95 festgeschrieben wurde. *MPEG-2* ist abwärtskompatibel zu *MPEG-1* und kann damit als direkte Fortführung von *MPEG-1* verstanden werden. Viele grundlegende Techniken von *MPEG-1* sind auch in *MPEG-2* integriert worden. Am Standard *MPEG-2* sind fortlaufend noch bis heute zahlreiche Ergänzungen und Korrekturen vorgenommen worden. Heute besteht der Standard aus folgenden Teilen:

- Teil 1: *Systems* (Syntax, Datenstrombeschreibung, Multiplexstrukturen)
- Teil 2: *Video*
- Teil 3: *Audio*
- Teil 4: *Compliance* (Konformität zu den Teilen 1 bis 3)
- Teil 5: *Software* (Referenzimplementationen)
- Teil 6: *Digital Storage Media Command and Control - DSM-CC* (Protokolle, und Transportarchitekturen für MPEG-2)
- Teil 7: *Advanced Audio Coding – AAC* (verbesserte Audiocodierung)
- Teil 8: *10 Bit Video* (nicht fortgeführt)
- Teil 9: *Extensions for a Real Time Interface for Systems Decoders* (Definition der MPEG-2 Schnittstelle für einen Digital-Decoder)
- Teil 10: *Conformance Extensions for DSM-CC* (Konformität zu Teil 6)
- Teil 11: *IPMP* (Verschlüsselungs-, Rechte- und Urhebermanagement)

Die ISO/IEC Norm *13818-2* (Teil 2 – *Video*) ist identisch mit der *ITU-T H.262*. In diesem Kapitel sollen wieder insbesondere *MPEG-2 Video* und die zugehörigen Spezifikationen der Syntax – *Systems* (Teil 1) behandelt werden. Nachfolgend sind zunächst die wichtigsten Erweiterungen und Verbesserungen des *MPEG-2 Standards* gegenüber dem *MPEG-1 Standard* zusammengefasst.

Profiles und Levels

Übersicht Erweiterungen

Innerhalb des MPEG-2 Standards stehen verschiedene *Funktionalitäts-* und *Auflösungsstufen* zur Auswahl. So sind Datenraten zwischen 2 und 300 Mbit/s bei Ortsauflösungen vom *CIF-Format* (352 x 288 Pixel) bis hin zum hochauflösenden Bildformat mit 1920 x 1088 (ursprünglich: 1152) Pixeln möglich.

Weiterhin besteht die Möglichkeit der Skalierbarkeit in Bezug auf den erzielbaren Rauschabstand (*SNR Scalability – Graceful Degradation*) und in Bezug auf die Übertragung unterschiedlicher Ortsauflösungen innerhalb eines Videodatenstroms (*Spatial Scalability*). Weitere Einzelheiten werden in Abschnitt 5.3.2 vorgestellt.

Farbunterabtastung

In Abhängigkeit von den verschiedenen Funktionalitätsprofilen (*Profiles*) sind die Farbabtastformate 4:2:2, 4:1:1 und 4:2:0 möglich. Selbst ein Format von 4:4:4 ist prinzipiell machbar. Mit den unterschiedlichen Farbunterabtastrastern ändert sich auch die Anzahl der Blöcke je Makroblock. Abb. 5.3-1 zeigt die Makroblock-Block Zuordnung bei den Formaten 4:4:4, 4:2:2 und 4:2:0.

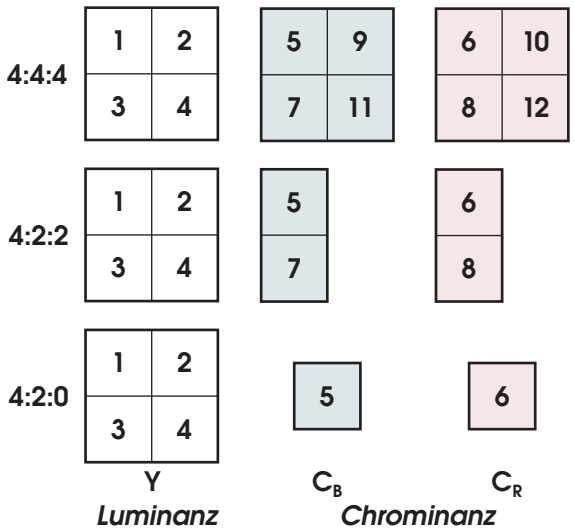


Abb. 5.3-1: Zuordnung Makroblock-Block bei 4:4:4, 4:2:2 und 4:2:0

5.3.1.1 Zeilensprungverarbeitung

2-3 Pulldown

Ähnlich wie bei der DV-Codierung (vgl. Abschnitt 4.1.2) ermöglicht auch die *MPEG-2* Codierung eine geeignete Verarbeitung von Zeilensprungmaterial. Auch die so genannte *2-3 Pulldown* Unterstützung, die eine 60 Hz Zeilensprungwiedergabe einer 24 Hz Filmsequenz ermöglicht, ist vorgesehen. Einzelheiten finden sich in Abschnitt 5.3.3.

Bildwiederholraten und Bildseitenverhältnis

frame rate code

Der Parameter *frame rate code* ermöglicht unterschiedliche Bildwiederholraten. Tabelle 5.3-1 zeigt die Zuordnung des Parameters.

Tab. 5.3-1: *Parameter Bildwiederholraten (frame rate code) bei MPEG-2*

Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001 ... 1111
Wert	nicht erlaubt	23.976	24	25	29.97	30	50	59.94	60	reser-viert

aspect ratio information

Beim Bildseitenverhältnis sind ebenfalls verschiedene Möglichkeiten über einen Parameter wählbar (*aspect ratio information*). Tabelle 5.3-2 gibt die Zuordnung dieses Parameters wieder.

Tab. 5.3-2: *Parameter Bildseitenverhältnis (aspect ratio information) bei MPEG-2*

Code	0000	0001	0010	0011	0100	0101 ... 1111
Wert	nicht erlaubt	quadra-tische Pixel	4:3	16:9	2,21:1	reserviert

MPEG-2 Verbesserungen

Die Codierung selbst wurde gegenüber *MPEG-1* mit folgenden Änderungen bzw. Verbesserungen versehen:

Slice-Struktur

- Auf der Slice-Ebene wurden die Möglichkeiten der *Slice-Struktur* bewusst etwas eingeschränkt um die Komplexität des erforderlichen Decoders herabsetzen zu können (vgl. Abb. 5.2-4). So sind bei *MPEG-2* nur noch Slice-Strukturen möglich, die maximal über eine Bildzeile gehen.

Genauigkeit DCT

- Bei der DCT ist eine höhere Genauigkeit von 9 bzw. 10 Bit möglich. Parallel dazu ist auch eine höhere Präzision der Quantisierung mit ebenfalls 9 oder 10 Bit vorgesehen.

Alternate Scanning

- Die Zusammenfassung der quantisierten Koeffizienten eines Blockes kann statt des standardmäßig üblichen Zick-Zack-Scanning nun auch mit einem weiteren Verfahren dem so genannten *Alternate Scanning* vorgenommen werden, das besser für Zeilensprungmaterial geeignet ist. Vorgestellt wird dieses Verfahren im Abschnitt 5.3.3.

16 x 8 Mode

- Der Bereich der möglichen Verschiebungsmessung für die Bewegungskompensation ist deutlich ausgeweitert worden. Zudem wurde ein weiterer Mode in der Bewegungskompensation hinzugefügt: 16 x 8 Mode. In diesem Mode können jeweils der obere und untere 16 x 8 Bereich eines Makroblockes mit unterschiedlichen Verschiebungsvektoren eine Bewegungskompensation beschreiben. Es ist zu beachten, dass dieser Mode nicht nur für Zeilensprung- sondern in einigen Fällen auch für Progressivmaterial eine deutliche Erweiterung und

Verbesserung der Codierqualität bedeutet. Weiter entwickelte Codierkonzepte, wie beispielsweise *MPEG-4Video*, haben diesen Ansatz systematisch fortentwickelt (vgl. Abschnitt 7.2). Für GOP-Strukturen ohne *B-Bilder* sind zudem noch die Prädiktionsverfahren *Dual Prime* eingeführt worden, die eine Bewegungskompensation von Zeilensprungmaterial mit zusätzlichen Verschiebungsvektoren ermöglichen. Eine Zusammenstellung der verschiedenen Prädiktionsverfahren für *MPEG-2* findet sich beispielsweise in [5.7] und [5.8].

- In den Videodatenstrom wurde ein Fehlerschutzmechanismus für Verschiebungsvektoren eingeführt, der eine Sicherung der wichtigen Verschiebungsdaten für die Bewegungskompensation bereits vor der Kanalcodierung möglich macht.

Anwendungen für MPEG-2

Diese Erweiterungen zeigen, dass sich mit *MPEG-2* eine Fülle von Anwendungen erfolgreich realisieren lassen, unter Beibehaltung eines vom Prinzip her gleichen Blockschaltbildes. Die Hauptanwendungsgebiete für *MPEG-2* liegen beim Digitalen Fernsehen (*Digital Video Broadcasting – DVB*), bei der digitalen Video Disk – *DVD* (auch *SVCD*) sowie bei einigen digitalen Magnetband Aufzeichnungen (z. B. *IMX*).

Nachfolgend wird auf die wichtigsten Besonderheiten von *MPEG-2* näher eingegangen.

5.3.2 Profiles und Levels

Wie bereits erwähnt kennt *MPEG-2* verschiedene *Funktionalitäts-* und *Auflösungsstufen*. Die Funktionalität bzw. Komplexität wird in verschiedenen *Profiles* unterschieden, die Beschränkung der Parameter für den Encoder und den Decoder ist über verschiedene *Levels* geregelt. Tabelle 5.3-3 zeigt die Kombinationsmöglichkeiten zwischen *Profiles* und *Levels*.

Profile @ Level

Tab. 5.3-3: *Profile-Level Kombinationsmöglichkeiten bei MPEG-2 (Profile horizontal, Level vertikal)*

Profile @ Level	Simple	Main	High	4:2:2	SNR	Spatial	Multi-view
High		MP@HL	HP@HL	422P@HL			MVP@HL
High1440		MP@H14L	HP@H14L			SSP@ H14L	MVP@ H14L
Main	SP@ML	MP@ML	HP@ML	422P@ML	SNRP@ ML		MVP@ML
Low		MP@LL			SNRP@ LL		MVP@LL

Profile-Typen im Detail

Die *Profiles* legen einen Teilbereich (*Teilstandard*) fest (*Farbabtastung*, Verwendung der *Bildtypen*, Möglichkeit einer *Hierarchischen Codierung - Skalierbarkeit*). Tabelle 5.3-4 zeigt einige wichtige Eigenschaften der *Profiles*.

MPEG-1 CPB

MPEG-1 CPB

Um die Abwärtskompatibilität zu *MPEG-1* zu gewährleisten gibt es ein speziell zu *MPEG-1* kompatibles Profil (*MPEG-1 constrained parameters bitstream – CPB*; nicht in Tabelle 5.3-4 aufgeführt). Obwohl die Forderung besteht, dass *MPEG-2* Decoder auch *MPEG-1* Datenströme korrekt verarbeiten sollen, ist dies in der Praxis nicht immer der Fall.

Profiles: Funktionalitäten

Tab. 5.3-4: *Auswahl von Funktionalitäten bei verschiedenen Profilen (Profile) (vgl. [5.8])*

Funktionalität	***Simple*** Profile	***Main*** Profile	***High*** Profile	***4:2:2*** Profile	***SNR*** Profile	***Spatial*** Profile	***Multi-view*** Profile
Farbabtastraster	4:2:0	4:2:0	4:2:2, 4:2:0	4:2:2, 4:2:0	4:2:0	4:2:0	4:2:0
mögliche Bildtypen	I, P	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B
Skaliermode	–	–	SNR / örtlich	–	SNR	SNR / örtlich	zeitlich
Genauigkeit DC-Koeffizient [Bit]	8, 9, 10	8, 9, 10	8, 9, 10, 11	8, 9, 10, 11	8, 9, 10	8, 9, 10	8, 9, 10

Simple Profile (SP)

Simple Profile

Dieses Profile ist für preiswerte und wenig komplexe Anwendungen ausgelegt. Aufwändige *B-Bild* Prädiktion und -Kompensation sind nicht erlaubt, ebenso wenig wie eine hierarchische Codierung („Skalierung“, verstanden als Multiplex, bestehend aus mehreren Videodatenströmen mit unterschiedlichen Parametern). Es ist nur eine Auflösungsstufe (*Level*) möglich.

Main Profile (MP)

Main Profile

Hierbei handelt es um das Standardprofil, das für viele Anwendungen (*DVD, DVB*) genutzt wird. In insgesamt 4 Auflösungsstufen (*Levels*) vom CIF-Format bis hin zu HD-Formaten sind alle Bildtypen zugelassen. Eine örtlich oder zeitlich orientierte hierarchische Codierung ist auch hier nicht vorgesehen.

5.3.2.1 High Profile (HP)

High Profile

Dieses Profil ist für die hochqualitative Codierung der verschiedenen HD-Formate vorgesehen. Es stehen insgesamt 3 Auflösungsstufen zur Verfügung. darüber hinaus besteht die Möglichkeit, eine hierarchische Codierung zu verwenden, die jeweils einen *Base-Layer* und einen *Enhancement-Layer* zusammenfasst. Die *Level*-Eigenschaften des *Base-Layers* liegen in etwa eine Auflösungsstufe niedriger als die des *Enhancement-Layers*.

4:2:2 Profile (422P)

4:2:2 Profile

Für den Fernseh-Produktionsbereich wurde 1996 als Ergänzung zum *MPEG-2* Standard ein eigenes Profil hinzugefügt, das den hohen Anforderungen bei der Verarbeitung von Studio-Videomaterial gerecht wird. Das Profile *422P* ist aus dem *Main Profile* mit dem 4:2:2-Abtastraster hervorgegangen. Als maximale Datenrate sind nun jedoch 50 Mbit/s erlaubt. Diese hohe Datenrate ist notwendig, um besonders kurze GOPs mit hoher Qualität übertragen bzw. aufzeichnen zu können. GOPs aus wenigen Bildern sind wiederum wichtig, um Szenenschnitte (quasi) Bildgenau im Studio vornehmen zu können. Wegen der zunehmenden Bedeutung von HD-Formaten ist hierfür inzwischen auch ein *High-Level* Format[37] innerhalb des *4:2:2 Profiles* definiert worden. Mit diesem sind Datenübertragungsraten bis 300 Mbit/s möglich.

SNR, Spatial (SSP) und Multiview (MVP) Profile

SNR Profile

Diese Profile sind für spezielle Anwendungen vorgesehen. In jeweils zwei Auflösungsstufen (*Level*) ermöglicht das *SNR*-Profil eine zweistufige hierarchische Codierung, wobei sich die Skalierbarkeit ausschließlich auf die *Qualität* bezieht. Als Anwendungsfall ist hier beispielsweise an eine Videoübertragung mit zwei Prioritätsstufen gedacht. Jede der beiden Qualitätsstufen hat zwar die gleiche Video Auflösung ist aber in Bezug auf die Übertragung mit einem unterschiedlichem Fehlerschutz versehen. So wird die Bildwiedergabe nicht abrupt abgebrochen, wenn sich die Qualität des Übertragungsmediums verschlechtert, sondern in dann etwas schlechterer Qualität weitergeführt. Mit dieser Eigenschaft wird ein analog-ähnliches Übertragungsverhalten simuliert, welches als *Graceful Degradation* bezeichnet wird. Abb. 5.3-2 veranschaulicht den Sachverhalt.

Graceful Degradation

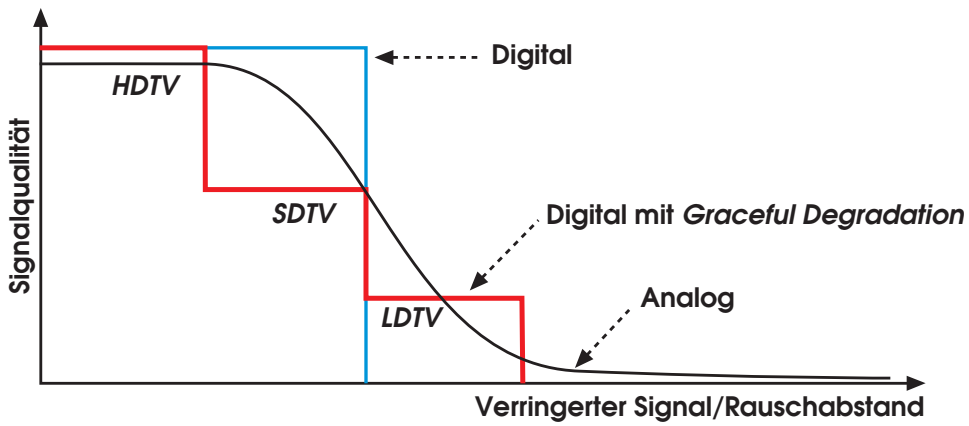


Abb. 5.3-2: *Graceful Degradation* beim *SNR*-Profil Signalqualitätsstufen:
HDTV – High Definition Televison,
SDTV – Standard Definition Television,
LDTV – Low Definition Television

37 Verschiedene Level-Parameter werden in Tabelle 5.3-5 spezifiziert.

Spatially Scalable Profile

Beim *Spatially Scalable Profile* (*SSP*)ist ebenfalls eine hierarchische Codierung vorgesehen. Hier unterscheiden sich allerdings die Hierarchieebenen, *Base-Layer* und ein *Enhancement-Layer*, in der jeweiligen Auflösung. Maximal drei Stufen sind möglich, wenn eine der beiden Hierarchieebenen noch zusätzlich in Form einer *Qualitätsskalierung* aufgespaltet wird. Gedacht ist dieses Profil, um in einem Videodatenstrom parallel und kompatibel hochaufgelöste und normal aufgelöste Bildsequenzen zu übertragen.

Multiview Profile

Das *Multiview Profile* (*MVP*) ist für Anwendungen vorgesehen, bei denen mehrere zeitgleiche Blickwinkel notwendig sind. Ein Beispiel hierfür wäre ist die Übertragung von stereoskopischem Bildmaterial.

Die hierarchische Codierung von *MPEG-2* hat sich in der Praxis nicht durchsetzen können und ist daher von untergeordneter Bedeutung. Dort, wo mehrere Videodatenströme zeitparallel zusammengefasst und übertragen werden sollen, wird eine der zahlreichen Möglichkeiten im Rahmen des Transportmultiplexes (s. u.) einer der hier dargestellten Profile vorgezogen.

Level-Typen

Man unterscheidet zwischen vier Auflösungsstufen, welche mit den oben genannten Profilen in einer Matrix verknüpft werden können. Tabelle 5.3-5 zeigt die maxima-

Level

len Werte für die wichtigsten Parameter der vier *Levels* des *Main Profiles.*

Tab. 5.3-5: *Level-Parameter für das Main Profile*

Parameter	***Low*** Level	***Main*** Level	***High 1440*** Level	***High*** Level
Auflösung	352 x 288	720 x 576	1440 x 1080	1920 x 1080
Bildwiederholrate	25/30	25/30	50/60	50/60
Datenrate	4 Mbit/s	15/50 Mbit/s	60/80 Mbit/s	80/300 Mbit/s

Wichtige Profile-Level Kombinationen

Tabelle 5.3-6 fasst noch einmal die wichtigsten Parameter des *MPEG-2 Main* und *4:2:2 Profiles* zusammen. Fett gedruckt sind die Parameter der Kombinationen *MP@ML* und *MP@HL*, welche die größte praktische Bedeutung besitzen. *MP@ML* wird von der *DVD*-Spezifikation verlangt, *MP@HL* von HD-Formaten, die im TV-Sektor eingesetzt werden.

Tab. 5.3-6: *Parameter für das Main und 4:2:2 Profile*

Profile @Level	**Auflösung [Pel]**	**Bildwiederholrate [Hz]**	**Datenrate [Mbit/s]**	***VBV* Größe [kByte]**
422P@HL	1920 x 1080	60	300	5760
MP@HL	**1920 x 1080**	**50/60**	**80**	**1194**
MP@HL14	1440 x 1080	50/60	60	696
422P@ML	720 x 608	50/60	50	1152
MP@ML	**720 x 576**	**25/30**	**15**	**224**
MP@LL	352 x 288	25/30	4	58

5.3.3 Codierung von Halbbildern

Halbbildcodierung

MPEG-2 ermöglicht ähnlich wie das DV-Verfahren (vgl. Abschnitt 4.1.2) eine Verarbeitung von Halbbildern. Auf den verschiedenen Hierarchieebenen wird durch entsprechende Signalisierung zwischen der Halbbild- und Vollbildverarbeitung unterschieden: Sequenz-Ebene (*progessive_sequence*), Bild-Ebene (*picture_structure*, *top_field_first*) und Makroblock-Ebene (*dct_type*). Nachfolgend wird die Codierung von Halbbildern auf Makroblock-Ebene betrachtet. In Abhängigkeit der auftretenden Bewegung im Block wird die Verarbeitung zwischen *Frame-Codierung* (Vollbildverarbeitung) und *Field-Codierung* (Halbbildverarbeitung) unterschieden. Im Unterschied zum DV-Verfahren wird bei der *MPEG-2* Codierung ein Makroblock mit 16 x 16 Bildpunkten herangezogen. Die Abb. 5.3-3 und Abb. 5.3-4 veranschaulichen die Sortierung der Zeilen für die beiden Fälle.

Frame-Codierung
Field-Codierung

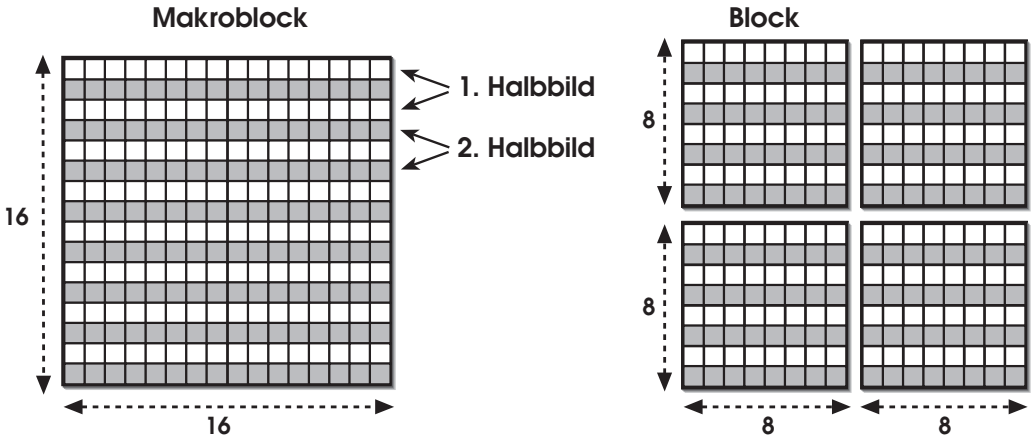


Abb. 5.3-3: Beibehaltung der Zeilensortierung des Makroblockes für die vier Luminanz-Blöcke (*Frame-Codierung* – Vollbildverarbeitung)

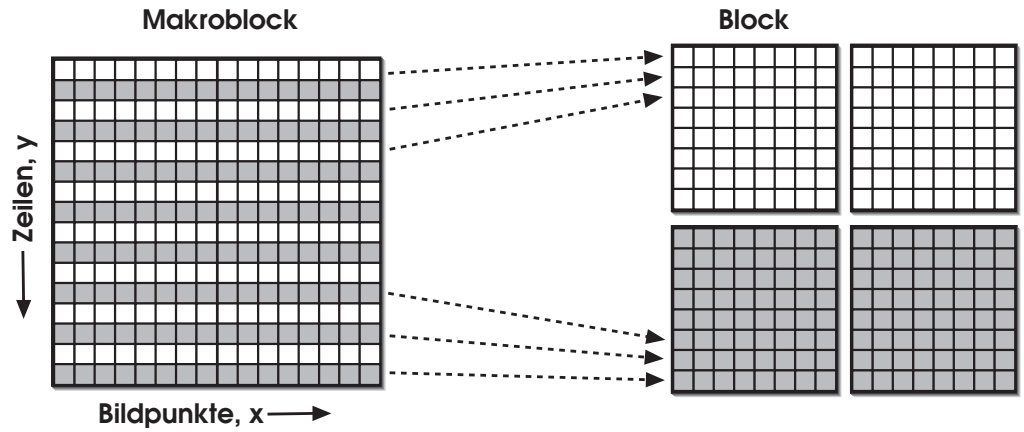


Abb. 5.3-4: Umsortierung Zeilen des Makroblockes für die vier Luminanz-Blöcke (*Field-Codierung* – Halbbildverarbeitung)

Die Umsortierung bei der Halbbildverarbeitung ist aber nur der erste Teil der Verarbeitung. *MPEG-2* bietet eine Reihe von Varianten, die beiden *„halben“* 16 x 8 Makroblöcke zu codieren. So stehen für die Codierung eines I-Makroblockes drei Möglichkeiten zur Auswahl:

- *Intra-Codierung* des kompletten Makroblockes (keine Umsortierung: *Frame-Codierung* wie in *MPEG-1*); geeignet bei wenig Bewegung im Bildmaterial,
- *Intra-Codierung* der beiden 16 x 8 großen *„halben“* Makroblöcke (Umsortierung: *Field-Codierung*); geeignet bei Bewegung im Bildmaterial,
- *Kombination* aus *Intra-Codierung* eines *„halben“* 16 x 8 Makroblockes und *Inter-Codierung* (*P-Prädiktion*) des zweiten *„halben“* 16 x 8 Makroblockes (Umsortierung: *Field-Codierung*); geeignet wenn ein Verschiebungsvektor eine gute Prädiktion zwischen den beiden Halbbildern herstellen kann.

Ähnlich wie bei I-Makroblöcken können auch P- und B-Makroblöcke behandelt werden. Die zur Verfügung stehende Möglichkeiten zur Codierung sind damit sehr viel größer. Dabei ist zu beachten, dass die *Field-Codierung* nicht ausschließlich bei Zeilensprungmaterial angewendet werden muss, oder umgekehrt, die *Frame-Codierung* nur bei Progressivmaterial. Entscheidend für die Wahl des Verarbeitungstyps ist vielmehr die Minimierung des erforderlichen Datenaufkommens bei der Codierung selbst. Tatsächlich wird der Algorithmus oft bei schnellen Bewegungen im Bild die *Field-Codierung* bevorzugen und bei wenig Bewegung die *Frame-Codierung*, unabhängig davon, ob das Ausgangsmaterial progressiv ist, oder im Zeilensprungformat vorliegt.

Da die 8 x 8 DCT-Bildblöcke bei der *Field-Codierung* aus einem Ursprungsbild mit der doppelten vertikalen Ausdehnung stammen, existiert für diesen Fall eine weitere Möglichkeit der Zusammenfassung von quantisierten DCT-Koeffizienten, das so genannte *Alternate Scanning*, das in Abb. 5.3-5 dargestellt ist.

Alternate Scanning

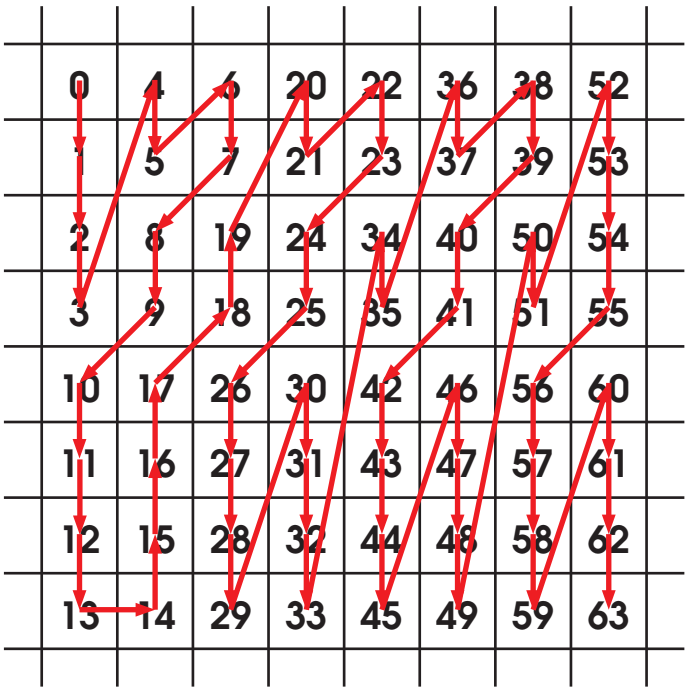


Abb. 5.3-5: Zusammenfassung der DCT-Koeffizienten für Zeilensprungmaterial (*Alternate Scanning*)

5.3.4 MPEG-Datenstrukturen und -Multiplex

Programmstrom – PS

Die von den *MPEG*-Encodern erzeugten Datenströme werden als *Elementary Streams* (*ES*) bezeichnet. Auch die entsprechende Encoder für Audio und Zusatzdaten erzeugen *Elementary Streams*. In den beiden Standards *MPEG-1* und *MPEG-2* ist in der *Systems Specification* (*MPEG-1*: *ISO/IEC 11172-1*; *MPEG-2*: *ISO/IEC 13818-1*) ein so genannter *Programmmultiplex* vorgesehen, der die Verschachtelung der verschiedenen *Elementary Streams* zu einem gemeinsamen *Program Stream* (*PS*) vornimmt. Damit diese Verschachtelung möglich wird, müssen zuvor die Datenströme der *Elementary Streams* in Pakete (*packets*) aufteilt werden. Der so umgeformte Datenstrom, bestehend aus einer Folge von Paketen des *Elementary Streams*, wird *Packetized Elementary Stream* (*PES*) genannt. Im Programmmultiplex befindet sich zudem noch eine gemeinsame Zeitbasis (*System Clock Reference – SCR*), die aus der Systemuhr (*System Time Clock – STC*) abgeleitet wird. Diese wird bei *MPEG-1* mit 90 kHz getaktet. Bei *MPEG-2* ist die Genauigkeit der *STC* erheblich gesteigert worden und entspricht dem Multiplextakt des SDI-Signals (vgl. Gl. 2.3-1) von 27 MHz.

Elementary Stream – ES

Program Stream – PS

Packetized Elementary Stream – PES

System Clock Reference – SCR

System Time Clock – STC

Abb. 5.3-6 zeigt das Blockschaltbild für die Erzeugung des *Program Streams*.

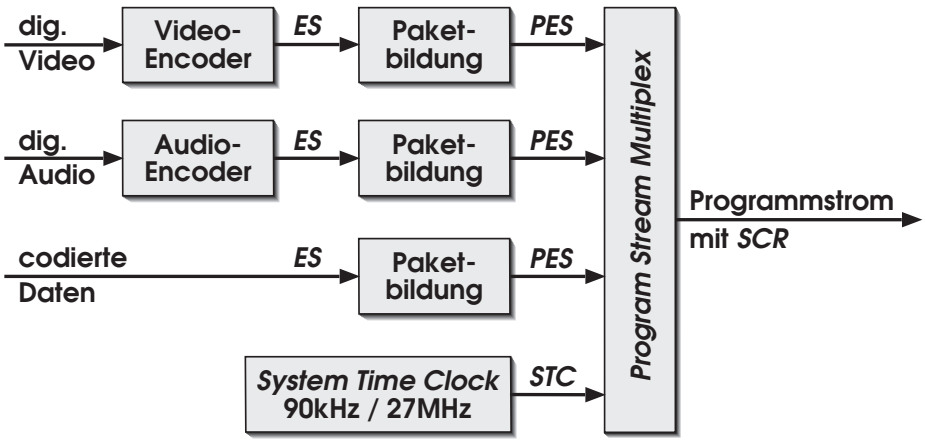


Abb. 5.3-6: Erzeugung des *Program Streams*

Die Größe eines einzelnen *PES-Pakets* kann bis zu

PES-Paket Größe

$$R_{PES} \leq 2^{16} - 1\ Byte = 65535\ Byte \qquad 5.3\text{-}1$$

betragen. Die typischen Werte liegen bei 1 kByte oder 2 kByte. Für Video-Pakete sind auch deutlich größere Pakete möglich, wobei jedoch ein *PES-Paket* immer nur die Daten eines *ES-Typs* (Video, Audio, Daten) aufnehmen darf. Das *PES-Paket* besteht aus zwei oder drei Teilen, den *Kopfdaten* und den *Nutzdaten*. Zwischen Kopfdaten und Nutzdaten kann optional noch ein so genannter *PES-Header* eingefügt werden, der zahlreiche *PES*-spezifische Informationen beinhaltet, wie eine zusätzliche Zeitmarke (*Decoding Time Stamp – DTS* oder *Presentation Time Stamp*

Decoding Time Stamp – DTS

Presentation Time Stamp – PTS

- *PTS*) oder einen Paketzähler mit dessen Hilfe Paketverluste bei der Übertragung erkannt werden können.

Die Kopfdaten (6 Bytes) beginnen mit einem 3 Byte langen Header (*packet start code prefix*). Es folgt eine 1 Byte lange *Strom-ID*, die den Typ des Datenstroms (Video, Audio, Daten) kennzeichnet. Die letzten 2 Bytes geben den Umfang der eigentlichen Nutzdaten (*Payload*, maximal 65526 Bytes) des *PES*-Pakets an.

PES-Paket

Abb. 5.3-7 zeigt den vereinfachten Aufbau eines *PES*-Pakets.

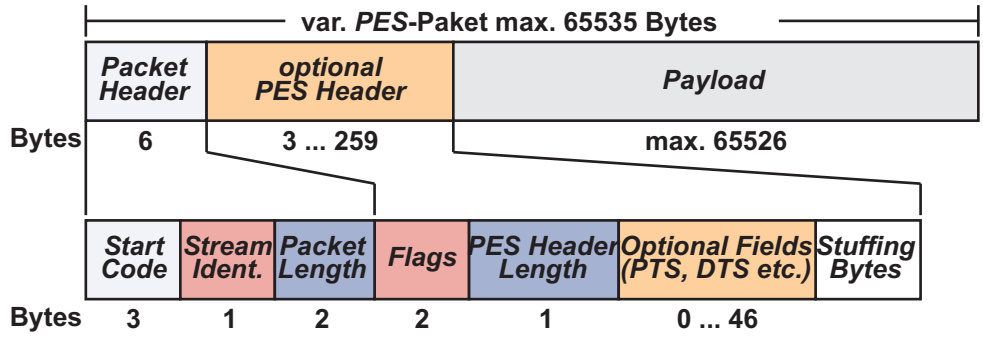


Abb. 5.3-7: Vereinfachter Aufbau eines *PES*-Pakets

PS-Multiplex

Der Multiplex des Programmstroms *PS* besteht aus der Folge von mehreren *PES*-Paketen, der in regelmäßigen Abständen durch einen weiteren *Pack-Header* (13 Byte) und einem mindestens 12 Bytes langen *System-Header* unterbrochen wird. Die Aufgabe dieser zusätzlichen *Header* besteht darin, Informationen über den Multiplex (*program mux rate*) und die o. g. Zeitbasis *SCR* an den Decoder zu übermitteln. Abb. 5.3-8 veranschaulicht den Aufbau des kompletten Programmstroms.

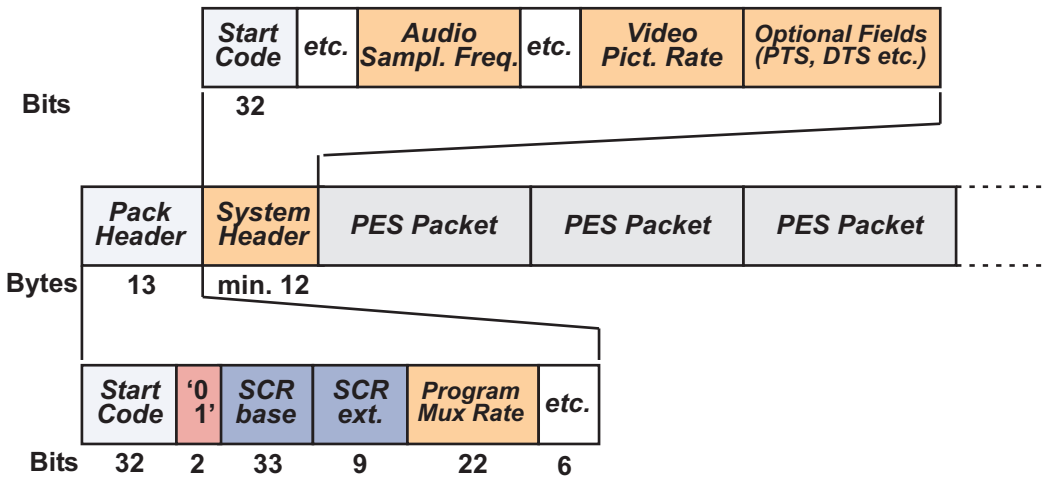


Abb. 5.3-8: Aufbau des Programmstroms

Anwendung PS-Multiplex

Da die einzelnen Pakete des Programmstroms variabel und sehr lang sein können, ist dieser Multiplex nicht optimal für die Übertragung in gestörten, paketorientierten Netzen geeignet. Daher bleibt die Nutzung des Programmstroms auf Medien

beschränkt, bei denen die Paketstruktur keine Große Rolle spielt und die ein weitgehend störsicheres Verhalten zeigen. Bei der *CD* und *DVD* sind diese Randbedingungen recht gut erfüllbar (vgl. Abschnitt 6.1).

Transportstrom – TS

Transportstrom – TS

Im Unterschied zum *Programmstrom Multiplex* arbeitet der *Transportstrom Multiplex* aus einer Folge von *TS*-Paketen mit einer festen Paketlänge von 188 Bytes[38]. Die Bildung eines *Transportstromes* ist in *MPEG-2 Systems* nicht jedoch in *MPEG-1 Systems* möglich. Es lassen sich mehrere Zeitbasen im Datenstrom unterbringen, so dass ein Multiplex aus mehreren Programm- oder Elementardatenströmen mit unterschiedlichen Zeitreferenzen möglich wird. So können beispielsweise mehrere Programme eines TV-Anbieters in mehreren Transportströmen innerhalb eines gemeinsamen *Transportstrom Multiplex* übertragen werden.

Program Specific Information – PSI
Program Clock Reference – PCR

Die einzelnen *Transportströme* enthalten weiterhin *TS*-Pakete mit der so genannte *Program Specific Information – PSI* sowie die *Program Clock Reference – PCR*. Die *PSI* beschreibt u. a., aus wie vielen und welchen Elementarströmen ein Programm zusammengesetzt ist. Die *PCR* dient als Zeitbasis und wird wie der *SCR* beim *Programmstrom* (*PS*) aus der *STC* gewonnen. Weitere Details zu den verwendeten Zusatzinformationen werden im Zusammenhang mit der Behandlung des digitalen Fernsehens (*DVB*) im Kapitel 6 behandelt. Abb. 5.3-9 zeigt im Blockschaltbild den *Transportstrom Multiplex* mit den verschiedenen Datenströmen.

Blockschaltbild TS Multiplex

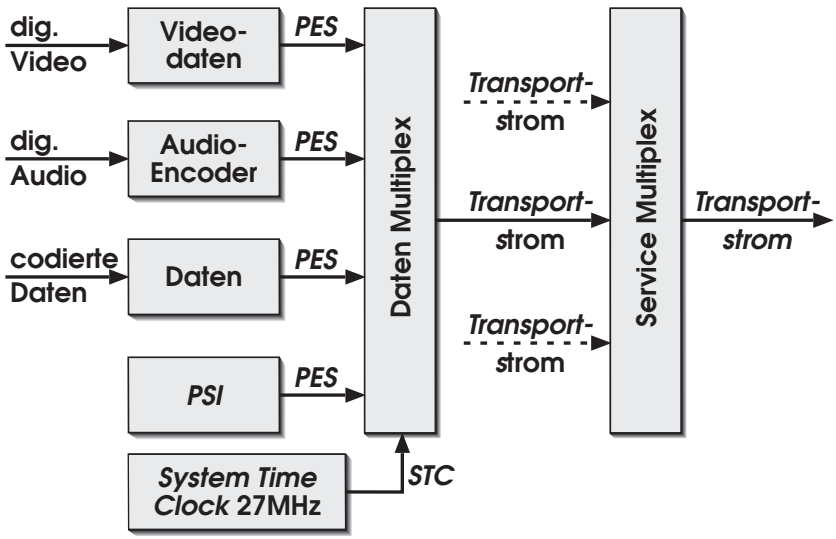


Abb. 5.3-9: Blockschaltbild zum Daten- und *Transportstrom Multiplex*

Ähnlich wie ein *PS*-Paket besteht auch ein einzelnes *TS*-Paket aus zwei bzw. drei Bereichen: einem 4 Byte langen *Header*, einem optionalen *Adaptionsfeld* mit einer variablen Länge bis zu 184 Bytes sowie dem Bereich der eigentlichen Nutzdaten

38 Aus Kompatibilitätsgründen wurde die Paketlänge von 188 Byte als Vierfaches der Länge des Datenblocks einer *ATM*-Zelle (*Asynchronous Transfer Modus*) mit 47 Bytes festgelegt.

TS-Paket

zur Aufnahme der *PES*-Daten (*Payload*, bis 184 Bytes lang). Abb. 5.3-10 zeigt den Aufbau eines einzelnen *TS*-Pakets.

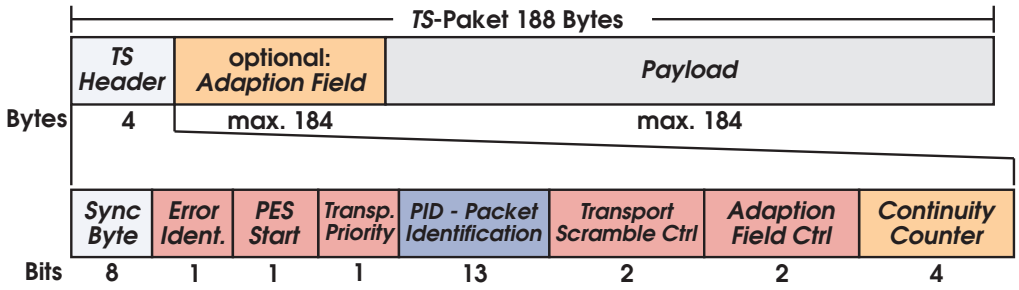


Abb. 5.3-10: Vereinfachter Aufbau eines TS-Pakets

Packet Identification – PID
Program Clock Reference – PCR

Der *TS*-Paket *Header* beginnt mit einem *Sync Byte*, womit die Rahmensynchronisierung auf die 188 Byte langen *TS*-Pakete erfolgt. Die *Packet Identification – PID* ist Teil des *Headers* und gibt Aufschluss über die Art der Programmstrominhalte, die sich in dem verbleibenden Teil des *TS*-Pakets befindet (Audio, Video etc.). Im Adaptionsfeld ist etwa alle 100 ms[39] die *Program Clock Reference* (*PCR*) angegeben, sowie weitere Daten, die sich auf das Programm beziehen. Ist noch Platz im *TS*-Paket, können weitere Daten eines *PES* (z. B. Video) untergebracht werden. In Abb. 5.3-11 ist hierzu ein Beispiel dargestellt.

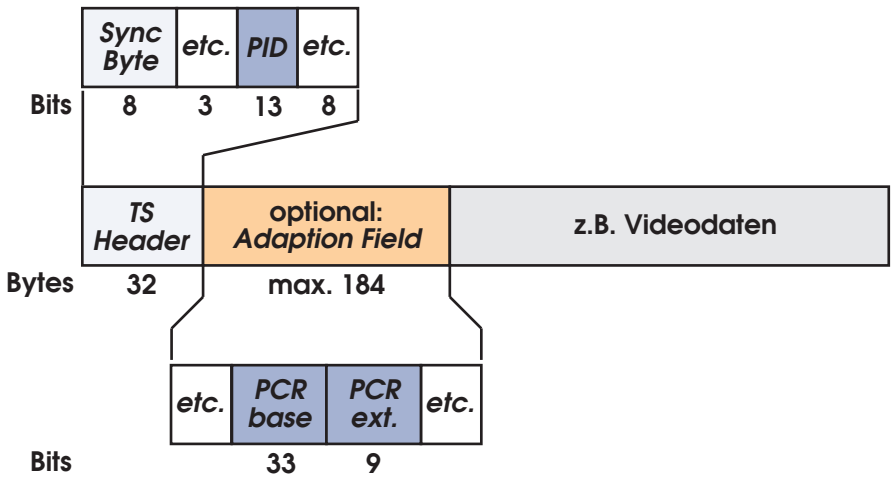


Abb. 5.3-11: Beispiel eines *TS*-Pakets mit *PCR* im *Adaptionsfeld*

Eine genaue Beschreibung zur Funktion des *PCR* ist in [5.10] angegeben.

Abb. 5.3-12 veranschaulicht noch einmal die Zuordnung der *PES*-Paketdaten auf mehrere *TS*-Pakete eines *Transportstrom Multiplexes*.

39 Bei *DVB* sind es nur 40 ms.

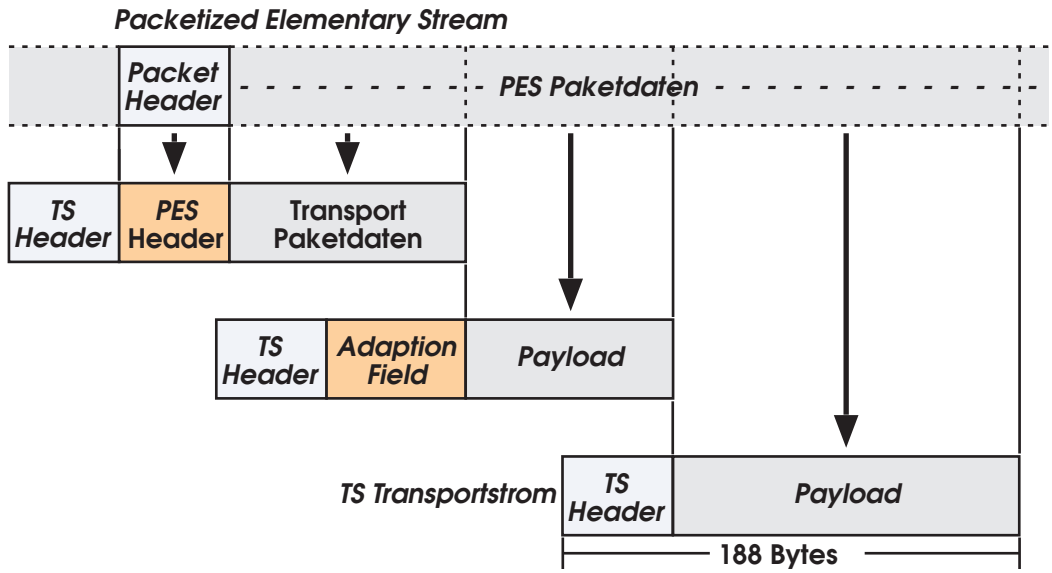


Abb. 5.3-12: Aufbau des *Transportstrom Multiplexes*

Selbsttestaufgabe 5.3-1:

Nennen Sie die sieben MPEG-2 Profiles. Welche Profile-Level Kombinationen sind für die Codierung von ITU-R BT.601 Videomaterial geeignet?

Selbsttestaufgabe 5.3-2:

Welche beiden Codierungen sind bei MPEG-2 für Halbbilder möglich? Welche zusätzlichen Modes wurden hierfür bei der Bewegungskompensation eingeführt?

Selbsttestaufgabe 5.3-3:

Erklären Sie die zweite Zusammenfassungsmethode (Alternate Scanning) der quantisierten Koeffizienten bei MPEG-2! Warum ist dieses Verfahren besser an Zeilensprungmaterial angepasst?

Selbsttestaufgabe 5.3-4:

Warum ist der Programmstrom (PS) eines Audio-Videomaterials für eine Übertragung im Internet weniger gut geeignet als der Transportstrom (TS)?

Selbsttestaufgabe 5.3-5:

Was ist der Unterschied zwischen der SCR und der PCR?

6 Videostandards in Anwendungen

Die in Kapitel 5vorgestellten Videostandards *MPEG-1* und *MPEG-2* sind in viele Anwendungen integriert worden. Zwei besonders verbreitete Anwendungen sollen in diesem Kapitel näher behandelt werden.

Video-CD DVD

In den Abschnitt 6.1 und Abschnitt 6.2 werden Videostandards für die optischen Medien *CD* (*Video-CD* und *Super Video-CD*) und *DVD* (*DVD-Video*) beschrieben.

DVB

Das europäische digitale Fernseh-Übertragungsystem *Digital Video Broadcasting - DVB* hat das analoge PAL-Fernsehen abgelöst. In drei Schritten werden die Grundlagen von *DVB* eingeführt. Abschnitt 6.3 befasst sich mit den Daten-Erweiterungen, die für *DVB* in den Transportdatenstrom integriert wurden. In Abschnitt 6.4 „*Kanalcodierung für die digitale TV-Übertragung*“ werden verschiedene Konzepte zum Fehlerschutz vorgestellt. Abschnitt 6.5 geht auf die besonderen Anforderungen der Leitungscodierung (Modulation) für die verschiedenen Übertragungswege *Satellit*, *Kabel* und *Terrestrik*[40] ein. Aus den vorgestellten Bausteinen werden die drei eingeführten *DVB*-Standards *DVB-S* (*Satellit*), *DVB-C* (*Kabel*) und *DVB-T* (*Terrestrik*) zusammengesetzt.

6.1 Videostandards für die CD

6.1.1 Entwicklung der Video-CD

Video-CD

Die *ISO* und die *IEC* haben die CD-Standards, Normen und Spezifikationen für die Datenspeicherung auf CD-Medien in „*farbigen*“ Büchern festgelegt, den so genannten *Books*. Der älteste und wohl bekannteste Vertreter ist der 1982 von Sony und Philips definierte Standard für die digitale Speicherung von Audiodaten auf CD (*IEC 908 – CD-DA*: *Red-Book*). Der erste CD-Standard, der für die Speicherung von Audio und Video auf einer CD geeignet war, ist das 1990 verabschiedete *Green-Book*. Erweiterungen wurden 1993 und 1994 hinzugefügt.

CD-I

Der im *Green-Book* enthaltende *CD-I* (*Compact Disc Interactive*) Standard wurde als erster Standard für interaktive Audio-Video-Anwendungen definiert. Obwohl bei *CD-I* erstmals auch *MPEG-1* als Codierverfahren für Video und Audio eingesetzt wird, konnte sich der Standard nicht durchsetzen. Nachteilig war sicherlich, dass die bei *CD-I* verwendeten Datenformate auf dem Betriebssystem OS-9 basieren.

Wesentlich erfolgreicher war die Weiterentwicklung des Green-Books zur so genannten *Bridge-Disc*, aus der 1992 zunächst die *Karaoke-CD* (*Video-CD* Version 1.0) in Asien hervor ging. Diese wurde im *White-Book* 1993 fortgeführt und

40 Terrestrik: Über erdgebundene Sendernetze verteilt.

stellt heute den *Video-CD* Standard 1.1 bzw. *Video-CD* Standard 2.0 (1995 definiert) dar. Die Bestrebungen, Bewegtbilddarstellungen in höherer Bildqualität auf den CD-Datenträger zu speichern, gingen noch weiter.

S-VCD

Ein letztes noch relativ weit verbreitetes Videocodierkonzept für die CD ist die so genannte *Super Video-CD* (*S-VCD*; *IEC-62107*). Zwischenzeitlich waren auch Namen wie *CVCD* oder *HQVCD* für dieses Format gebräuchlich. Im Unterschied zur *Video-CD* nutzt es schon die Vorteile der *MPEG-2* Codierung und kann daher auch Zeilensprungmaterial verarbeiten. Eine Übersicht der verschiedenen Videoformate für das CD-Medium findet sich beispielsweise in [6.3].

6.1.2 Parameter der Video-CD 2.0 und der Super Video-CD

Vergleich Video-CD 2.0 S-VCD

In der nachfolgenden Tabelle 6.1-1 sind die wichtigsten Parameter der beiden Standards *Video-CD 2.0* und *Super Video-CD* zusammengestellt[41].

Tab. 6.1-1: *Vergleich wichtiger Parameter zwischen Video-CD 2.0 und Super Video-CD*

	Video-CD 2.0	***Super Video-CD***
Video Codierverfahren	*MPEG-1*	*MPEG-2*
Bildrate [Hz]	25 (625/50) 29.97 (525/60)	25 (625/50) 29.97 (525/60)
Auflösung (625/50-System)	*CIF* 352 x 288	*2/3 ITU-R BT.601* 480 x 576
Auflösung (525/60-System)	*CIF* 352 x 240	*2/3 ITU-R BT.601* 480 x 480
Auflösung Stillbilder	*Cropped ITU-R BT.601* 704 x 576 (625/50) 704 x 480 (525/60)	*Cropped ITU-R BT.601* 704 x 576 (625/50) 704 x 480 (525/60)
progressiv / interlace	nur progressiv	progressiv / interlace
Videodatenrate	max. 1150 kbit/s *CBR*	max. 2520 kbit/s *CBR* oder *VBR*
Datenratenkontrolle	*CBR*	*CBR, VBR*
VBV Puffergröße	40 kByte	112 kByte
Audiodatenrate (*MPEG-1*, Layer II 44.1 kHz)	192 oder 224 kbit/s *CBR* 1 x Stereo/Mono oder 2 x Mono	32 bis 384 kbit/s *VBR* bis 2 x Stereo oder 4 x Mono
Überlagerte Grafik oder Untertitel	Nein	2 Bit pro Bildpunkt: 4 Farben / 4 Transparenzen

S-VCD Eigenschaften

Beim *S-VCD* Standard fällt die ungleiche Verteilung zwischen horizontaler und vertikaler Auflösung auf. Die um 2/3 horizontal reduzierte Auflösung kann für die meisten Anwendungen im Konsumerbereich immer noch als recht gut bezeichnet werden. Insgesamt lassen sich mit dem *S-VCD* Standard Videodatenströme mit deutlich höherer Bildqualität verarbeiten als mit der einfachen *VCD*. Der Bildqualitätsunterschied entspricht in etwa dem Unterschied zwischen einer *VHS*-Aufnahme

41 Die Bedeutung der einzelnen Parameter wird in den folgenden Abschnitten erläutert.

und einer *S-VHS*-Aufnahme. Hinzu kommen bei der *S-VCD* die erweiterten Möglichkeiten, Menüstrukturen für die Navigation und Untertitel einzubetten. Aufgrund der höheren Datenrate lassen sich auf der *S-VCD* nur maximal 35 bis 40 Minuten Videomaterial unterbringen, im Vergleich zu etwa 74 Minuten bei der *VCD*.

6.1.3 Beschreibung der Video-CD Standards im Detail

Mixed Mode CD

Die Daten für die *VCD* und die *S-VCD* werden in mehreren unterschiedlichen *Tracks* auf das Medium geschrieben. Dabei wird eine so genannte *Mixed-Mode CD* nach dem *Bridge Disc Konzept* erstellt.

Der erste Track der *VCD* bzw. *S-VCD* wird als Datentrack ausgelegt und besitzt gemäß den Vorgaben des *Yellow Books* (*ISO 10149*) ein Dateisystem nach der *ISO 9660* (hier: *CD-ROM XA*). Untergebracht werden jeweils Datenpakete von

Sektorgröße Mode 2, Form 1

$$R_{\text{Sektor,Mode2,Form1}} = 2048\ Bytes \qquad \text{6.1-1}$$

pro Sektor (*CD-ROM XA* Mode 2, Form 1). Auf dem Track sind das *CD-I* Anwendungsprogramm (nur bei *CD-I* für OS-9), die Trackinformationen für *Karaoke* bzw. *Musikvideo* (optional), *Einstiegspunkte*, *Play-Listen* und *Stillbilder* abgelegt.

Dem Datentrack folgen ein bis maximal 99 Tracks (*CD-ROM XA* Mode 2, Form 2), die den eigentlichen *MPEG*-Programmstrom (*MPEG-1 PS* bzw. *MPEG-2 PS*, vgl. Abschnitt 5.3.4), bestehend aus Video- und Audiodaten, beinhalten. Dabei wird pro Track nur genau eine *MPEG*-Programmstrom Datei untergebracht. Die Größe der einzelnen *PS-Pakete* ist an die Sektorengröße innerhalb der Tracks angepasst. Im Mode 2, Form 2 der *CD-ROM XA* Norm sind die Sektoren

Sektorgröße Mode 2, Form 2

$$R_{\text{Sektor,Mode2,Form2}} = 2324\ Bytes \qquad \text{6.1-2}$$

groß. Es werden daher *MPEG-PS Pakete* mit 2324 Byte (*VCD*) oder dem ganzzahligen Vielfachen von 2324 Byte (*S-VCD*) verwendet.

Optional können noch *Red Book* Tracks (*CD-DA*) folgen, die unkomprimiertes Audio im normalen *Audio-CD* Format enthalten. Die Tracks einer solchen hybriden CD können von einem normalen *Audio-CD* Spieler korrekt wiedergeben werden.

Verzeichnisstruktur VCD 2.0

Tabelle 6.1-2 zeigt die Verzeichnisstruktur für die *VCD 2.0*.

Tab. 6.1-2: *Verzeichnisstruktur der Video-CD 2.0*

Verzeichnis	Dateien	Bedeutung
\\VCD	entries.vcd	Liste der *Einstiegspunkte* in die einzelnen Videotracks
	info.vcd	Kennung / Identifikation der CD
	lot.vcd	List ID Offset Table
	psd.vcd	*Play Sequence Descriptor;* grundlegende Kontrollstruktur für Playliste, Auswahlliste, Menüelemente
\MPEGAV	AVSEQ*nn*.dat	Verweise auf die MPEG-1 Daten
\EXT	lot_x.vcd	Erweiterung zu lot.vcd
	psd_x.vcd	Erweiterung zu psd.vcd
	scandata.dat	Spielzeitbezogener Zugriff auf den MPEG-Stream (z. B. Kapitel)
	CAPT*nn*.dat	mögliche Zusatzdaten für Nutzung auf Computern
\SEGMENT	ITEM*nnnn*.mpg	falls vorhanden: Menüelemente (max. 150) (Stillbilder, Bewegtbilder, Audio)
\CDDA (Audio) optional	AUDIO*nn*.dat	Verweis auf *CD-DA* (*Red Book*) konforme Audio Tracks (s. u.)
\CDI optional	verschieden	CD-I Programm (OS-9) und zugehörige Daten
\KARAOKE optional	Karaoke.xxx	Informationsdateien für Karaoke-Anwendung

In Tabelle 6.1-3 ist die Verzeichnisstruktur für die *S-VCD* wiedergeben. Die Datei- und Verzeichnisstruktur ist zwar der *VCD* recht ähnlich, doch gibt es einige wesentliche Unterschiede. Ein optimierter Suchlauf wird beispielsweise ermöglicht, in dem in einer entsprechenden Datei (*search.dat*) die Einstiegspunkte für die I-Bilder des MPEG-Datenstroms eingetragen sind. Weiterhin werden inhaltsbezogene Informationen wie die Spielzeit eines Tracks oder die Anzahl der zugehörigen Audioströme in einer weiteren Datei *tracks.svd* vermerkt.

Beurteilung Bildqualität

Obwohl sich die erzielbare Bildqualität mit dem *DVD-Video* Standard (vgl. Abschnitt 6.2) noch deutlich steigern lässt, sind in den asiatischen Ländern die *VCD* und die *S-VCD* weiterhin recht verbreitet. Die wichtigsten Argumente für die beiden Formate liegen in ihrer sehr preiswerten Realisierung. Überall wo nur Videobeiträge bis zu etwa einer halben Stunde Dauer in semiprofessioneller Qualität gespeichert werden sollen, kann die *S-VCD* nach wie vor ein interessanter Standard sein. Mittlerweile setzen sich aber auch Alternativen durch, die bis zu etwa 2 Stunden Video- und Audiodaten in vergleichbarer Qualität auf das Speichermedium CD bannen können. Ersichtlich ist hierfür ein besonders effektives Videocodierverfahren nötig. Diese verbesserten Verfahren werden in Kapitel 7 (Stichwort: *MPEG-4*) vorgestellt.

Verzeichnisstruktur S-VCD

Tab. 6.1-3: *Verzeichnisstruktur der Super Video-CD*

Verzeichnis	Dateien	Bedeutung
\SVCD	entries.svd	Liste der *Einstiegspunkte* in die einzelnen Videotracks
	info.svd	Kennung / Identifikation der CD
	lot.svd	List ID Offset Table
	psd.svd	*Play Sequence Descriptor;* grundlegende Kontrollstruktur für Playliste, Auswahlliste, Menüelemente
	search.dat	Liste von I-Bildern als Einstiegpunkte für eine Suchfunktion
	tracks.svd	Inhaltsbezogene Informationen (Spielzeit, Anzahl Audioströme etc.)
\MPEGAV	AVSEQ*nn*.mpg	Verweise auf die MPEG-2 Daten
\EXT	lot_x.svd	Erweiterung zu lot.svd
	psd_x.svd	Erweiterung zu psd.svd
	scandata.dat	Spielzeitbezogener Zugriff auf den MPEG-Stream (z. B. Kapitel)
	CAPT*nn*.dat	mögliche Zusatzdaten für Nutzung auf Computern
\SEGMENT	ITEM*nnnn*.mpg	falls vorhanden: Menüelemente (max. 150) (Stillbilder, Bewegtbilder, Audio)

6.2 Der DVD-Video Standard

UDF

Im Unterschied zur CD, bei der jedes Format eine andere Datenstruktur auf dem Datenträger besitzt, verwendet man bei der DVD nur eine Datenstruktur für alle Anwendungen, das *Universal Disc Format – UDF*. Die Standardisierung der DVD Familie hat das *DVD-Forum* übernommen. In verschiedenen Büchern (*Books*) werden die einzelnen Anwendungen der DVD spezifiziert. Die beiden wichtigsten Vertreter sind die *DVD-ROM* für die allgemeine Speicherung von Daten und die *DVD-Video* für die Speicherung von Video-, Audio- und Interaktionsdaten. Die Standards sind jeweils in drei Bereiche aufgeteilt. Der *Part 1* legt *Physical Layer* des jeweiligen DVD-Formates fest. Das logische Verzeichnissystem ist im *Part 2* spezifiziert. Die Eigenschaften für die Anwendung selbst werden im *Part 3* definiert. Der im folgenden behandelte Standard für die *DVD-Video* stammt in seiner ersten Version aus dem Jahr 1996.

6.2.1 Parameter der *DVD-Video*

Parameter DVD-Video

Tabelle 6.2-1 vergleicht die wichtigsten Parameter der beiden Standards *DVD-Video* und *S-VCD*.

Tab. 6.2-1: *Vergleich wichtiger Parameter zwischen Super Video-CD und DVD-Video*

	S-VCD	***DVD-Video***
Video Codierverfahren	MPEG-2 4:2:0	MPEG-2 4:2:0
Bildrate [Hz]	25 (625/50) 29.97 (525/60)	25 (625/50) 29.97 (525/60)
Auflösung (625/50-System)	*2/3 ITU-R BT.601* 480 x 576	*ITU-R BT.601* *(Cropped / 1/2 / MPEG-1)* 720/704/352 x 576/288
Auflösung (525/60-System)	*2/3 ITU-R BT.601* 480 x 480	*ITU-R BT.601* *(Cropped / 1/2 / MPEG-1)* 720/704/352 x 576/288
Auflösung Stillbilder	*Cropped ITU-R BT.601* 704 x 576 (625/50) 704 x 480 (525/60)	*ITU-R BT.601* 720 x 576 (625/50) 720 x 480 (525/60)
Videodatenrate	max. 2520 kbit/s	max. 9800 kbit/s
VBV Puffergröße	112 kByte	224 kByte
Audioabtastrate	44.1 kHz	44.1 KHz, 48 kHz, 96 kHz
Audiokanäle	2 oder 4	bis 8
Audiodatenrate (*MPEG-1*, Layer II)	32 bis 384 kbit/s (44.1 kHz)	64 bis 384 kbit/s (48 kHz)
Weitere Audioformate	nein	unkomprimiert LPCM, *AC-3* (bis 448 kbit/s), *DTS* (bis 1536 kbit/s), MPEG-2 Audio mit Mehrkanalton (bis 912 kbit/s)
Überlagerte Grafik oder Untertitel	2 Bit pro Bildpunkt: 4 Farben / 4 Transparenzen	2 Bit pro Bildpunkt: 16 Farben / 16 Transp.
Spieldauer Medium	ca. 35 - 40 min.	> 120 min. ein Layer bzw. >240 min. 2 Layer

Verbesserungen DVD-Video S-VCD

Erkennbar sind die wesentlichen Verbesserungen gegenüber S-VCD:

- die volle *ITU-R BT.601* Auflösung, zusammen mit der erhöhten Bildqualität, die durch die höhere Videodatenrate möglich ist,
- die vielen differenzierten Audioformate (auch Mehrkanalton),
- die erheblich höhere Kapazität der *DVD-Video* und damit verbunden eine längere Spieldauer.

Bemerkenswerte Erweiterungen gibt es zudem auch bei den Menü- und Verzweigungsoptionen, die hier nur grob skizziert werden können. Weitere Einzelheiten finden sich in [6.3] oder [6.4].

6.2.2 Besonderheiten der DVD-Video

Sektorgröße UDF

Das *UDF*-Dateiformat besitzt eine logische Blockgröße für die Sektoren von

$$R_{\text{Sektor,UDF}} = 2048\ Byte. \qquad 6.2\text{-}1$$

Daher werden Pakete, bestehend aus *MPEG-2* Programmströmen (Audio, Video und zusätzlicher Informationen wie Navigation und grafische Überlagerungen), für *DVD-Video* Anwendungen ebenfalls in dieser Größe generiert. Wie Tabelle 6.2-1 zeigt, basieren die Parameter bei *DVD-Video* auf der *MPEG-2 Profile-Level* Kombination *Main Profile* und *Main Level* (*MP@ML*). Es ist jedoch zu beachten, dass die *DVD-Video* Spezifikation in ihren Parameterbereichen gegenüber *MPEG-2 MP@ML* eingeschränkt ist. Die wichtigsten Beschränkungen sind:

Beschränkungen DVD-Video

- Die Datenrate für Video darf maximal 9800 kbit/s sein. Praktische Werte liegen zwischen 4000 kbit/s und 7000 kbit/s.
- Erlaubt sind nur die Bildseitenverhältnisse 4:3 und 16:9. Breitwandformate wie beispielsweise 2,21:1 werden nicht vom Standard unterstützt.
- Bei den Auflösungen sind nur die in Tabelle 6.2-1 angegebenen Formate möglich.
- Die *GOP*-Länge ist auf maximal 18 Vollbilder bzw. 36 Halbbilder bei 525-Zeilen Systemen und 15 Vollbilder bzw. 30 Halbbilder bei 625-Zeilen Systemen beschränkt. Zusätzlich benötigen bestimmte *GOP*s einen zusätzlichen *Header*.
- Die Audioabtastrate ist festgelegt auf 48 kHz.
- Für die Farbdarstellung sind nur die Primärvalenzen und die entsprechenden Umwandlungsmatrizen gemäß *ITU-R BT.624 Norm M* (525-Zeilen System), *Norm B* oder *G* (625 Zeilen System) und *SMPTE 170 M* zugelassen.
- Der *MPEG-2* Modus *Low Delay* (ohne *B-Bilder* zwischen *I-* und *P-Bildern*) wird nicht unterstützt.

Die festgeschriebenen Randbedingungen für die Bildcodierung bei der *DVD-Video* schränken die Anwendungsvielfalt des Standards nicht wesentlich ein. Sie waren jedoch bei der Festlegung der Spezifikation erforderlich, um den Hardwareaufwand für *DVD-Video* konforme Wiedergabegeräte beherrschbar zu halten. Die voranschreitende Entwicklung in der Halbleitertechnologie für Video- und *General Purpose* Prozessoren machen aber die oben angeführten Einschränkungen aus technischer Sicht inzwischen überflüssig. So existieren zahlreiche DVD Wiedergabegeräte, die mit Video- und Audiodatenströmen zurecht kommen, die nicht streng *DVD-Video* konform sind.

Interaktivität bei DVD-Video

Die besondere Attraktivität der *DVD-Video* liegt jedoch nicht allein in der relativ hohen Video- und Audioqualität durch die *MPEG-2* Codierung, sondern in den vielschichtigen Möglichkeiten, eine ansprechende Benutzerinteraktivität zu gestalten. Zu den besonderen Interaktionselementen gehören die *Multi Angle*- und *Multi Language*-Funktionen, die dem Anwender die Wahl zwischen verschiedenen Sprachversionen und Blickwinkeln eines Basisszenenmaterials ermöglicht. Überdies spielt es für den Rechteinhaber der Inhalte eine große Rolle, dass die DVD-

Video über ein Konzept zum Rechtemanagement verfügt. Mit Hilfe des Regionalcodes können beispielsweise Inhalte auf bestimmte Regionen beschränkt werden. Zusätzlich hat die Dateiverschlüsselung *Content Scrambling System – CSS*[42] die Aufgabe, nur zertifizierten DVD-Laufwerken die Entschlüsselung des Datenstroms zu ermöglichen und die unberechtigte Erstellung von Kopien zu verhindern.

Datenstrukturen und logische Navigation

Bei der DVD-Video muss sorgfältig zwischen *Datenstrukturen* und *logischen Navigationsstrukturen* unterschieden werden, da Autorenwerkzeuge diese in der Praxis oft voneinander abweichend realisieren. Nachfolgend sollen zunächst die *Datenstrukturen* der DVD-Video erläutert werden.

6.2.3 Datenstruktur der DVD-Video

Dateistruktur DVD-Video

Um das vielfältige Navigationssystem der DVD-Video realisieren zu können, wird eine komplexe Daten-Hierarchie eingesetzt, die sich in einer Verzeichnisstruktur nach Abb. 6.2-1 widerspiegelt.

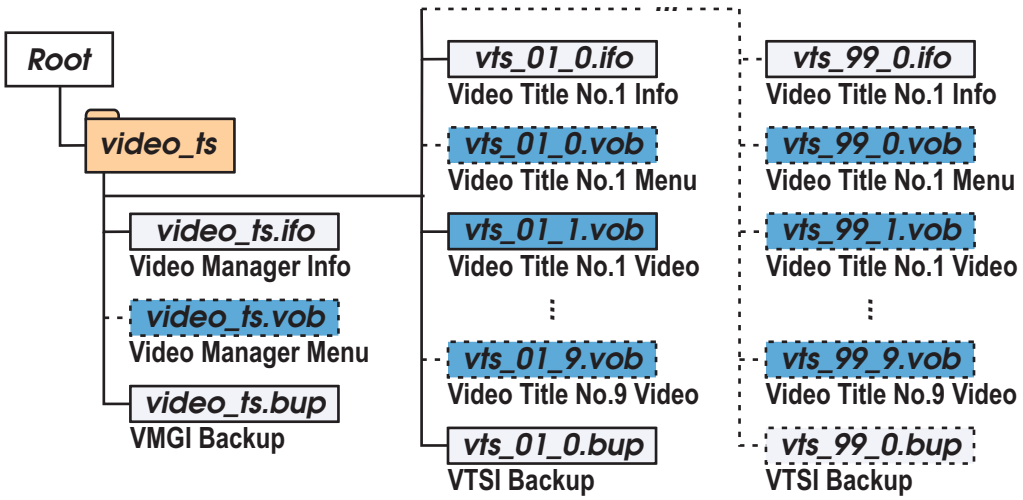


Abb. 6.2-1: Verzeichnis- und Dateistruktur der *DVD-Video*

Beschränkung Dateien

Da die DVD-Video nicht nur zum *UDF*-Dateisystem, sondern auch zur *ISO 9660*[6.1] konform sind, dürfen Dateien maximal 1 GByte groß sein. Ein größeres Medienaufkommen wird auf mehrere Dateien aufgeteilt, was aufgrund der Paketierung der MPEG-2 Basisdaten kein besonderes Problem darstellt. Für die Berechnungen der Kapazität einer DVD-Video muss beachtet werden, dass im Unterschied zu den sonstigen Geflogenheiten bei Computerumgebungen 1 GByte exakt 1 Milliarden Bytes entsprechen.

Für die Dateinamen sind gemäß den Konventionen der ISO 9660 nur Buchstaben, Zahlen und der Unterstrich „_“ zulässig. Eine Groß- oder Kleinschreibung wird in Dateinamen nicht unterschieden. Es gilt die 8.3-Konvention, d. h. es sind für den Dateinamen nur 8 Zeichen zulässig, für die Endung 3.

42 Es existieren weitere Schutzmechanismen, auf die hier nicht weiter eingegangen wird.

Dateigruppen

Auf der DVD-Video werden zwei Dateigruppen unterschieden. Zum einen gibt es Informationsdateien (**.ifo*, **.bup*[43]), die Inhaltsverzeichnisse, Menüstrukturen, Navigationstabellen und -verweise enthalten. Die andere Gruppe von Dateien (**.vob*) enthalten in der untersten Ebene einer komplexen Datenhierarchie die eigentlichen Medieninhalte wie *Video-*, *Stillbild-*, *Audio-* und *Subpicturedaten*[44] sowie *Timinginformationen* und *Sprungmarkeninformationen*. Die oberste Ebene der Hierarchie bilden die *Video Object Sets* (*VOBS*). Sie können Mediendaten aus mehreren völlig verschiedenen Videos (*Video Objects – VOB*) enthalten. Abb. 6.2-2 zeigt die physikalische Datenstruktur eines solchen *Video Object Sets*.

Datenstruktur DVD-Video

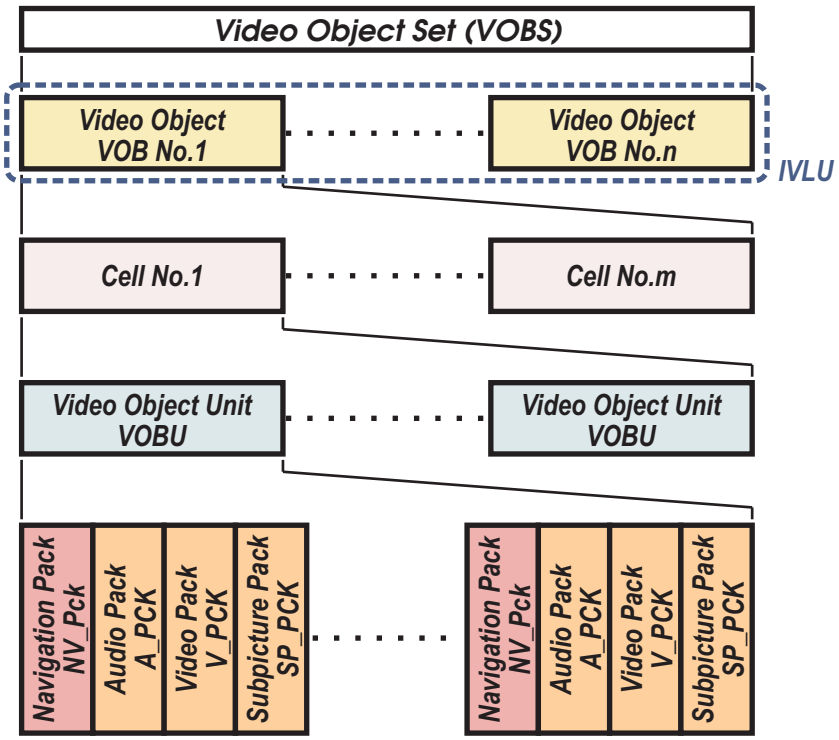


Abb. 6.2-2: Physikalische Datenstruktur eines *Video Object Sets*

Multi-Angle ILVU

Die physikalische Reihefolge der *Video Objects* auf der DVD-Video kann linear oder in geteilten *Blöcken* den *Interleaved Video Units* (*ILVU*) sein. Der Sinn dieser Zwischenstrukturebene *ILVU* ergibt sich aus der Möglichkeit der DVD-Video eine *Multi-Angle*-Wiedergabe realisieren zu können. Dazu werden bei durchlaufenden Ton[45] die verschiedenen Kamerawinkel des gleichen Szenenabschnittes miteinander in den *ILVU*s verschachtelt. Die Wahl des Kamerawinkels durch den Benutzer legt fest, welcher Sequenzabschnitt aus einem Block gezeigt werden soll. Die Größe der *ILVU*s wird so vereinbart, dass die Datenpuffer der Wiedergabegeräte optimal genutzt werden, also einerseits Unterbrechungen durch das laufende Überspringen

43 *bup*-Dateien sind identisch mit den entsprechenden *ifo*-Dateien und dienen lediglich als Backup auf der *DVD-Video* für eine eventuell notwendige Fehlerkorrektur.

44 überlagerte Grafik, z. B. Menü-Rahmen oder -Schalftlächen (Buttons), Untertitel

45 Man beachte, dass die Tondaten dazu physikalisch mehrfach auf der *DVD-Video* vorhanden sein müssen.

von Medienpassagen nicht auftauchen und es andererseits nicht zu größeren Verzögerungen beim Umschalten zwischen den Kamerawinkeln kommt. In den meisten Fällen unterstützen die produzierten Medien der DVD-*Video* keine Wahl des Kamerawinkels. In diesen Fällen entfallen die *IVLU*s und die *Video Objects* werden linear hintereinander angeordnet.

Cell VOBU NV_PCK

Ein *Video Object* besteht aus einer oder mehrerer *Zellen* (*Cell*). Der Inhalt einer Zelle kann dem Inhalt einer Szene entsprechen. Daher besitzt jede Zelle eine eindeutige *Cell-ID*, die für die Navigation notwendig ist. Jede *Zelle* kann in eine oder mehrere *Video Object Units – VOBU* aufgeteilt sein. Eine VOBU kann als die kleinste Einheit einer Szene verstanden werden und repräsentiert eine Medienlänge von etwa 0.4 bis 1.2 Sekunden Dauer. Eine VOBU besteht mindestens aus dem so genannten *Navigation Pack – NV_PCK* und optional aus einer oder mehreren MPEG-2 konformen Medienpacks, den schon bekannten *GOP*s (vgl. Abschnitt 5.2.3). In den meisten Fällen besteht die VOBU aus dem NV_PCK und den Mediendaten, die genau einer GOP von beispielsweise 15 Vollbildern entsprechen. Die Mediendaten der GOP sind im *MPEG-Programmstrom-Multiplex* organisiert, wie er in Abschnitt 5.3.4 beschrieben wurde. Dabei sind die einzelnen *PES*-Pakete nur etwa 150 bis 300 Byte lang, so dass etwa 6 bis 12 *PES-Pakete* in einen *PS-Multiplex* von 2048 Bytes Größe passen. Dies entspricht genau der Sektorengröße der DVD-Video (vgl. Gl. 6.2-1). Im Unterschied zu MPEG-Programmströmen für andere Anwendungen beinhaltet das *PS-Multiplex* bei der DVD-Video jeweils nur PES-Pakete eines Medientyps. Die möglichen Medientypen sind *Video-Packs* (*V_PCK*), *Audio-Packs* (*A_PCK*) und *Subpicture-Packs* (*SP_PCK*).

CSS

Die Packs können bei Bedarf mit dem *Content Scramble System - CSS* verschlüsselt werden. Der zugehörige *CSS-Title-Key* liegt nicht im Anwendernutzbereich (*User Data*) des Sektors von 2048 Byte, sondern eine Hierarchieebene höher im Kopf des 2064 Byte großen *Data Sectors*, der seinerseits mit weiteren 302 Byte für die physikalische Aufzeichnung fehlergeschützt[46] ist.

6.2.4 Logische Navigationsstruktur der DVD-Video

Video Manager VMG

Wie bereits erwähnt befinden sich die Daten für die logische Navigationsstruktur in den *ifo*-Dateien. Die Informationsdatei *video_ts.ifo* spielt dabei eine zentrale Rolle. Sie wird als *DVD Video Manager – VMG* bezeichnet, stellt das zentrale Inhaltsverzeichnis der DVD-Video dar und verwaltet den Einstiegspunkt in die Navigation beispielsweise mit einer *Autoplay*-Anweisung. Anschließend wird oft in ein Hauptmenü verzweigt, das in einem der bis zu 99 *Video Title Set Title* (*VTS-TT* Dateien *vts_nn_0.ifo*) aus dem *Video Title Set* (*VTS*) verwaltet wird. Die *VTS_TT* beinhaltet die so genannte *Video Title Set Information – VTSI* und ist in ihrer Form ähnlich wie der *VMG* aufgebaut. Für gewöhnlich enthält das *VTS* einen oder mehrere logische Titel (*VTS_TT*), die einem kompletten Film entsprechen können. Jeder *VTS_TT*

46 Als Fehlerschutz wird ein *Reed-Solomon Produkt Code* verwendet. Fehlerschutzmechanismen werden in Abschnitt 6.4 vorgestellt.

kann auf mehrere *Program Chains – PGC* verzweigen, welche logisch meistens die Kapitelmarken symbolisieren. Die weitere Aufgliederung geschieht in *Programs* (*PG*) und *Zellen* (*Cell Pointer*), die Szenen oder Szenenabschnitte repräsentieren können. Die logische *Cell*-Ebene und die darunter liegenden Ebenen (*VOBU*, *Pack* etc.) sind mit den bereits vorgestellten Datenstrukturen kongruent.

Titelsuchstruktur

Bei der *Titelsuchstruktur* einer DVD-Video wird ein Verfahren verwendet, das sich geringfügig von der logischen Navigationsstruktur unterscheidet. Der zentrale *Video Manager* (*VMG*) verwaltet wie gehabt seine Titel in den *VTS_TT*s, deren Teile (*Parts of Title – PTT*) können aber direkt auf einzelne *Programs* (*PG*) zugreifen, ohne den Umweg über die *Program Chain* (*PGC*) nehmen zu müssen. Die Hierarchie für die logische Navigation ist in Abb. 6.2-3 dargestellt.

Logische Navigation

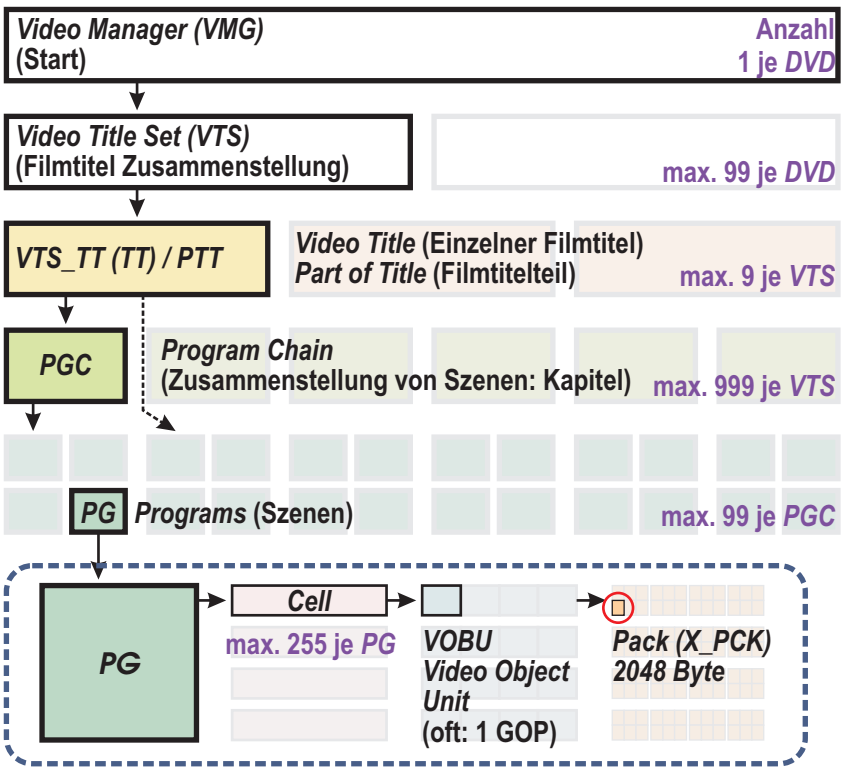


Abb. 6.2-3: Logische Hierarchieebenen für die Navigation

6.2.5 Navigation und Interaktivität

Navigationskommandos

Wie bereits beschrieben, unterscheidet die DVD-Video bezüglich der Benutzerinteraktivität zwischen *Navigationskommandos* (*Navigation Commands*) und benutzerkontrollierten *Suchfunktion*[47] (*Search Functions*). Die Suchfunktionen teilen sich in folgende Typen auf:

Suchfunktion

- *Title Search* – Aussuchen aus einer Liste von verfügbaren Titeln,
- *Part_of_Title Search* – Auswahl eines bestimmten Kapitels,
- *Program Search* – Auswahl einer bestimmten Szene,
- *Time Search* – Auswahl einer Einsprungzeit innerhalb eines *PGC*s,
- *Scan* – Schneller Vor- und Rücklauf innerhalb eines Titels,
- *GoUp* – Sprung zum Start des nächsten *PGC*s (oft: Return).

Der Autor der DVD-Video bestimmt, welche der Suchfunktionen für den jeweiligen Titel zugelassen sein sollen. Mit Hilfe der Navigationskommandos wird festgelegt, wie auf Benutzereingaben in den Menüs reagiert werden soll. Es werden sechs verschiedene Typen von Navigationskommandos unterschieden:

- *GoTo* – Verzweigung zwischen Kommandos,
- *Link* – Übergabe innerhalb der gleichen Hierarchieebene,
- *Jump* – Übergabe zwischen zwei verschiedenen Hierarchieebenen,
- *Compare* – Vergleichen und Erkennen von Parameterwerten,
- *Set System* – Einstellen von Systemparametern (*SPRM*) des DVD-Video Players,
- *Set Calculate* – Einstellen von allgemeinen Parametern (*GPRM*).

User Operation – UOP

Von der Benutzerseite aus stehen bis zu 31 verschiedene *User Operations* (*UOP*) zur Verfügung, die teilweise noch mit weiteren Angaben (z. B. Zahlenangaben über Zehnertastatur) verknüpft sein können. Die *UOP*s werden beispielsweise mit Hilfe der Fernbedienung übermittelt.

Der Ablauf der Navigation selber wird über eine eigene DVD-Video spezifische Programmiersprache realisiert. Hiermit lassen sich recht komplexe Navigationsstrukturen realisieren. Die verschiedenen Zustände des Gesamtsystems werden in einer Reihe von Variablen abgespeichert. Die DVD-Video unterscheidet 24 *System Parameter Registers – SPRM*, 16 *General Parameter Registers – GPRM* sowie 25 *User Operation Flags – UOP Flag*. Weitere Einzelheiten zu diesem Themenkomplex lassen sich in [6.4] nachlesen.

47 Der Begriff *"Suchfunktion"* mag missverständlich sein. Hieruntern fallen in der Praxis selbstverständlich auch alle Benutzereingaben, die für die Ausführung einer bestimmten Navigation notwendig sind. Navigationskommandos sind in diesem Sinne Reaktionen des DVD-Video Programm Codes auf Benutzeringaben.

Selbsttestaufgabe 6.2-1:

Warum kann eine CD-I nur auf einem speziellen Wiedergabegerät abgespielt werden?

Selbsttestaufgabe 6.2-2:

Begründen Sie, warum ein Autorentool für die Realisierung einer Multi-Angle DVD-Video für jeden Kamerawinkel einen eigenen Audiodatenstrom auf dem Medium ablegen muss, auch wenn dieser für jeden Kamerawinkel exakt gleich ist!

Selbsttestaufgabe 6.2-3:

Wie viel Speicherplatz belegt ein V_PCK auf einer DVD-Video? Nach welchem Multiplexverfahren werden V_PCK, A_PCK, SP_PCK miteinander verschachtelt?

6.3 Multiplexstrukturen für Broadcast-Anwendungen

6.3.1 Einführung

Programmspezifische Informationen

Im Abschnitt 5.3.4 wurden der *Programmstrom* (*PS*) und *Transportstrom* (TS) Multiplex von MPEG-2 eingeführt. Dabei wurde gezeigt, dass sich der Multiplex verschiedener MPEG-2 Elementardatenströme (Video, Audio, Daten) in Pakete mit konstanter Größe für Netzwerkanwendungen besonders gut eignet. Es wurde gezeigt, dass die Verschachtelung der Multiplexstrukturen auch über mehrere Hierarchiestufen gehen kann: Ein MPEG-2 TS beinhaltet beispielsweise alle notwendigen Daten für ein einzelnes Programm; mehrere einzelne TSs werden daraufhin in einem weiteren TS-Multiplex zusammengeführt. Die Trennung und Interpretation der einzelnen Transportströme am Decoder kann allerdings nur gelingen, wenn zusätzliche Begleitinformationen in die Struktur eingebettet wird. Dazu sind bereits im Standard für MPEG-2 so genannte *programmspezifische Informationen* (*ProgramSpecific Information – PSI*) definiert worden. Speziell für Broadcastanwendungen sind diese durch *dienstebezogene Informationen*[48] ergänzt worden, die sich wegen der zur PSI kompatiblen Datenstruktur problemlos in den MPEG-2 TS einbinden lassen.

Dienstebezogene Informationen

DVB Service Information

Beim europäischen digitalen TV-System – *Digital Video Broadcasting* (*DVB*) – sind diese beiden Informationsformen unter dem Begriff *Service Information* (*SI*) als *ETSI* Norm *EN 300 468*[6.5] festgelegt worden. Mit den Service Informationen lassen sich prinzipiell die folgenden Funktionen realisieren:

- automatische und Teilnehmer spezifische Konfiguration der Empfangsgeräte (z. B. Programmkonfiguration, Initialisierung, Abgleich Datum und Uhrzeit),
- komfortable Benutzerführung (z. B. über einen *Electronic Program Guide – EPG*), die eine aufbereitete Navigation durch die zahlreichen Dienste und Programme gewährleistet,
- Einführung weiterer Mehrwertdienste (z. B. Tele-Shopping).

6.3.2 *PSI*-Tabellen beim MPEG-2-TS

PSI-Tabellen MPEG-2

Die in jedem MPEG-2-TS Paket vorhandene *Packet Identification– PID*[49]teilt dem Decoder mit, welche Datenpakete für die Auswertung der jeweiligen PSI-Daten aus dem Transportstrom zu verwenden sind. Für MPEG-2 konforme PSI unterscheidet man in vier verschiedenen Arten von Tabellen:

---

48 In *DVB* wurde eine weitere, dienstebezogene Information definiert, die insbesondere die Benutzerinterakitvität über einen Rückkanal (z. B. Telefonleitungen) ermöglicht: Es ist die *Multimedie Home Platform - MHP*. Sie stellt dem Anwender sehr viele neue, teilweise auch komplexe Funktionen zur Verfügung. Auf MHP wird in diesem Kurs nicht näher eingegangen.

49 Die PID wurde in Abschnitt 5.3.4 eingeführt.

- *Program Association Table – PAT*,
- *Program Map Table – PMT*,
- *Conditional Access Table – CAT*,
- *Network Information Table – NIT*.

PAT – Program Association Table

Die *PAT* enthält eine Liste aller im Transportstrom enthaltenen Dienste / Programme (Video, Audio und Daten). Sie steht damit an der obersten Hierarchieebene und verweist auf die PIDs der zugehörigen *Program Map Table* (*PMT*). Für die PAT ist die PID Nummer Null reserviert. So ist sichergestellt, dass alle Empfangsgeräte sicher in die oberste Hierarchieebene bei der Interpretation der TS-Pakete einsteigen können.

PMT – Program Map Table

Auf der nächst niedrigen Hierarchieebene enthält die *PMT* die Verweise auf alle PIDs, die zu einem einzelnen Dienst oder Programm gehören (*Video-PID*, ein oder mehrere *Audio-PID*, *Untertitel-PID*, *Teletext-PID* etc.). Alle für die PID möglichen Ziffern außer der Null dürfen frei für die gewünschten Dienste-PIDs bzw. Programm-PIDs verwendet werden. Abb. 6.3-1 veranschaulicht das Zusammenspiel zwischen den beiden Tabellen PAT und PMT bei der Verschachtelung im Transportmultiplex von MPEG-2 an einem einfachen Beispiel.

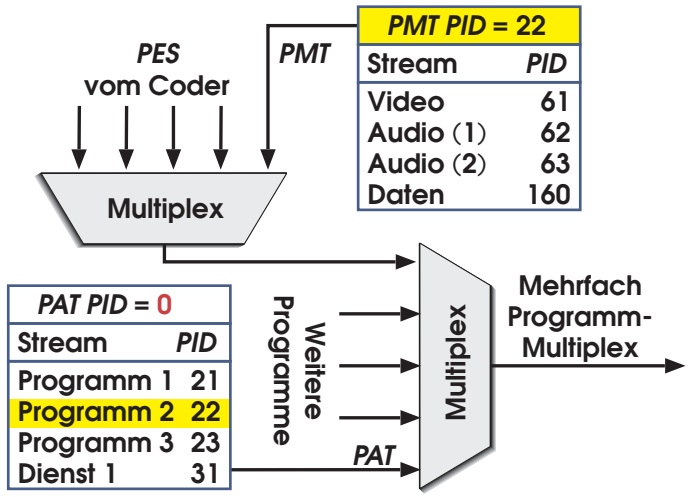


Abb. 6.3-1: Beispiel für einen MPEG-2 Transportstrom Multiplex mit mehreren Programmen

Ensemble, Bouquet

Es wird deutlich, dass viele Dienste und / oder Programme gemeinsam in einem Transportstrom-Multiplex untergebracht sind. Als Beschränkung ist hier lediglich die maximal zulässige Datenrate zu sehen, die der jeweilige Übertragungskanal zur Verfügung stellt. Die Zusammenstellung von mehreren Diensten und / oder Programmen wird bei Broadcastanwendungen meist *Ensemble* genannt, wenn es sich nur um einen Übertragungsweg handelt. Der Begriff *Bouquet* ist üblich, wenn die Zusammenfassung unabhängig vom Übertragungsweg ist.

CAT – Conditional Access Table

Die *CAT* enthält Angaben über die im Datenstrom verwendeten Verschlüsselungssysteme (falls vorhanden) und informiert über die Verfahren zur Entschlüsselung und die Zugangsberechtigungen.

NIT – Network Information Table

In der *NIT* werden die *„privaten Daten“* des Dienste- oder Netzbetreibers angegeben. Normalerweise sind das die beim jeweiligen Übertragungssystem relevanten Parameter wie *Name des Anbieters, Kanalbandbreite, Orbitalposition* und *Transpondernummer* (beim *Digitalen Satelliten Broadcast – DVB-S*), *Fehlerschutz* und *Modulationsart*.

6.3.3 Zusätzliche *Service Information* bei DVB

Wie bereits erwähnt werden alle Tabellen der *Service Informationen* (*SI*) von DVB zur PSI kompatible als paketierte Elementarströme (*PES*) in den MPEG-2 TS eingebracht. Die Charakterisierung der einzelnen SI-Tabellen wird über entsprechende Beschreibungselemente, den so genannten *Descriptoren* realisiert. Jede SI-Tabelle hat mehrere Descriptoren, die zwar unterschiedliche Informationen beinhalten können aber in ihrer Datenstruktur identisch aufgebaut sind.

In *DVB* werden standardmäßig fünf SI-Tabellen verwendet, die in ihrem Datenumfang maximal 1024 Bytes bzw. 4096 Bytes (*EIT*) groß werden können:

SI-Tabellen bei DVB

- *Bouquet Association Table* (*BAT*),
- *Service Description Table* (*SDT*),
- *Event Information Table* (*EIT*),
- *Time and Data Table* (*TDT*),
- *Running Status Table* (*RST*).

BAT Die *Bouquet Association Table* (*BAT*) enthält neben dem Namen des Bouquets des Anbieters die Angaben über alle enthaltenen Dienste und / oder Programme.

SDT In der *Service Description Table* (*SDT*) werden die Dienste und Programme kurz beschrieben (*Name des Dienstes* bzw. *Programms, Name des Veranstalters, Art und Verfügbarkeit des Dienstes* bzw. *Programms*). Während die BAT Übertragungsweg übergreifend definiert sein kann muss für jeden Transportstrom eine eigene SDT vorhanden sein.

EIT Die *Event Information Table* (*EIT*) ist in der Regel die umfangreichste Tabelle. In ihr sind die Angaben über die *gerade laufenden* und die *kommendenProgrammbeiträge*, bezogen auf den aktuellen empfangenen Transportstrom abgelegt. Dazu gehören folgende Informationen über den Programmbeitrag: *Startzeit, Dauer, Name, Beschreibung des Inhalts*. Diese Information wird ausgewertet, um im Empfangsgerät eine elektronische Programmzeitschrift (*Electronic Program Guide - EPG*) realisieren zu können. Für jeden Dienst bzw. jedes Programm existiert eine eigene EIT. Die EIT ermöglicht es auch, Informationen über Dienste bzw. Programme aus anderen empfangbaren Transportströmen zu erhalten.

TDT Mit der *Time and Data Table* (*TDT*) kann die interne Uhr des Empfangsgerätes synchronisiert werden. In der *Running Status Table* (*RST*) sind Informationen über den Status der Programmbeiträge abgelegt (*laufend, nicht laufend, pausierend*). Damit lässt sich der Start eines Beitrags exakt mitteilen, auch wenn dieser entgegen der

Progammankündigung früher oder später erfolgt. Die RST ist für die Steuerung von Aufzeichnungsgeräten vorgesehen und vergleichbar mit dem *VPS*-Dienst, der bei der analogen Fernsehtechnik diese Aufgabe übernimmt.

Teletext in DVB

Im DVB-Standard haben sich noch eine Reihe von weiteren Diensten nach dem Konzept der SI-Information etabliert. So wird beispielsweise auch der klassische Teletext in PES-Paketen übertragen [6.6]. Das vorgestellte Konzept zeigt, dass der MPEG-2 Transportdatenstrom recht flexibel mit zusätzlichen Daten angereichert werden kann, ohne dass bisherige Empfangsgeräte bei der Decodierung Schwierigkeiten bekommen müssen. Pakete mit unbekannten Diensten können unproblematisch übersprungen werden.

6.3.4 Statischer und dynamischer Transportmultiplex

ST – Stuffing Table

Ein DVB Transportdatenstrom beinhaltet in der Regel ein Bouquet mit vielen verschiedenen Diensten und Programmen. Bei der Zusammenstellung der vielen Einzeldatenströme ist zu beachten, dass die maximal zulässige Datenrate des Übertragungskanals nicht überschritten werden darf. Da insbesondere paketierte Elementardatenströme (PES) für Video eine stark variable Datenrate besitzen können, gilt es, Datenspitzen möglichst optimal innerhalb des Gesamtmultiplexes zu kompensieren. Der *statische Transportmultiplex* geht den einfachen Weg und reserviert für jeden PES die maximal zulässige Datenrate. Nicht benötigte Bandbreite wird mit so genannten *Stuffing Tables* (*ST*) aufgefüllt (z. B. mit PID-Nr. 8191), welche keinen Inhalt besitzen und auch dazu geeignet sind nicht mehr benutzte SI-Tabellen zu überschreiben.

Ersichtlich ist dieses Verfahren in Bezug auf die Bandbreitenausnutzung nicht sehr effektiv. Der Ansatz des *dynamischen Transportmultiplexes* berücksichtigt die Datenratenstatistik der parallel übertragenen Programme. Dieses Verfahren ist recht sinnvoll, da es unwahrscheinlich ist, dass die Datenratenspitzen bei Videoströmen auf Grund von komplexen Szenen bei allen Programmen zur gleichen Zeit auftreten. Das System weist jedem Programm zunächst eine feste Datenrate zu, die dem statistischen Durchschnitt entspricht, und lagert die Datenspitzen als Pakete in Programme aus, die momentan nur wenig Daten zu transportieren haben. Auf diese Weise kann die Bandbreiteneffizienz bei der Übertragung nochmals bis zu einem Faktor sechs gesteigert werden.

6.3.5 Beispiel

Abb. 6.3-2 zeigt zum Abschluss des Kapitels ein einfaches Beispiel für einen einfachen Transportstrom mit zwei Hauptprogrammströmen.

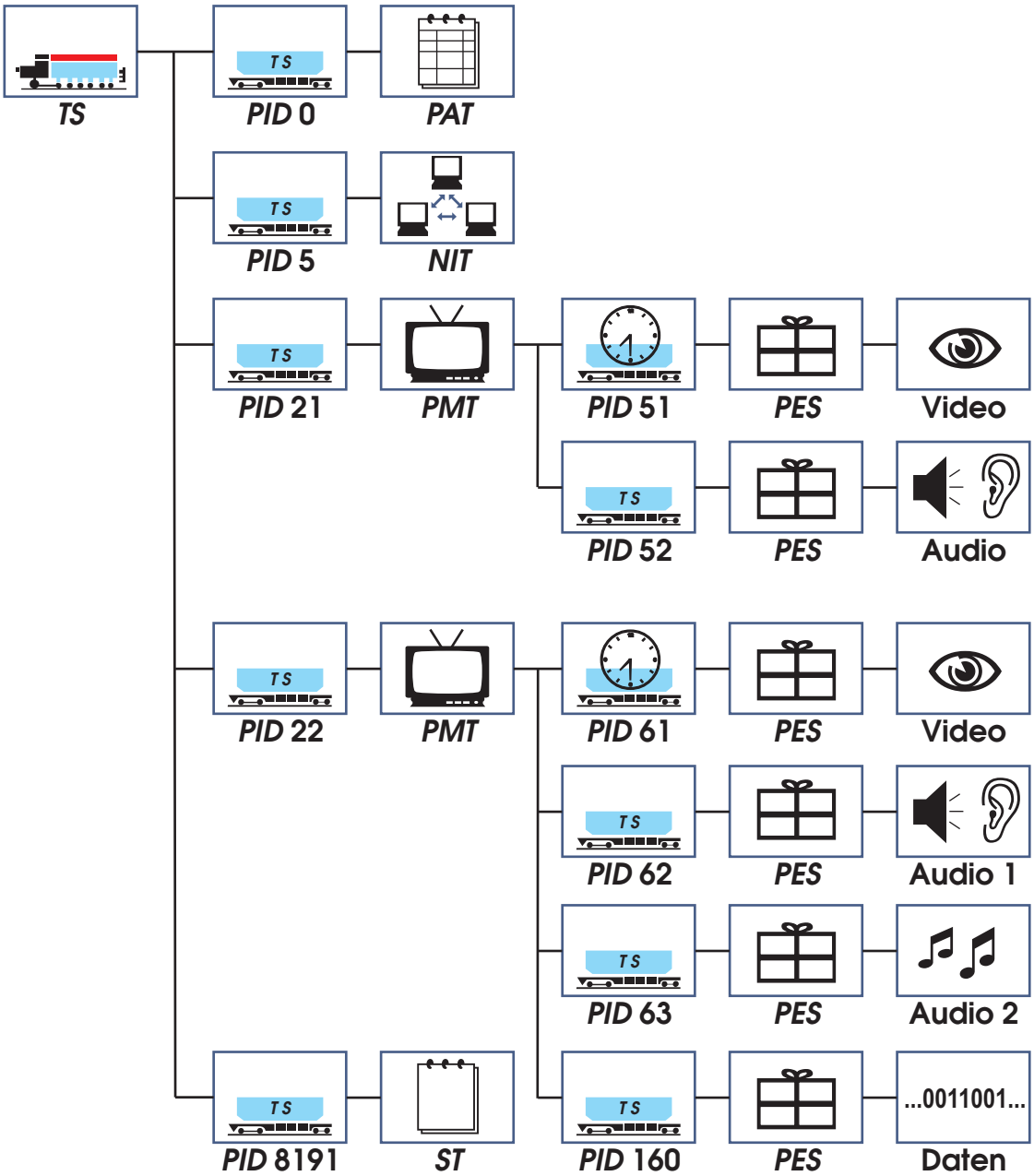


Abb. 6.3-2: Beispiel für einen Transportstrom mit zwei Hauptprogrammströmen

Das Programm, welches mit der PID 22 adressiert wird, besitzt zusätzlich zum Videodatenstrom noch zwei Audiodatenströme sowie einen Datenstrom, der beispielsweise einen Untertitel oder die Teletextinformation beinhalten kann. Mit der PID 8191 wird in diesem Beispiel die Stuffing Table (*Stopf Bits*) adressiert, um für das ganze Bouquet insgesamt einen konstantes Datenaufkommen erreichen zu können.

Selbsttestaufgabe 6.3-1:

Nennen Sie die vier verschiedenen Arten von Tabellen, welche im MPEG-2 Transportstrom definiert sind! Nennen Sie die fünf wichtigsten Service Information Tabellen von DVB!

Selbsttestaufgabe 6.3-2:

Wie viele verschiedene Transportströme lassen sich über die PID – theoretisch – maximal adressieren?

6.4 Kanalcodierung für die digitale TV-Übertragung

Die Quellencodierung, die in den vorangegangenen Kapiteln beschrieben wurde, führt eine Redundanz- und Irrelevanzreduktion durch, und verringert damit die sehr hohe Eingangsdatenrate des digitalen Videosignals. Es ist leicht einzusehen, dass der komprimierte Datenstrom eine höhere Fehlerempfindlichkeit gegenüber Störungen im Übertragungsweg besitzt als der unkomprimierte. So verursachen einzelne Bitfehler nicht mehr nur einzelne Pixelfehler im Bild, sondern können unter Umständen zum Ausfall eines kompletten Bildes führen, wenn die entsprechende sensible Steuerinformation betroffen ist.

FEC – Forward Error Correction

Um eine sichere Übertragung des codierten Videodatenstroms zu erreichen, werden besondere Maßnahmen zum Fehlerschutz ergriffen. Im Rahmen der so genannten *Kanalcodierung* wird zu diesem Zweck dem komprimierten Signal wieder gezielt Redundanzinformation hinzugefügt, die auf eine möglichst effektive Fehlerkorrekturmöglichkeit im Empfänger ausgelegt ist. Da die empfängerseitige Korrektur nur auf die ggf. gestörten Daten Zugriff hat, ist für diese Korrektur der Begriff *Forward Error Correction – FEC* gebräuchlich.

6.4.1 Energieverwischung

Pseudozufallszahlenfolge – PRBS

Betrachtet man das quellcodierte Signal, wird man feststellen, dass das korrespondierende Frequenzspektrum keine gleichmäßige Energieverteilung besitzt. Die Leistung konzentriert sich beispielsweise bei einer einfachen *QPSK*-Modulation (vgl. Abschnitt 6.5.2) in Zeiten von längeren gleichmäßigen Datenfolgen (z. B. Stopfpakete) auf die Trägerfrequenz. Solche Energiekonzentrationen sind unerwünscht, da sie leicht Übersprechstörungen mit benachbarten Sendekanälen verursachen können. Man fügt daher eine so genannte *Energieverwischung* ein, die mit Hilfe einer *Pseudozufallszahlenfolge* (*Pseudo Random Binary Sequence – PRBS*) realisiert wird [6.10].

Modulo-2: Ex-OR

Der Datenstrom wird über eine *Modulo-2* Verrechnung (*Ex-OR*) mit der Zufallsfolge verknüpft (*Scrambling*). Dabei wird die *Pseudozufallsfolge* mit folgender

Generatorpolynom

Generatorpolynom Gleichung gebildet:

$$g_{PRBS,DVB} = 1 + x^{14} + x^{15}; \qquad 6.4\text{-}1$$

Initialisierungsequenz

die zugehörige *Initialisierungssequenz – INIT* lautet:

$$Init_{\text{PRBS,DVB}} = 100101010000000_{\text{BIN}}. \qquad 6.4\text{-}2$$

Praktisch realisiert wird der Pseudozufallszahlengenerator mit der Hilfe eines rückgekoppelten Schieberegisters.

Enable-Signal

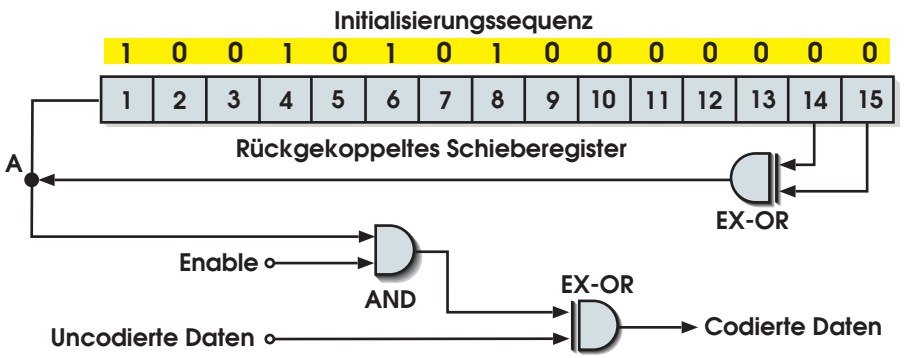
Über das *Enable-Signal* kann die Energieverwischung ein- und ausgeschaltet werden. Damit werden im MPEG-2 Transportdatenstrom die *Sync-Bytes*

$$Sync_{\text{TS}} = 01000111_{\text{BIN}} = 47_{\text{HEX}} \qquad 6.4\text{-}3$$

ohne Beeinflussung gelassen, so dass die Synchronisation auf die 188 Bytes großen Transportstrompakete weiterhin sicher gelingt[50]. Überdies wird der Generator alle acht Transportpakete mit der *INIT*-Sequenz neu initialisiert. Damit der Decoder den Beginn einer Initialisierung erkennen kann, wird das Sync-Byte zu Beginn einer neuen Achter-Sequenz invertiert:

Blockschaltbild Energieverwischung

Die Verwürfelung beginnt dann mit dem ersten Byte nach dem Sync-Byte. Animation 6.4-1 zeigt das logische Blockschaltbild der Energieverwischung, wie sie für das DVB-Konzept eingeführt ist.



Animation 6.4-1: Energieverwischung bei DVB mit Pseudozufallszahlengenerator

50 Man beachte, dass der Zufallszahlengenerator selbst weiter läuft.

6.4.2 Bitfehlerrate

Die Verteilung von DVB-konformen Datenströmen geschieht in der Regel in Übertragungskanälen, die bislang für analoge Fernsehprogramme vorgesehen waren. Die *digitale Modulation* (vgl. Abschnitt 6.5) ermöglicht es, solche Übertragungskanäle als Datencontainer mit definierter Datenkapazität zu verstehen. Bei einem digitalen Signal wird die Qualität der Übertragung nicht mit dem *Signal-Rauschabstand* angegeben, sondern mit der so genannten *Bitfehlerrate* (*Bit Error Rate – BER*), die wie folgt definiert ist:

Bitfehlerrate - BER

$$\text{BER} = \frac{\text{Anzahl falsch erkannter Bits}}{\text{Gesamtzahl gesendeter Bits}}. \qquad 6.4\text{-}5$$

Bei DVB liegen die üblichen Datenraten für ein einzelnes Programm zwischen 4 und 10 Mbit/s. Wie bereits erwähnt werden im Transportmultiplex mehrere Programme gemeinsam in einem Übertragungskanal übertragen. Die Übertragungsqualität gilt allgemein als *„gut“*, wenn die Bitfehlerrate unter normalen Empfangsbedingungen so gering ist, dass nicht mehr als 1 Fehler pro Stunde auftritt. Umgerechnet auf das übertragende Datenvolumen ergibt sich mit Gl. 6.4-5 die Forderung für die maximal tolerierbare *BER* am Ausgang der Fehlerschutzcodierung:

$$\text{BER} \overset{!}{<} 10^{-11}. \qquad 6.4\text{-}6$$

6.4.3 Bitfehlermuster

Bitfehler Burstfehler Symbolfehler

Um die BER aus Gl. 6.4-6 zu erreichen, muss der Fehlerschutz gezielt an den Übertragungskanal und die zu erwartenden Fehlertypen angepasst sein. Die Auswirkungen von Störungen bei der Übertragung auf den Datenstrom verursachen in der Regel unterschiedliche *Bitfehlermuster*. Wird nur ein einzelnes Bit verfälscht, spricht man von einem *Bitfehler*. Ein *n* Bits langer *Burstfehler*[51] ist hingegen definiert durch einen Block mit einer Länge von *n* Bits, bei dem mindestens das erste und das letzte Bit falsch sind. Ein *Symbolfehler* bezeichnet ein verfälschtes *Symbol*[52] der Länge *l*, innerhalb dessen bis zu *l* Bitfehler in beliebiger Zusammensetzung auftreten dürfen. Abb. 6.4-2 veranschaulicht diese Differenzierung der Bitfehlermuster.

51 Der *Burstfehler* wird manchmal auch *Bündelfehler* genannt.

52 Oft entspricht ein Symbol 1 Byte.

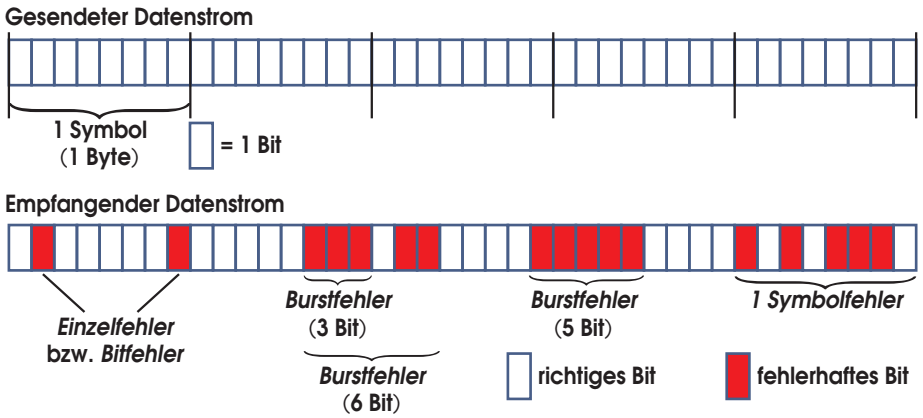


Abb. 6.4-2: Mögliche Bitfehlermuster bei der digitalen Übertragung

6.4.4 Code-Klassen

Blockcodes Faltungscodes

Aus der Literatur sind eine Reihe von *Code-Klassen* bekannt, die sich für den Fehlerschutz eignen (vgl. [6.11]). Die Code-Klassen lassen sich grob in die zwei Gruppen *Blockcodes* und *Faltungscodes* unterteilen, die sich jeweils für unterschiedliche Bitfehlermuster eignen.

Redundanzsymbol

Beim *Blockcode* (vgl. Abschnitt 6.4.3) werden die Eingangsdaten in Blöcke von *m* Symbolen aufgeteilt. Nach jedem Block werden *k Redundanzsymbole* angehängt. Die resultierende Blocklänge eines *Blockcodes* ist also

$$D_{\text{Blockcode}} = n = m + k. \tag{6.4-7}$$

Coderate R

Das Verhältnis der Informationsbits *m* zur Gesamtblocklänge *n* wird als *Coderate R* bezeichnet:

$$R = \frac{\text{Anzahl Informationsbits}}{\text{Anzahl der Informationsbits} + \text{Anzahl der Fehlerschutzsbits}}. \tag{6.4-8}$$

Die Struktur der Blockcodes prädestiniert diese für die Korrektur von Symbolfehlern.

Beim *Faltungscodes* wird das Eingangssignal kontinuierlich in ein Schieberegister eingelesen. Die Ausgangsdaten werden aus der Kombination von verschiedenen Abgriffen am Register erzeugt. Die *Coderate R* ergibt sich hier wieder als Quotient der *m* auf einmal eingelesenen Bits zu den *n* auf einmal ausgelesenen Bits. Durch diese Art der Codierung (*Filterung*) sind *Faltungscodes* gut dazu geeignet, einzelne Bitfehler zu korrigieren.

6.4.5 Codeverkettung

Äußerer und innerer Code

Treten bei der Übertragung verschiedene Bitfehlermuster auf, kann die Effizienz der Fehlerschutzcodierung erhöht werden, indem man Codes aus verschiedenen Code-Klassen miteinander verkettet. Der erste Code, Coderate $R1$ wird *äußerer*, der zweite mit der Coderate R_2 wird *innerer Code* genannt. Bei einer zu übertragenden Datenrate R_u ergibt sich die Gesamtdatenrate R_{ges} zu

$$R_{ges} = \frac{R_u}{R_1 \cdot R_2}. \qquad \text{6.4-9}$$

Codeverkettung

Abb. 6.4-3 zeigt das Blockschaltbild für eine Codeverkettung mit zwei Codes.

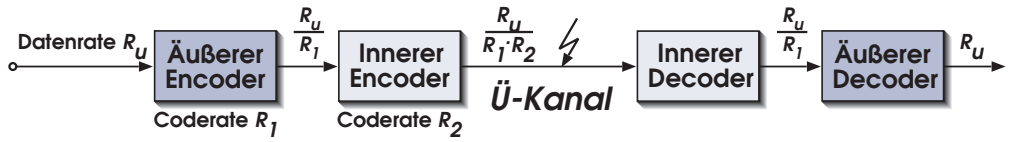


Abb. 6.4-3: Codeverkettung mit den zugehörigen Datenraten

Bei DVB wird üblicherweise für den äußeren Code ein Blockcode und für den inneren ein Faltungscode verwendet. Mit dieser Codeverkettung gelingt es *einzelne Bitfehler* und *kleinere Burstfehler* zu korrigieren.

6.4.6 Faltungsinterleaver

Faltungsinterleaver

Sollen auch *längere Burstfehler* korrigiert werden, wird zwischen äußerem und inneren Code ein so genannter *Faltungsinterleaver* eingefügt. Die Aufgabe des *Interleavers* besteht darin, den Datenstrom in unterschiedlich große Teilabschnitte neu zu verschachteln. Im Ergebnis liegen zwei ursprünglich benachbarte Bits um die so genannte *Interleavingtiefe I* auseinander. Es gilt zu beachten, dass der *Interleaver* selbst keine neuen, redundanten Daten hinzufügt, sondern lediglich für die Umsortierung der Daten verantwortlich ist.

Faltungsinterleaver nach Forney

Bei DVB wird der so genannte *Faltungsinterleaver nach Forney* [6.12] verwendet. Er besteht aus *I-1* Schieberegister, die über je einen Multiplexer und Demultiplexer mit dem Eingang bzw. Ausgang verbunden sind. Die Längen der einzelnen Schieberegister sind unterschiedlich und reichen von *M* bis (*I-1*) *M* Bytes, mit *M* als die so genannte *Basisverzögerung*, für die gilt:

Basisverzögerung

$$M = \frac{n}{I}, \qquad \text{6.4-10}$$

wobei *n* die *Rahmenlänge* des Blockcodes (äußerer Code, vgl. Abschnitt 6.4.4) ist. Abb. 6.4-4 zeigt den Sachverhalt mit den für DVB üblichen Zahlenwerten:

$$M = 17; \quad I = 12. \qquad \text{6.4-11}$$

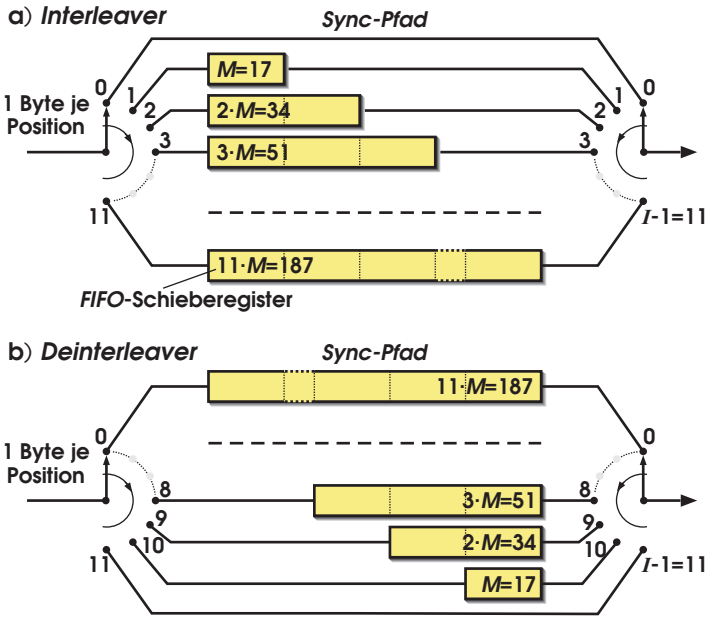


Abb. 6.4-4: Faltungsinterleaver bei DVB: M = 17, I = 12

Mit jedem Schritt wird jeweils ein Byte in das aktuelle Schieberegister eingelesen sowie ein Byte auf der anderen Seite aus dem aktuellen Schieberegister ausgelesen. Danach schaltet der Multiplexprozess auf das nächste Schieberegister. Der oberste Pfad (*Sync-Pfad*) dient dazu, das *Sync-Byte* des Transportdatenstroms ohne Verzögerung weiterzuleiten. Die Auslegung des Faltungsinterleavers sorgt dafür, dass das Sync-Byte immer durch den obersten Pfad geleitet wird, und damit sicher erkannt werden kann.

Der Faltungsinterleaver gewährleistet eine Trennung von benachbarten Symbolen durch die Einfügung von $M\,I$ Symbole. Der *Deinterleaver* (Abb. 6.4-4) ist entsprechend umgekehrt aufgebaut. Die resultierende *Gesamtverzögerung* ist damit für alle Symbole gleich, nämlich:

$$D_{\text{Forney}} = M \cdot (I-1) \cdot I. \qquad \text{6.4-12}$$

Mit den Werten für DVB (vgl. Gl. 6.4-11) ergibt sich eine Gesamtverzögerung von

$$D_{\text{Forney,DVB}} = 2244\ Bytes. \qquad \text{6.4-13}$$

6.4.7 Reed-Solomon Code bei DVB

Für den äußeren Coder wird bei DVB ein so genannter *Reed-Solomon Code* eingesetzt. Reed-Solomon Codes sind symbolorientierte *Blockcodes*, die sich gut für die Erkennung und die Korrektur von *Burst-* und *Symbolfehlern* eignen. Die Berechnung des Reed-Solomon Codes ist recht komplex und soll hier nicht weiter skizziert werden. Weiterführende Details lassen sich beispielsweise in [6.7] finden. Wie bereits beschrieben, werden bei einem Blockcode zu *m Informationsbytes k Korrekturbytes* hinzugefügt.

Ein Reed-Solomon Code mit *m* Informationsbytes und *k* Korrekturbytes wird ein *RS* (*n*, *m*) Code genannt, mit *n* als *Rahmenlänge*:

$$n = m + k. \qquad 6.4\text{-}14$$

Ein derartiger Code kann maximal *k/2* falsche Symbole (Bytes) pro Codewort korrigieren und *k* erkennen. Die Coderate des Blockcodes errechnet sich nach Gl. 6.4-8 zu

$$R_{\text{Blockcode}} = \frac{m}{n} = 1 - \frac{k}{n}\ . \qquad 6.4\text{-}15$$

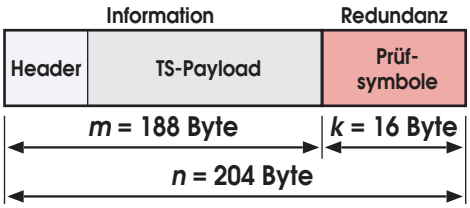


Abb. 6.4-5: *RS* (*204,188*) Blockcode bei DVB

RS-Blockcode bei DVB

Bei DVB kommt eine *RS* (*204,188*) zur Anwendung. Erkennbar erstreckt sich der Codeblock genau über ein MPEG-2 Transportpaket von 188 Byte. Es existieren 16 Korrekturbytes, mit denen maximal 8 falsche Bytes im Transportpaket korrigiert werden können. Die Redundanz des *RS*(*204,188*) Codes ist mit *k* = 8% relativ gering. Die Coderate für den *RS*(*204,188*) errechnet sich zu:

$$R_{\text{RS(204,118)}} = \frac{188}{204} \approx 0.9216. \qquad 6.4\text{-}16$$

Abb. 6.4-5 veranschaulicht den Aufbau des Blockcodes für die Verhältnisse bei DVB (*m* = 188, *n* = 204).

6.4.8 Faltungscode bei DVB

Beim Faltungscode werden die Informationsbits des Eingangsdatenstroms kontinuierlich in ein Schieberegister eingelesen. Es werden zwei Ausgangsströme gebildet, indem zweimal das aktuelle Eingangsbit mit einer bestimmten Anzahl von vorangegangenen Bits an den Anzapfungen des Schieberegisters über eine Modulo-2 Verknüpfung (s. o.) miteinander verrechnet werden. Abb. 6.4-6 zeigt das Blockschaltbild für den Faltungscoder, wie er beim DVB-Standard verwendet wird.

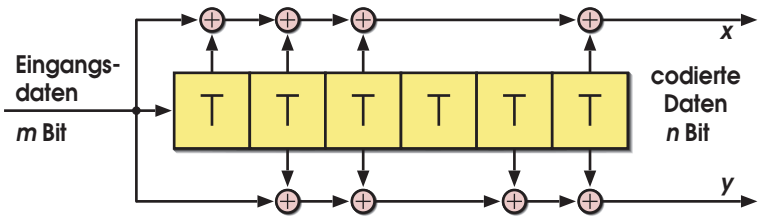


Abb. 6.4-6: Faltungscoder für den DVB-Standard

Eingangsrahmenbreite
Ausgangsrahmenbreite

In diesem Beispiel beträgt die so genannte *Eingangsrahmenbreite*[53] *m* = 1 Bit, die entsprechende *Ausgangsrahmenbreite*[54] *n* = 2 Bit. Für die Coderate *R* ergibt sich daraus ein Wert von

$$R_{\text{Faltungscode}} = \frac{m}{n} = \frac{1}{2}. \qquad \text{6.4-17}$$

Beeinflussungslänge

Aus der Länge *s* des Schieberegisters lässt sich die so genannte *Beeinflussungslänge K*

$$K = (s+1) \cdot m = 7, s = 6, m = 1 \qquad \text{6.4-18}$$

Generatorpolynom

errechnen. Die Anordnungen der Abgriffe beim Schieberegister werden häufig durch *Generatorpolynome G* beschrieben, die sich auch als *Oktalzahl* angegeben lassen. Im vorliegenden Fall (Abb. 6.4-6) ergibt sich für die beiden Generatorpolynome $G_1 = G_{DVB,x}$ und $G_2 = G_{DVB,y}$:

$$G_{DVB,x} = G_1 = 1111001_{\text{BIN}} = 171_{\text{OCT}} \qquad \text{6.4-19}$$

$$G_{DVB,y} = G_2 = 1011011_{\text{BIN}} = 133_{\text{OCT}}\ . \qquad \text{6.4-20}$$

Parameter Faltungscode bei DVB

Die nachfolgende Tabelle 6.4-1 fasst noch einmal die wichtigsten Parameter des Faltungscodes für das DVB-System zusammen.

---

53 Die *Eingangsrahmenbreite* entspricht der Anzahl der parallel zugeführten Datenströme am Eingang.

54 Die *Ausgangsrahmenbreite* entspricht der Anzahl der parallelen Datenströme, die am Ausgang abgegriffen werden.

Tab. 6.4-1: *Parameter des Faltungscodes bei DVB*

Parameter	Bezeichnung/ Formel	Wert bei DVB
Eingangsrahmenbreite	*m*	1
Ausgangsrahmenbreite	*n*	2
Coderate	$R = \frac{m}{n}$	1/2
Gedächtnis	*s * m*	6
Mögliche Zustände	$2^{s \cdot m}$	64
Beeinflussungslänge	*K = (s+1)*m*	7
Generatorpolynom 1	G_1	$x^6 + x^3 + x^2 + x + 1 = 171_{\text{OKT}}$
Generatorpolynom 2	G_2	$x^6 + x^5 + x^3 + x^2 + 1 = 133_{\text{OKT}}$

6.4.9 Punktierung von Faltungscodes

Die Coderate des Faltungscodes ist mit einem Wert von *R* = 1/2 relativ gering. Es besteht jedoch prinzipiell die Möglichkeit, aus den beiden Ausgangsströmen *x* und *y* wieder einzelne Bits nach einem bestimmten Muster zu entfernen. Damit verringert sich die Menge der zu übertragenden Daten und ebenfalls die Redundanz im Signal. Ersichtlich wird auch die Korrekturfähigkeit des Codes etwas verschlechtert. Der Vorteil liegt jedoch darin, dass die Anzahl der zu streichenden Bits am Ausgang eine variable Größe darstellt, mit der man die Brutto-Datenrate einstellen kann und auch die Korrekturfähigkeit in gewissen Grenzen an die Anforderungen anpassen kann. Dieser Vorgang wird *Punktierung* genannt. Abb. 6.4-7 zeigt, wie die Punktierung am Ausgang des Faltungscodes angekoppelt wird.

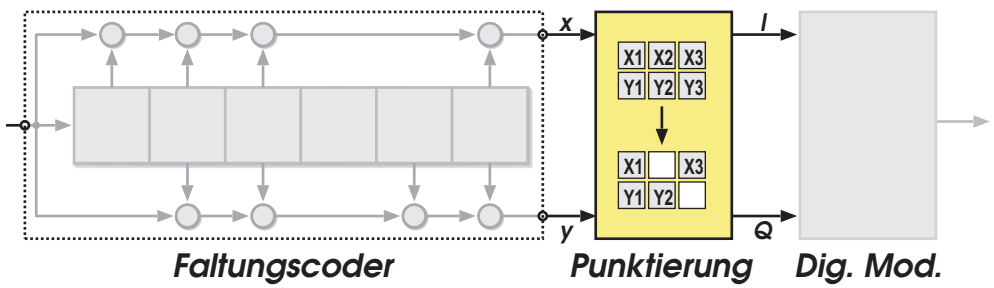


Abb. 6.4-7: Parameter des Faltungscodes bei DVB

Der DVB-Standard sieht fünf verschiedene Punktierungen vor. Aus den beiden Datenströmen *x* und *y* werden zunächst bestimmte Bits ausgeblendet (*punktiert*). Anschließend werden die verbleibenden Werte neu angeordnet, damit wieder beide Ausgänge *I* und *Q* gleichmäßige Datenströme hervorbringen. Abb. 6.4-8 zeigt die Punktierungen und die anschließende Umsortierung nach dem DVB-Standard.

Punktierungen bei DVB

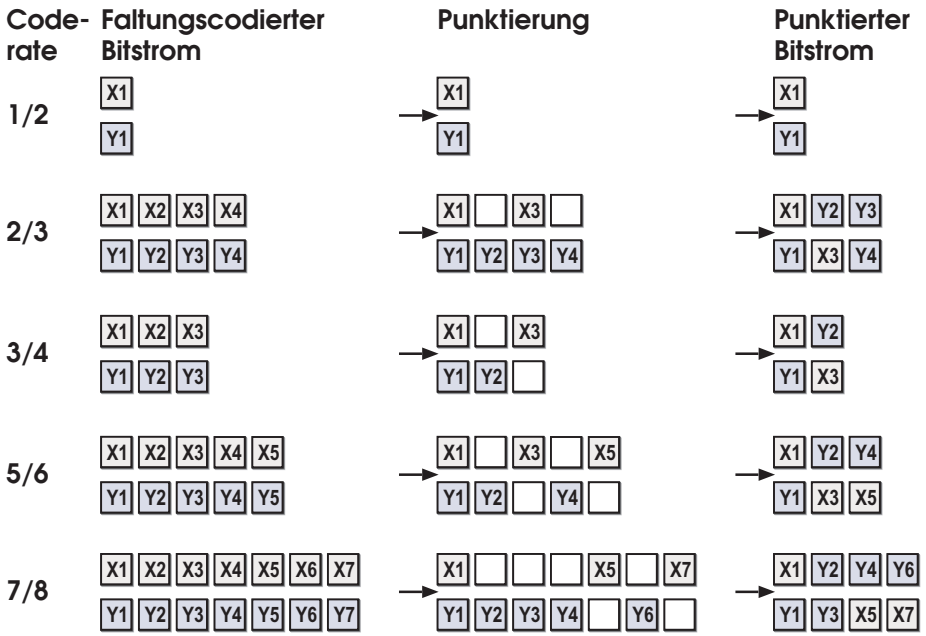


Abb. 6.4-8: Punktierung und Umsortierung beim DVB-Standard

Viterbi-Decoder Trellis-Diagramm

Auf der Decoderseite wird das mit dem Faltungscode verarbeitete Signal meist mit einem so genannten *Viterbi-Decoder* ausgewertet. Empfangsseitig wird mit Hilfe von so genannten *Trellis-Diagrammen* aus den aufeinander folgenden Bit-Kombinationen der beiden Datenströmen sukzessiv die wahrscheinlichste Route ermittelt, aus der sich der Ursprungsdatenstrom zurückrechnen lässt.

Hard-, Softdecision

Da bei der Übertragung in einem realen Kanal das Signal oftmals gestört wird, sind am Empfänger die Amplituden der digitalen Signale nicht mehr diskret, sondern über den gesamten Bereich verstreut. Um dem *Viterbi-Decoder* diskrete Werte anbieten zu können wird eine so genannte *Hard-* oder *Softdecision* vorgeschaltet, die aus den stochastischen Signalen Zuordnungen für x_i und $y_i$über Entscheidungsschwellen vornimmt.

Die Berechnungen im Viterbi-Decoder sind relativ komplex und können hier nicht weiter behandelt werden. Weiterführende Einzelheiten lassen sich in [6.7] finden.

6.5 DVB-Übertragungstechniken

6.5.1 Signalvorbereitung

DVB wird in der Regel über analoge, bandbegrenzter Übertragungskanäle verbreitet, die auch für das analoge Fernsehen genutzt werden. Das korrespondierende Frequenzspektrum einer binären Datenfolge hat theoretisch eine unendliche Ausdehnung. Da das Betragsspektrum jedoch mit 1/*f* zu höhere Frequenzen abnimmt, ist davon auszugehen, dass das für die Auswertbarkeit des Signals erforderliche Frequenzspektrum endlich ist. Eine geeignete Bandbegrenzung erscheint also zweckmäßig.

Filter mit Roll-Off-Flanke

Nach dem ersten *Nyquistkriterium* ist eine eindeutige Erkennung der übertragen-

den Symbolfolge möglich, wenn wenigstens die doppelte Frequenz der ersten harmonischen Grundfrequenz für den schnellstmöglichen Zustandwechsel (101010... Bitfolge) gewählt wird. Um die *Verzerrungen* des Signals im Zeitbereich möglichst gering zu halten hat sich die *Impulsformung* bzw. *Basisbandfilterung* durch ein so genanntes $cos^2 - Filter$ bewährt (auch: *Filter mit Roll-Off-Flanke* oder *Nyquistfilter*). Die Besonderheit der Filterflanke liegt in der Symmetrie zur Grenzfrequenz fg: Das Maß für die Steilheit des Kurvenverlauf des Filters im Frequenzbereich wird mit dem so genannten *Roll-Off-Faktor r*

Roll-Off-Faktor

$$r = \frac{f_{\max} - f_{\min}}{f_{\max} + f_{\min}} = \frac{\Delta f}{f_g} \qquad 6.5\text{-}1$$

ausgedrückt. Abb. 6.5-1 skizziert die *cos*^{2}*-Roll-Off-Filterung* im Frequenzbereich.

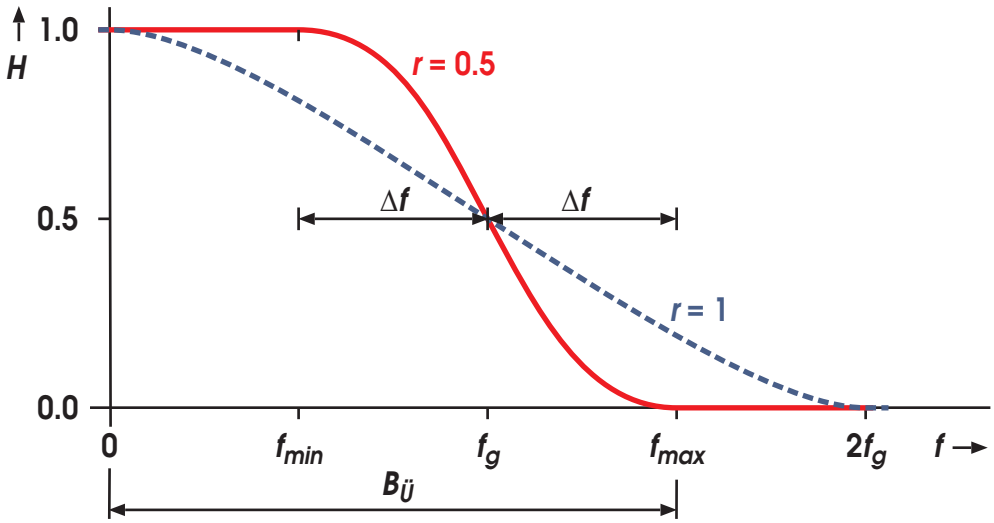


Abb. 6.5-1: *cos*^{2}*-Roll-Off* Filterflanke mit *r* = 1 und *r* = 0.5

Die notwendige Übertragungsbandbreite $f_{max} = B_{\text{Ü}}$ und $f_g = B_N$für das bandbegrenzte Spektrum ergibt sich zu

$$B_{\text{Ü}} = B_N \cdot (1 + r), \qquad 6.5\text{-}2$$

wobei der Roll-Off-Faktor *r* in einem Bereich von

$$0 \leq r \leq 1 \qquad 6.5\text{-}3$$

liegt. Mit *r* lässt sich so in gewissen Grenzen ein Kompromiss bezüglich *Bandbreitenbedarf* und *Verzerrung des Signals* erreichen. Von dem Grad der Verzerrung hängt die Auswertbarkeit des Signals entscheidend ab. Eine gebräuchliche Darstellung zur Auswertbarkeit des verzerrten Datensignals liefert das so genannte *Augendiagramm*. Man erhält dieses durch laufendes Übereinanderschreiben des Signalverlaufs während der Bitdauer T_{Bit}mit Hilfe eines Oszilloskops, das mit dem Bit-Takt getriggert wird. Abb. 6.5-2 zeigt ein Beispiel für ein Augendiagramm mit einem *Roll-Off-Faktor* von *r* = 0.5.

Augendiagramm

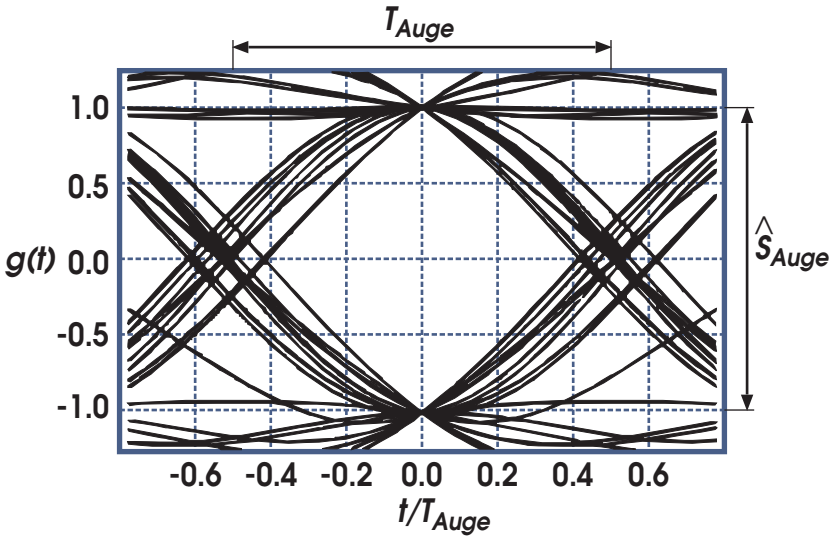


Abb. 6.5-2: Beispiel für ein Augendiagramm (r = 0.5)

Augenöffnungen

Bei der Auswertbarkeit wird zwischen der *vertikalen* und der *horizontalen Augenöffnung* unterschieden. Je größer die relative, vertikale Augenöffnung ist, umso eindeutiger kann eine Schwellenentscheidung („*1*“ oder „*0*“) getroffen werden. Eine möglichst große horizontale Augenöffnung ist wichtig, um eine *Schwankungsreserve* für den Abtasttakt zu haben.

6.5.2 Digitale Modulation für Satellitenkanäle

Digitale Modulation

Verfahren der *digitalen Modulation* werden eingesetzt, um die digitalen Datenströme in vorgegebene Funk-, Kabel- oder Satellitenkanälen zu übertragen. In Abhängigkeit des in der Signalform vorgefilterten digitalen Datenstroms wird eine Änderung der *Amplitude*, der *Frequenz* oder der *Phase* eines Trägersignals vorgenommen. Da nur eine diskrete Anzahl von Zuständen erreicht wird, spricht man auch von *Umtastung*. Drei verschiedene Grundformen der digitalen Modulation werden unterschieden:

Grundformen digitaler Modulationen

- *Amplitudenumtastung – Amplitude Shift Keying* (*ASK*),
- *Frequenzumtastung – Frequency Shift Keying* (*FSK*),
- *Phasenumtastung – Phase Shift Keying* (*PSK*).

DVB-S: Q-PSK

Für die DVB-Übertragung via Satellit – *DVB-S*[55] wird die so genannte *Q-PSK* gewählt. In Abhängigkeit von den zwei binären Signalen *I* und *Q*[56] werden 4 verschiedene *Phasenzustände* generiert. Abb. 6.5-3 zeigt das prinzipielle Blockschaltbild der Q-PSK.

55 Siehe Abschnitt 6.5.3

56 *I* und *Q* sind die beiden Datenströme, die der in Abschnitt 6.4.8 beschriebene Faltungscoder ausgibt. Diese werden über die in Abschnitt 6.5.1 beschriebene Signalvorbereitung über ein cos^2-Roll-Off-Filter (r = 0,35) weiter verarbeitet.

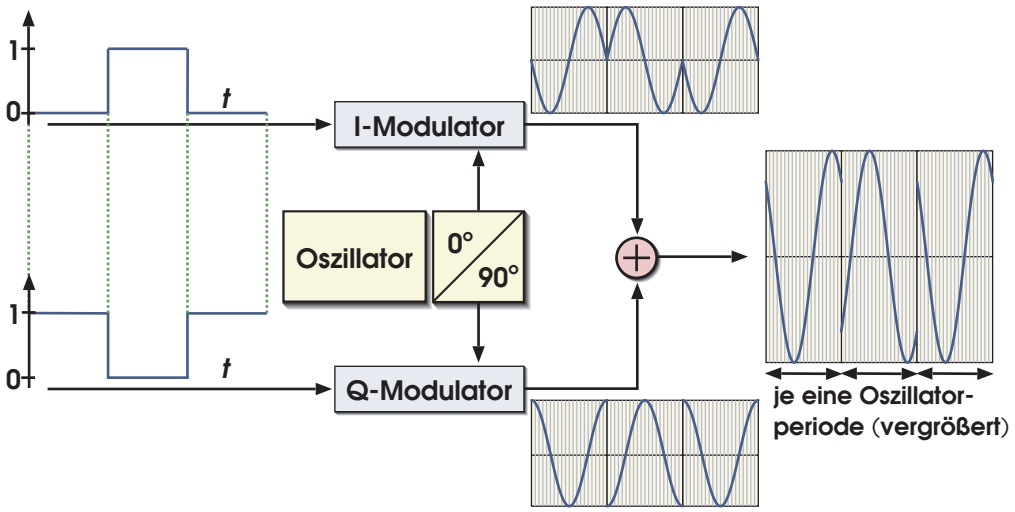


Abb. 6.5-3: Gewinnung des Q-PSK-Signals

Der Vorteil der *Q-PSK* liegt in seiner Unempfindlichkeit gegenüber Amplitudenschwankungen, wie sie in Satellitenkanälen auftreten. Weiterhin fordert der geringe Störabstand auf Empfängerseite ein besonders sicheres Modulationsverfahren. Eine höherwertige PSK (z. B. *8-PSK* oder *16-PSK*) oder eine *QAM* (siehe Abschnitt 6.5.4) hätte zwar eine bessere Bandbreiteneffizienz, wäre aber auch entsprechend störanfälliger. Ein Satellitenkanal stellt eine relativ große Bandbreite zur Verfügung und kann damit den schlechten Störabstand ausgleichen.

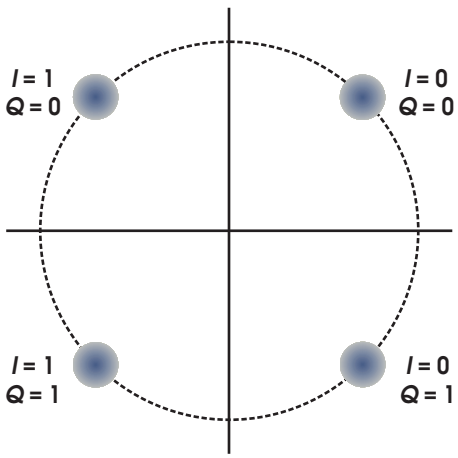


Abb. 6.5-4: Phasenzustandsdiagramm der Q-PSK

Phasenzustandsdiagramm Q-PSK

Die Phasenzustände von digitalen Modulationen werden in so genannten *Phasenzustandsdiagrammen* dargestellt. Ein solches ist für die Q-PSK Modulation in Abb. 6.5-4 skizziert. Da bei der Q-PSK Modulation 2 Bit pro Symbol verarbeitet werden, beträgt die *Bandbreitenausnutzung B* im Idealfall

Bandbreitenausnutzung

$$B_{\text{Q-PSK}} = 2\frac{\text{Bit}s \cdot Hz}{.} \qquad 6.5\text{-}4$$

Die Basisbandfilterung (r_{DVB-S} = 0.35) und andere Faktoren reduzieren allerdings das tatsächliche Verhältnis zwischen *Bandbreite* $B_{\ddot{U}}$ und $Symbolrate R_S$. In der Praxis wird hierfür ein Wert von

$$\frac{B_{\ddot{U}}}{R_S} = 1.27 \frac{Hz}{Symbole/s} \qquad 6.5\text{-}5$$

angegeben [6.7]. Das reduziert die *Bandbreitenausnutzung B* auf

$$B_{\text{Q - PSK}} = \frac{Bits\ pro\ Symbol}{\frac{B_{\ddot{U}}}{R_S}} = \frac{2\,Bit/Symbol}{1.27 \frac{Hz}{Symbole/s}} = 1.57 \frac{Bit}{s \cdot Hz}. \qquad 6.5\text{-}6$$

Unter der Berücksichtigung der Codeverkettung lässt sich näherungsweise bei gegebener Kanalbandbreite $B_{\ddot{U}}$ aus Gl. 6.4-9 und Gl. 6.5-6 die erzielbare Netto-Datenrate R_u für DVB-S errechnen.

6.5.3 DVB-S Übertragungskonzept

Die Parameter des europäischen digitalen Satellitenfernsehens (*DVB-S*) sind 1995 in der schon zitierten ETSI-Norm *EN 300 421* festgelegt worden [6.10]. Abb. 6.5-5 zeigt das vereinfachte Blockschaltbild des DVB-S Encoders.

DVB-S Encoder

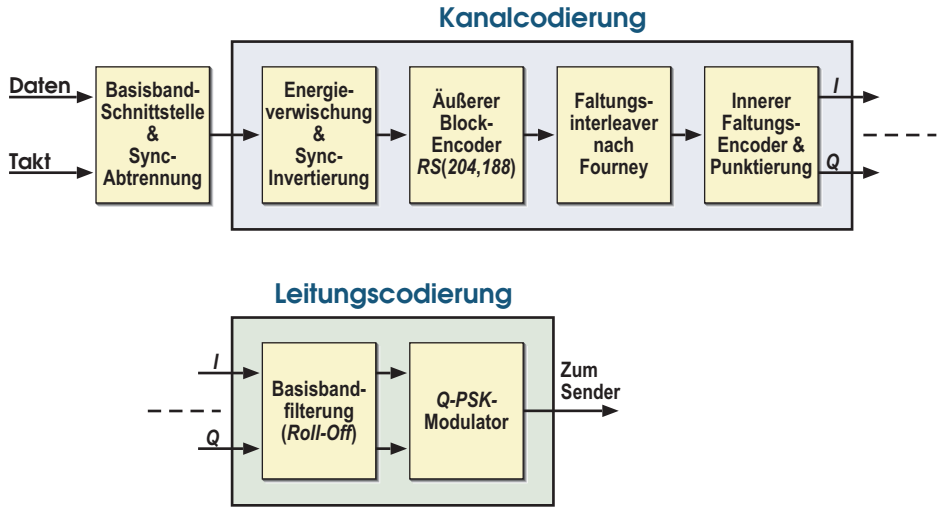


Abb. 6.5-5: Blockschaltbild des DVB-S Encoders (ohne Quellencodierung)

Das Übertragungskonzept beinhaltet im Bereich der Kanalcodierung die Energieverwischung und die verketteten Komponenten zur Fehlerschutzcodierung (*RS*(*204,188*) Blockcode, Faltungsinterleaver, Faltungscode mit einstellbaren Coderaten nach Abb. 6.4-8 über die Punktierung). Die Signalaufbereitung (*r* = 0.35) und die Q-PSK-Modulation gehören zur Leitungscodierung.

DVB-S Decoder

Im Empfänger müssen die Schritte in umgekehrter Reihenfolge erfolgen (Abb. 6.5-6). Wichtig für die korrekte Decodierung ist hier eine Rückkopplung aus der Fehlerschutzdecodierung zurück zur Leitungsdecodierung, um die Referenzphase für die Q-PSK Demodulation laufend korrigieren zu können.

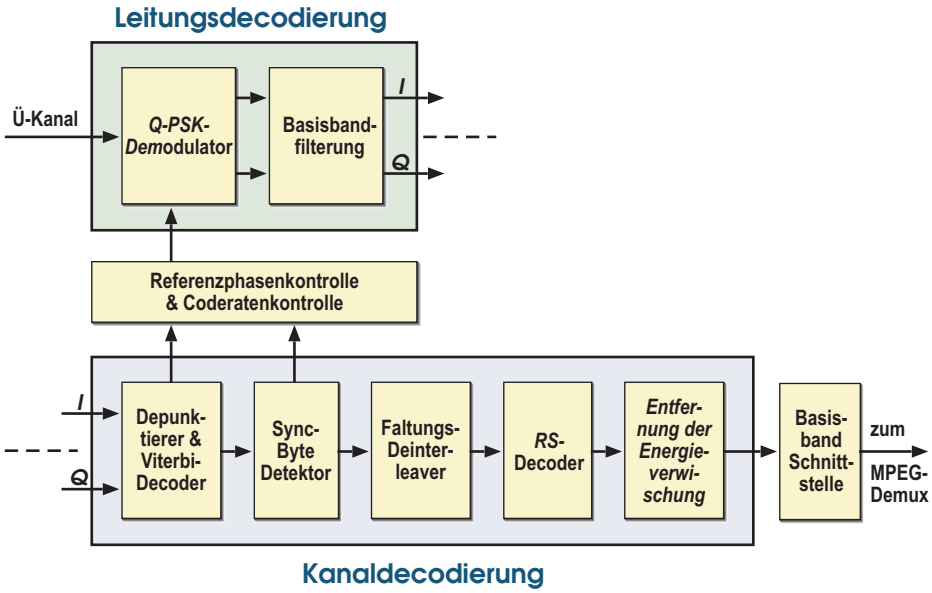


Abb. 6.5-6: Blockschaltbild des DVB-S Decoders (ohne Quellendecodierung)

6.5.4 Digitale Modulation für Breitbandkabelkanäle

DVB-C

Die Übertragungsverhältnisse zwischen Satelliten- und Breitbandkabelkanal unterscheiden sich in vielen Parametern erheblich. Daher wird für die Übertragung in Breitbandkabelkanälen eine veränderte Kanal- und Leitungscodierung angewendet. In der ETSI Norm *EN 300 429*[6.13] sind die Einzelheiten für diesen als *DVB-C* (*C – Cable*) bezeichneten Standard festgehalten.

Bei der digitalen Modulation kann ein Verfahren eingesetzt werden, das eine höhere Bandbreiteneffizienz aufweist, da im Breitbandkabel bessere Störabstände vorherrschen. Auf der anderen Seite steht aufgrund des festen Rasters von meist 8 MHz eine geringere Bandbreite pro Kanal zur Verfügung als beim Satellitenstandard DVB-S. Ein kleiner Teil der Bandbreiteneinsparung wird in der Basisbandfilterung erzielt. Der *cos*^{2}*-Roll-Off-Faktor r* wird reduziert auf

$$r_{\text{DVB-C}} = 0.15. \qquad 6.5\text{-}7$$

QAM

Es wird eine digitale Modulation mit den kombinierten Grundformen *ASK* und *PSK* verwendet. Ähnlich wie bei der *Q-PSK* werden zwei Signale *I* und *Q* in Quadratur moduliert. Diese Modulation wird *Quadratur-Amplituden-Modulation –QAM*[57] genannt. Anders als bei der *PSK* nehmen *I* und *Q* jedoch bei der *QAM* verschiedene diskrete Pegelzustände an.

m-Tuple Conversion

Die Bit-Zuordnung für diese diskreten Pegel lassen sich über einen *Seriell-Parallel-*

57 Korrekt ist eigentlich die Bezeichnung *Quadratur-Amplituden-Phasenumtastung - QAPSK*. Diese Bezeichnung drückt im Namen deutlicher die Kombination aus Amplituden- und Phasenumtastung aus. In der Literatur ist jedoch die Bezeichnung *m-QAM* üblich, wobei m die Zahl einer 2er-Potenz darstellt. Die hier verwendete Modulation darf nicht mit der gleichnamigen analogen *Quadratur-Amplituden-Modulation – QAM* verwechselt werden.

Wandler aus dem Datenstrom erzeugen. Dieser Vorgang wird auch *m-Tuple Conversion* genannt. Die Anzahl der Pegelzustände ist gleichzeitig der Divisionsfaktor für den neuen Takt, mit dem zwei *DA*-Wandler die mehrpegeligen Analogsignale erzeugen. Auch diese Stufensignale benötigen wieder eine bandbegrenzende Basisbandfilterung bevor sie in Quadratur moduliert werden können. Abb. 6.5-7 zeigt das Blockschaltbild für die Erzeugung eines *16-QAM* Signals.

Erzeugung 16-QAM

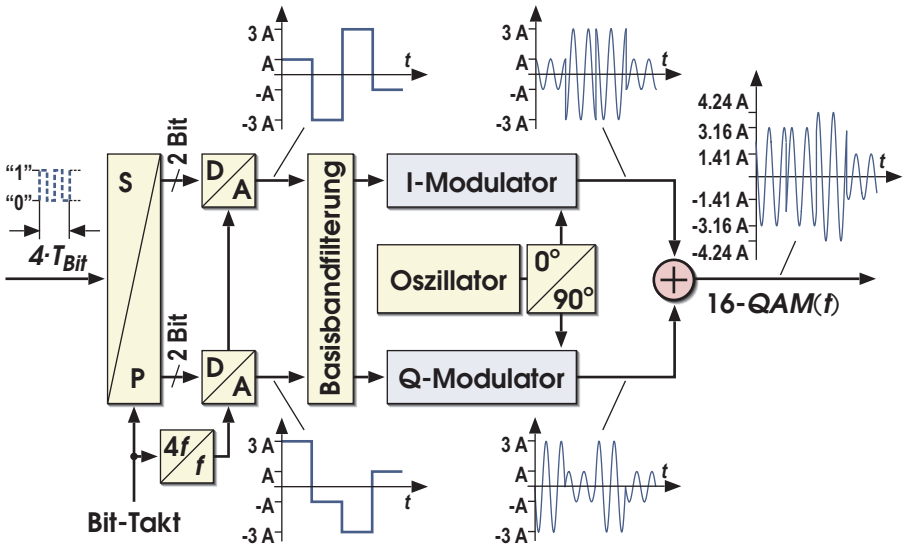


Abb. 6.5-7: Erzeugung eines 16-QAM Signals

Die Anzahl der möglichen Pegel- und Phasenzustände von *I* und *Q* hängt von der Pegel- und Phasenstabilität des Kabelkanals ab. Systembegrenzende Störsignale sind *Intermodulations-Produkte*, die meistens aufgrund nichtlinearer Signalverarbeitung (bei der Verstärkung und Filterung) durch gegenseitige Wechselwirkung verschiedener Kanäle entstehen.

Je mehr Zustände zugelassen sind, umso kleiner werden auch die Entscheidungsbereiche für den Decoder. Im Phasenzustandsdiagramm (Abb. 6.5-8) ist dieser Zusammenhang am Beispiel einer 16-QAM verdeutlicht. An den Amplituden- und Phasenzuständen sind die zugeordneten Codeworte des Seriell-Parallelwandlers eingetragen. Man beachte, dass die ersten beiden Bits mit einer *Phasendifferenzcodierung* übertragen werden [6.13].

Phasenzustandsdiagramm 16-QAM

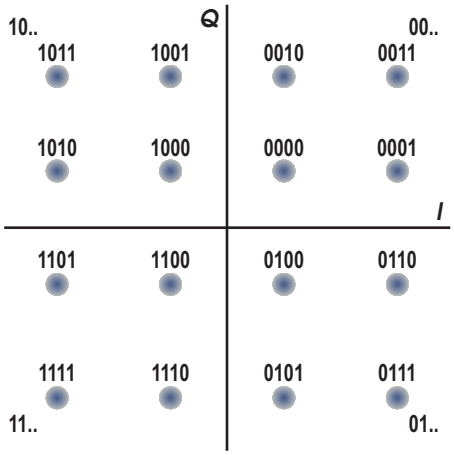


Abb. 6.5-8: Phasenzustandsdiagramm für die 16-QAM

In der Praxis ist in Breitbandkabelkanälen das 64-QAM Verfahren im Einsatz. Mitunter werden auch die *16-QAM* oder *32-QAM* verwendet. In besonderen Fällen sind die *128-QAM* und sogar *256-QAM* einsetzbar[58]. In Analogie zur Q-PSK (Gl. 6.5-4) errechnet sich die ideal ermittelte *Bandbreitenausnutzung B* für die 16-QAM zu:

Bandbreitenausnutzung

$$B_{16-QAM} = 4\,\frac{Bit}{s \cdot Hz}. \qquad 6.5\text{-}8$$

Bei der 64-QAM werden 6 aufeinander folgende Bits zu einem Symbol zusammengefasst. Damit ergibt sich hierfür eine theoretische *Bandbreitenausnutzung B* von

$$B_{64-QAM} = 6\,\frac{Bit}{s \cdot Hz}. \qquad 6.5\text{-}9$$

Auch hier reduziert die Basisbandfilterung (siehe Gl. 6.5-7) das tatsächliche Verhältnis zwischen *Bandbreite* $B_{\ddot{U}}$ und *Symbolrate* R_S. Werden weitere Einflüsse außer Acht gelassen ergibt sich

$$\frac{B_{\ddot{U}}}{R_S} = 1.15\,\frac{Hz}{Symbole/s}. \qquad 6.5\text{-}10$$

Für die 64-QAM reduziert sich damit die *Bandbreitenausnutzung B* auf

$$B_{64-QAM} = \frac{Bits\ pro\ Symbol}{\frac{B_{\ddot{U}}}{R_S}} = \frac{6\,Bit/Symbol}{1.15\,\frac{Hz}{Symbole/s}} = 5.22\,\frac{Bit}{s \cdot Hz}. \qquad 6.5\text{-}11$$

58 Für diese Fälle kann ein zusätzlicher innerer Fehlerschutz wie bei DVB-S angeraten sein.

6.5.5 DVB-C Übertragungskonzept

Im Breitbandkabel steht das für die analoge TV-Verbreitung genutzte Spektrum von 45 MHz bis 862 MHz zur Verfügung. Die Bandbreite der meisten Kanäle beträgt 8 MHz. Es stehen gelegentlich auch 7 MHz und teilweise 12 MHz Kanäle zur Verfügung. Damit die bestehende analoge Infrastruktur der Kabelnetze weiter verwendet werden kann, darf das Raster und die Bandbreite der Kanäle nicht verändert werden. Überdies muss sichergestellt werden, dass sich analoge und digitale Signale auf unterschiedlichen Kanälen nicht gegenseitig beeinflussen (*Intermodulation*).

Obwohl sich Kabel- und Satellitenkanäle in vielen Parametern von einander unterscheiden, ist aus ökonomischer Sicht anzustreben, viele Komponenten der Kanal- und Leitungscodierung identisch zu halten. Die Abbildung der übertragenden Daten eines Satellitenkanals soll identisch auch auf einen Kabelkanal möglich sein. Damit kann gewährleistet werden, dass alle Programme eines MPEG-2 Bouquets auch im Kabelkanal einen gemeinsamen Datenmultiplex bilden.

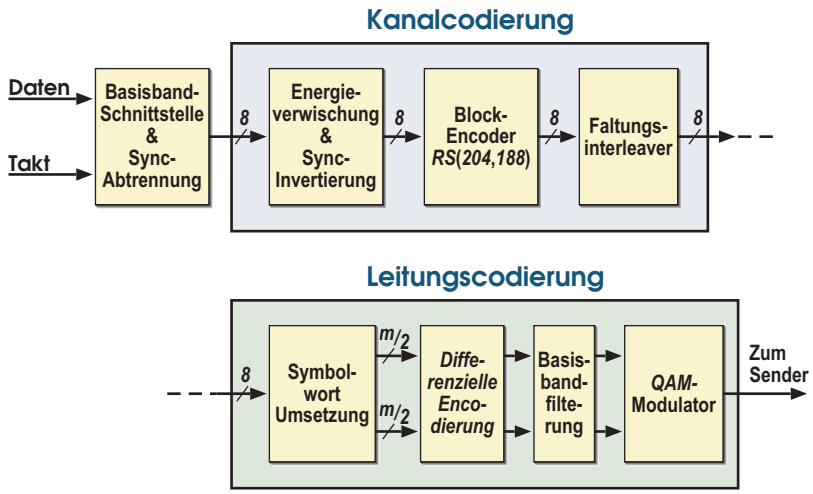


Abb. 6.5-9: Blockschaltbild des DVB-C Encoders

DVB-C Encoder

Praktisch wird dieses Konzept so umgesetzt, indem der äußere Fehlerschutz (RS-Blockcode) und der Faltungsinterleaver bei der Kanalcodierung erhalten bleiben. Der innere Fehlerschutz (Faltungscode) entfällt ersatzlos. Für die digitale Modulation wird vorzugsweise die im letzten Kapitel beschriebene 64-QAM verwendet. Diese wird jedoch nicht auf die in der Symbolwort-Umsetzung gebildeten Symbole direkt, sondern teilweise auf ihre Differenzen angewendet (*Differenzielle Encodierung* der beiden höchstwertigen Bits des Symbols[59]). Abb. 6.5-9 zeigt das Blockschaltbild des DVB-C Encoders, wie er erstmals Ende 1994 als ETSI-Norm *EN 300 429*[6.13] festgelegt wurde.

Der DVB-C Decoder ist korrespondierend aufgebaut. Die folgende Tabelle 6.5-1 zeigt die Datenraten, die mit den unterschiedlichen QAM-Verfahren erreicht werden können, wenn eine komplette Kanalbandbreite von 8 MHz genutzt werden kann[60].

Parameter DVB-C

Tab. 6.5-1: *DVB-C Parameter für eine 8 MHz-Kanalbandbreite*

QAM-Verfahren	Bits pro Symbol	Bruttodatenrate [Mbit/s]	Nutzdatenrate [Mbit/s]
16-QAM	4	27.83	25.64
32-QAM	5	34.78	32.05
64-QAM	6	41.74	38.47
256-QAM	8	55.65	51.29

6.5.6 Digitale Modulation für die terrestrische TV-Übertragung

Problematik des Mehrwegeempfangs

Mehrwegeempfang
Geisterbilder

Ähnlich wie beim Breitbandkabel existieren auch für die terrestrische TV-Übertragung Kanäle mit 7 MHz und 8 MHz Bandbreite. Von diesen gibt es jedoch nur insgesamt etwa 60 Kanäle; beim Breitbandkabel sind es bis zu 97 Kanäle. Die relativ große Reichweite der terrestrisch ausgestrahlten analogen Signale kann zum so genannten *Mehrwegeempfang* führen. Ein Empfänger wertet sowohl das Signal eines nahen, starken Senders aus, als auch das schwache Signal eines weiter entfernt liegenden Senders, der auf der gleichen Frequenz sendet. Die unterschiedlichen Entfernungen vom Sender zum Empfänger führen zu einem geringen Zeitversatz der empfangenden Signale. Beim analogen Fernsehen äußert sich dies durch so genannte *Geisterbilder*, die versetzt mit geringerer Amplitude auf dem Wiedergabebildschirm erkennbar sind.

Damit diese Störungen möglichst nicht auftreten, geschieht die analoge Versorgung eines einzelnen Programms in benachbarten Regionen über verschiedene TV-Kanäle. Damit reduziert sich jedoch auch die Anzahl der verschiedenen Programme, die terrestrisch flächendeckend möglich sind auf etwa 7 bis 8.

Trotz der Verteilung der Programme auf unterschiedliche Sendefrequenzen kann bei der analogen TV-Übertragung der Mehrwegeempfang prinzipiell nicht ausgeschlossen werden, da insbesondere Reflexionen an Gebäuden oder Gebirgszügen sich nicht überall verhindern lassen und ebenfalls sichtbare Geisterbilder (*Echos*) verursachen.

---

59 Die Differenzbildung ist notwendig, um eine korrekte Phaseninformation am Empfänger generieren zu können.

60 In [6.13] wird innerhalb eines 8 MHz Kabelkanals eine nutzbare Bandbreite von etwa 7.9 MHz bis maximal 7.96 MHz angenommen.

Bei der Einführung der digitalen terrestrischen TV-Übertragung wurde die Problematik der knappen Kanalressourcen und des Mehrwegeempfangs besonders berücksichtigt. Erwartungsgemäß verursacht nämlich der Mehrwegeempfang bei der digitalen Übertragung ein *Symbolübersprechen*, was einen erheblichen Anstieg der *Bitfehlerrate* nach sich zieht[61].

Gleichwellennetze – OFDM

Beherrschen lässt sich der Mehrwegeempfang in so genannten *Gleichwellennetzen* (*Single Frequency Networks – SFN*). Dazu wird zunächst ein *Multiträgersystem* mit einer großen Anzahl von orthogonalen Trägern (*Orthogonal Frequency Division Multiplex – OFDM*) aufgebaut. Die Daten werden zeitlich sequenziell in Symbolen[62] übertragen. Eine Auswertung der empfangenden Signale gelingt auch beim Mehrwegeempfang, wenn zwischen der Übertragung der Symbole Schutzintervalle (*Guard-Intervall*) eingefügt werden, die in der Praxis etwa ein viertel so lang sind, wie die Symboldauer selbst. Die Kombination aus *OFDM* und Fehlerschutz wird *Coded OFDM – COFDM* genannt.

Guard-Intervall

Aufgrund der Komplexität dieses Themas können nachfolgend nur die Grundzüge der *COFDM* skizziert werden. Weitere Details lassen sich beispielsweise in [6.6] oder [6.7] nachlesen.

Orthogonaler Frequenzmultiplex – *OFDM*

Aufbau OFDM

In den bisher beschriebenen digitalen Modulationsverfahren wurden jeweils zwei orthogonale Träger verwendet. Die Zahl der Symbolträger muss aber nicht zwangsläufig auf zwei beschränkt sein. Es lassen sich vielmehr ganze Systeme von zueinander orthogonalen Trägern für eine Modulationsart definieren.

Das Besondere der OFDM liegt darin, dass der Gesamtdatenstrom auf eine Vielzahl von *n* einzelnen Trägern aufgeteilt wird, die ihrerseits jeweils für die Dauer eines Symbols *T* in kurzen Impulsen übertragen werden. Dieses Verfahren hat den Vorteil, dass selektiv auftretende Einbrüche in der Übertragungsfunktion des Kanals nur einzelne Unterträger betreffen, was durch geeignete Kanalcodierung ausgeglichen werden kann. Das Pulsen der einzelnen Träger kann *signaltheoretisch* als *Faltung* mit dem *Rechteckimpuls* verstanden werden. Dieser korrespondiert im Frequenzspektrum mit der so genannten *si-Funktion*:

si-Funktion

$$\sqcap_r(t) \quad = \begin{cases} \frac{1}{T} & \text{für} \quad |t| < \frac{T}{2} \\ \frac{1}{2T} & \text{für} \quad |t| = \frac{T}{2} \\ 0 & \text{für} \quad |t| > \frac{T}{2} \end{cases} \qquad si(f) := \frac{\sin(\pi f T)}{\pi f T} \quad . \tag{6.5-12}$$

61 Eine Richtantenne am Empfänger kann diese Problematik etwas entschärfen. Doch verbietet sich diese Lösung in der Regel für mobile Anwendungen. Hier wird eine ungerichtete Stabantenne verwendet.

62 Hier bezieht sich der Begriff des Symbols auf die Übertragung selbst und muss nicht zwangsläufig identisch mit dem Symbol sein, der für den Datenstrom definiert wurde.

Damit die korrespondierenden Spektren im Decoder wieder ohne Übersprechen von einander getrennt werden können (*Orthogonalitätsbeziehung*) muss der *Frequenzabstand* Δf zwischen den Einzelträgern dem reziproken Wert der Symboldauer T des Teildatenstromes entsprechen:

Frequenzabstand

$$\Delta f = \frac{1}{T}. \qquad 6.5\text{-}13$$

Die Symboldauer hängt auch von der Modulation der Einzelträger selbst ab. Beispiele für mögliche Modulationen sind die schon bekannte *4-PSK* ($T = 2T_{Bit}$) oder die *16-QAM* ($T = 4T_{Bit}$). Mit der Anzahl der Träger lässt sich die erzielbare Bitrate einstellen. Die Anzahl der in jedem Symbol übertragbaren Bits Q ergibt sich bei Verwendung der *Q-PSK* für eine gegebene Kanalbandbreite $B_{\text{Ü}}$ bzw. für eine gegebene Anzahl von Träger *n* zu:

$$Q = 2 \cdot n = 2 \cdot \frac{B_{\text{Ü}}}{\Delta f} = 2 \cdot B_{\text{Ü}} \cdot T. \qquad 6.5\text{-}14$$

In Abb. 6.5-10 ist der Sachverhalt der OFDM im Frequenzspektrum skizziert.

OFDM Frequenzspektrum

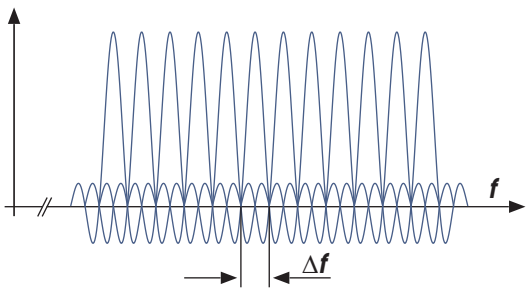


Abb. 6.5-10: OFDM im Frequenzbereich

Für die Realisierung der OFDM sind zwei Grundschritte notwendig. Zunächst muss der Datenstrom zyklisch in *n* verschiedene Einzeldatenströme zerlegt werden, die später auf die *n* verschiedenen Träger aufmoduliert werden sollen. Für die eigentliche Generierung des OFDM-Signals wird die *Discrete Inverse Fourier Transformation – DIFFT* genommen.

Realisierung OFDM mit DIFFT

Einführung des Schutzintervalls

Guard Intervall

Die Problematik des Mehrwegeempfangs bei der terrestrischen Funkübertragung wurde bereits beschrieben. Wird vor jedem zu übertragendem Symbol ein so genanntes *Schutzintervall – Guard Intervall* der Dauer T_G eingefügt, können die Interferenzen in diesem Intervall ausklingen, bis das Symbol sicher detektiert werden kann. Das Bemerkenswerte an dieser Technik ist die Tatsache, dass Echos und Nachbarsender sogar konstruktiv zur Signalstärke am Empfänger beitragen können. Es muss allerdings beachtet werden, dass das Guard Intervall die nutzbare Zeitspanne für das Symbol selbst verkürzt, weshalb ein sinnvoller Kompromiss zwischen der Dauer des Symbols und der Dauer des Guard Intervalls gefunden werden muss. Abb. 6.5-11 zeigt ein Beispiel für die konstruktive Ausnutzung von Signalen aus dem Mehrwegeempfang und die Funktion des Guard Intervalls.

Mehrwegeempfang und Guard Intervall

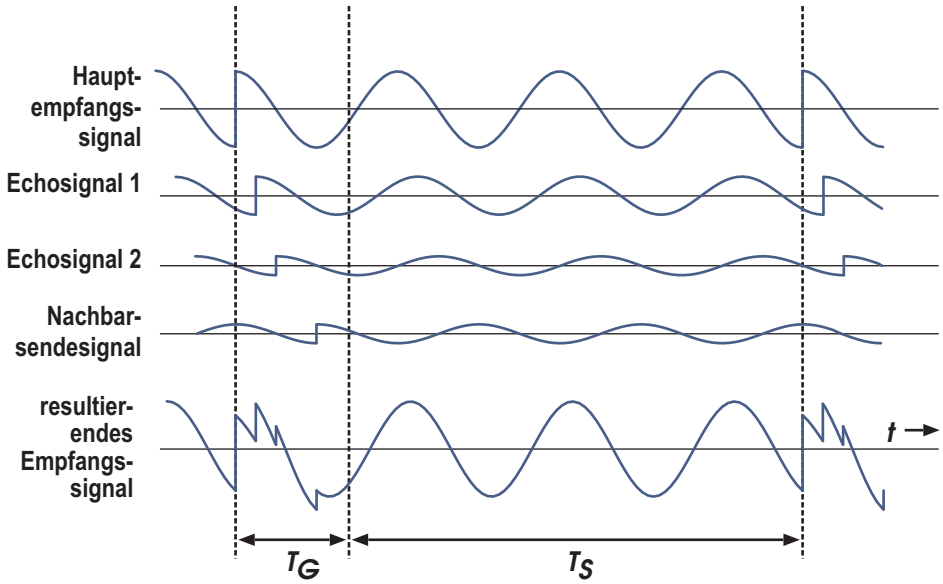


Abb. 6.5-11: Mehrwegeempfang und Einfügung eines Guard Intervalls für die Auswertung (T_G/T_S = 1/4)

Schutzintervalle bei DVB-T

Für die digitale terrestrische TV-Übertragung in Europa wurden im so genannten *DVB-T* Standard (ETSI-Norm *EN 300 744*[6.14]) Schutzintervalle mit den Verhältnissen T_G/T_S von 1/4, 1/8, 1/16 und 1/32 eingeführt.

Für die Berechnung des Guard Intervalls kann in DVB-T Netzen mit Senderabständen *d* von etwa 60 km ausgegangen werden. Soll das Schutzintervall weiterhin nicht kürzer sein als die Signallaufzeit zwischen den benachbarten Sendern, dann lässt sich TGwie folgt errechnen:

$$T_{G,60km} = \frac{d}{c_0} = \frac{60\,km}{3 \cdot 10^5\ \frac{km}{s}} = 200\ \mu s. \qquad 6.5\text{-}15$$

Bei einer Symbolnutzdauer von 896 µs[63] und einem viertel so langen Guard Intervall kann diese Bedingung erfüllt werden. Die zeitliche Verlängerung der Symboldauer durch das Guard Intervall führt im Frequenzbereich dazu, dass die OFDM-Träger entsprechend näher zusammenrücken.

2k-Modus, 8k-Modus

Aus technischen Gründen (OFDM-Bildung mit DIFFT) wird die Anzahl der Träger bei DVB-T als eine Potenz von 2 angesetzt. Es ist die OFDM mit 2048 (*2k-Modus*) und 8192 (*8k-Modus*) üblich. In der Praxis werden jedoch nur 1705 bzw. 6817 Einzelträger verwendet, was eine bessere Anpassung des Systems an die tatsächlichen Kanaleigenschaften (Bandgrenze, Störeinflüsse etc.) ermöglicht.

6.5.7 DVB-T Übertragungskonzept

Das digitale terrestrische Fernsehen DVB-T besitzt im Vergleich zu DVB-S und DVB-C die größte Komplexität bei der technischen Realisierung. In Deutschland wird DVB-T mit einer OFDM im 8k-Modus mit 16-QAM betrieben. Die Signalverarbeitung des DVB-T Senders ist in Abb. 6.5-12 dargestellt.

DVB-T Encoder

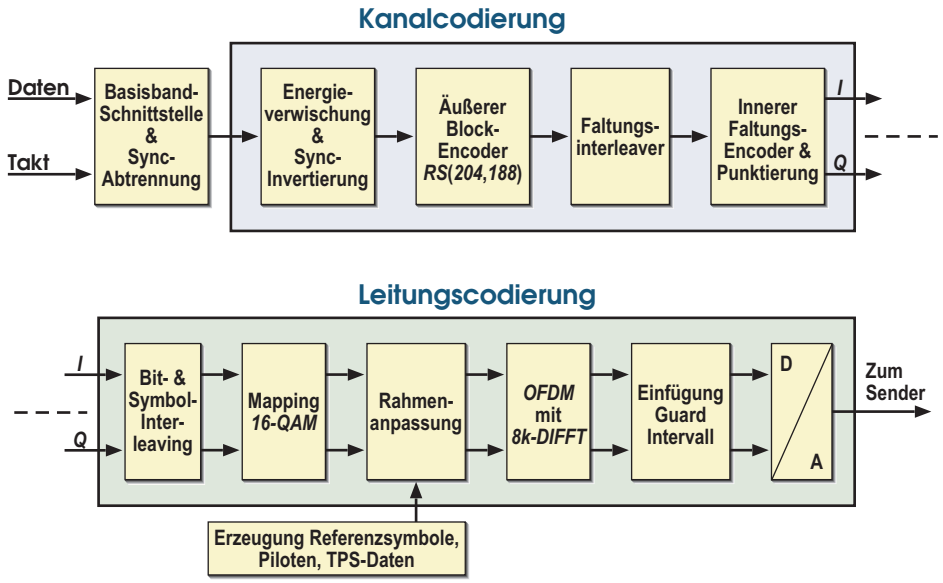


Abb. 6.5-12: Blockschaltbild des DVB-T Encoders

Die Kanalcodierung (*äußerer Fehlerschutz*, *Interleaver*, *innerer Fehlerschutz*) ist mit der des DVB-S Systems identisch. Man beachte, dass in einigen Systemkonzepten von DVB-T kein innerer Fehlerschutz vorgesehen ist. Für die COFDM wird der Datenstrom in mehreren Schritten (*Bit- undSymbol-Interleaving*, *16-QAM Mapping*) vorbereitet. Das OFDM-Signal wird in so genannten Rahmen (*Frames*) organisiert. Die Rahmen enthalten jeweils 68 OFDM-Symbole, von denen jedes sich auf 6817 Träger (8k-Modus) bzw. 1705 Träger (2k-Modus) verteilt.

Pilotsignale, TPS

Um eine sichere Übertragung zu gewährleisten werden zusätzliche Pilotträger mit einer Phasenlage von 0° oder 180° integriert. Diese lassen sich ebenfalls aus dem 16-QAM Signal wegen der besonderen Phasenlage ableiten. Man unterscheidet zwischen *kontinuierlichen Pilotsignalen* (*Continual Pilots*), *verstreuten Pilotsignalen* (*Scattered Pilots*) und dem *Transmission Parameter Signaling – TPS*.

Die *Continual Pilots* dienen zur *Frequenzsynchronisation* beim Empfänger. Die *Scattered Pilots* wechseln ständig ihre zeitliche und spektrale Position innerhalb des übertragenden Signals und werden zur *Kanalschätzung* verwendet. Mit dem *TPS* werden dem Empfänger die *Parameter des Übertragungsverfahrens* übermittelt (Modulationsverfahren, Coderate des Faltungscoders, Länge des Schutzintervalls, 2k- oder 8k-Modus). In der nachfolgenden Tabelle 6.5-2 sind wichtige Parameter des DVB-T Verfahrens zusammengefasst.

Parameter DVB-T

Tab. 6.5-2: *Parameter des DVB-T Übertragungskonzepts*

Parameter	2k-Modus				8k-Modus			
Anzahl nominelle Träger	2048				8192			
Anzahl tatsächliche Träger	1705				6817			
Anzahl Träger mit Nutzinformation	1512				6048			
Genutzte Symboldauer T_S	224 µs				896 µs			
Trägerabstand Δf	4464 Hz				1116 Hz			
Belegte Bandbreite	7.609 MHz				7.612 MHz			
Gesamtsymboldauer [µs]	280	262	238	231	1120	1008	952	924
Guard Intervall T_G [µs]	56	28	14	7	224	112	56	28
T_G / T_S	1/4	1/8	1/16	1/32	1/4	1/8	1/16	1/32
Symbolrate je Träger [kSym./s]	3.57	3.96	4.20	4.32	0.89	0.99	1.05	1.08
übertragbare Symbolrate mit Nutzinfo [MSym./s]	5.39	5.99	6.35	6.64	5.39	5.99	6.35	6.54
Brutto-Bitrate 16-QAM 64-QAM	21.56 - 26.16 Mbit/s 32.34 - 39.24 Mbit/s				***21.56*** - 26.16 Mbit/s 32.34 - 39.24 Mbit/s			
Netto-Bitrate[64] 16-QAM 64-QAM	13.24 - 16.07 Mbit/s 19.86 - 24.10 Mbit/s				***13.24*** - 16.07 Mbit/s 19.86 - 24.10 Mbit/s			

63 Wert für den 8k-Modus von DVB-T

64 Berücksichtigt sind bei der Kanalcodierung der *RS*(188,204) Blockcode (R = 0.9216) und der Faltungscode mit R = 2/3

Selbsttestaufgabe 6.5-1:

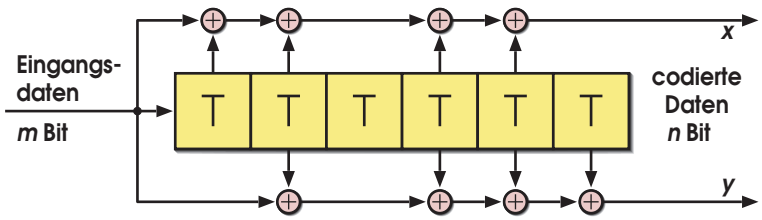
Welches logische Signal muss man am Enable-Eingang anlegen, um die Energieverwischung einzuschalten, welches, um sie wieder auszuschalten?

Selbsttestaufgabe 6.5-2:

Ein digitaler Datenstrom soll mit einem Read-Solomon-Code in Informationsblöcken zu je 100 Symbolen fehlergeschützt werden. Wie viele Korrektursymbole müssen ergänzt werden, wenn bis zu 10% der Daten korrigiert werden sollen? Geben Sie auch die Schreibweise für diesen RS-Code an!

Selbsttestaufgabe 6.5-3:

Gegeben sei der folgende Faltungscode.



Geben Sie die Generatorpolynome G_x und $G_y$als Oktalzahlen an!

Selbsttestaufgabe 6.5-4:

Für eine DVB-S Übertragungsstrecke stehe ein 46 MHz breiter Kanal zur Verfügung. Errechnen Sie die maximale Brutto- und maximale Nettodatenrate in [Mbit/s], welche sich bei den Coderaten 1/2, 2/3, 3/4, 5/6 und 7/8 erreichen lassen! Berücksichtigen Sie die Kanalcodierung (Read-Solomon-Code und Faltungscode) und die Leitungscodierung (Roll-Off-Filter und Modulation), die bei DVB-S eingesetzt wird!

Selbsttestaufgabe 6.5-5:

Geben Sie für den 2k- und 8k-Modus den maximalen Senderabstand für T_G/T_S = 1/4, 1/8, 1/16 und 1/32 an!

7 Verbesserte Codiersysteme für die Bildkommunikation

Übersicht

Im Kapitel 5 wurden verschiedene Codiersysteme für Bewegtbilder vorgestellt, die in Standardanwendungen wie TV-Broadcast und Speicherung auf DVD geeignet sind. Die Standards MPEG-1 und MPEG-2 verlangen für eine akzeptable Bildqualität Datenraten oberhalb von 1 bis 2 Mbit/s. Viele Kommunikationsnetze sind jedoch nicht für so hohe Datenraten ausgelegt (z. B. ISDN). Daher sind für diese Netzstrukturen weiter optimierte Bildcodiersysteme notwendig.

In Abschnitt 7.1 bis Abschnitt 7.3

werden drei verbesserte Codierkonzepte für die Bildkommunikation vorgestellt, die Standards *ITU-T H.263*, *ISO MPEG-4 Video* und der gemeinsame Standard der ITU und der ISO, *ITU-T H.264/AVC* bzw. *ISO/IEC MPEG-4 Part 10*. Abschnitt 7.4 schließt mit einem kurzen zusammenfassenden Vergleich der Bildcodiersysteme und einen Ausblick auf weitere Codierverfahren dieser Kurs ab.

7.1 ITU-T H.263

Einfache Videotelefonie ist mit dem ITU-T Standard H.261 erstmals möglich geworden (vgl. Abschnitt 5.1.2). Zwei wesentliche Mängel führten aber dazu, dass sich der Standard nicht auf breiter Front durchsetzen konnte:

Mängel von H.261

- Die erzielbare Bildqualität ist für den vorgesehenen Anwendungsfall, der Videotelefonie über *ISDN*-Netze mit 64 kbit/s, nicht hinreichend gut genug.
- Der Standard ist für leitungsvermittelnde Netze ausgelegt. Er eignet sich nicht so sehr für paketvermittelnde Netze (z. B. Internet), da die Datenstruktur hierfür nicht ausgelegt ist.

Mit der Verabschiedung der ersten Version des Standards *ITU-T H.263* Ende 1995 gelang ein vielversprechender Durchbruch für die Bildkommunikation bei niedrigen Datenraten. Etwa parallel dazu begann auch die Entwicklung der Videocodierung für den *MPEG-4* Standard (vgl. Abschnitt 7.2), der zwar ebenfalls für niedrige Datenraten ausgelegt ist, aber aufgrund seiner komplexen und interaktiven Multimediaarchitektur für die klassische Bildkommunikation oft überdimensioniert ist.

7.1.1 Übersicht

Der Standard *ITU-T H.263*[7.2] ist ursprünglich als direkter Nachfolger zum Standard *ITU-T H.261* für niedrigen Datenraten in die Bildkommunikation konzipiert worden. Das verbesserte Codierkonzept liefert für H.263 jedoch auch bei höheren Datenraten eine deutlich bessere Bildqualität als H.261 bei vergleichbaren Datenraten. Um quantitativ mit beiden Codiersystemen vergleichbare Bildqualitäten zu erzielen, werden für H.261 in etwa doppelt soviel Daten benötigt wie für H.263.

Allerdings ist dann auch der Einsatz von so genannten *Codieroptionen* von H.263 notwendig.

Doch auch schon der *Baseline-Algorithmus* beinhaltet einige wesentliche Neuerungen:

Eigenschaften H.263 Baseline

- Die *Genauigkeit der Bewegungskompensation* wurde von einem auf einen halben Bildpunkt angehoben. Wie bei H.261, MPEG-1 und MPEG-2 ist weiterhin nur ein Bewegungsvektor pro Makroblock zulässig. Mit der erhöhten Genauigkeit kann aber auf das Schleifenfilter (*Filter*) in der Rückkopplung verzichtet werden (Abb. 5.1-1).
- Neben den schon bei H.261 *gebräuchlichen AuflösungenCIF* (352 x 288) und *QCIF* (176 x 144) werden bei H.263 die drei weitere Auflösungen *SQCIF* (128 x 96), *4CIF* (704 x 576)[65] und *16CIF* (1408 x 1152)[66] unterstützt.
- Die *Effizienz der variablen Lauflängencodierung* wurde durch eine Verbesserung beim Zusammenfassen von verschiedenen Codierfällen gesteigert. So nutzt beispielsweise eine 3D-Entropiecodierung gemeinsam die drei Ereignisse „*End-Of-Block*“, Anzahl der im „Zig-Zag-Scanning“ zuvor zu Null quantisierten DCT-Koeffizienten und der Wert des „ungleich Null“ quantisierten DCT-Koeffizienten.

7.1.2 Struktur des ITU-T H.263 Standards in der Grundversion

In der Grundversion des Standards ITU-T H.263 haben sich die Blöcke des Funktionsschaltbildes nicht wesentlich zum Standard ITU-T H.261 (Abb. 5.1-1) geändert. Auch die definierten Hierarchieebenen (*Layer*) sind weitgehend übernommen worden:

- *Sequence Layer*,
- *Picture Layer*,
- *Group of Blocks Layer*,
- *Macroblock Layer* und *Block Layer*.

Der *Group of Blocks Layer* wird bei H.263 oft auch *Slice Layer* genannt.

Hierarchieebenen H.263

Gegenüber ITU-T H.261 hat sich die Aufteilung des Bildes in *Group of Blocks – GOB* (Abb. 5.1-3) durchgesetzt. In Abb. 7.1-1 ist die veränderte hierarchische Aufteilung des Bildes dargestellt. Die GOB ist nun zeilenweise organisiert, d. h. eine GOB geht über die gesamte Bildbreite. Beim CIF-Format existieren so 18 GOBs pro Bild. Jede GOB enthält 22 Makroblöcke, die selbst wie in ITU-T H.261 mit einer Farbunterabtastung von 4:2:0 organisiert sind.

---

65 Das entspricht *Cropped ITU-R BT.601*.

66 Wegen der 16:9 Darstellung bei HD-Formaten reicht die Auflösung nicht z. B. für die SMPTE 296M Norm mit 1920 x 1080 Bildelementen aus.

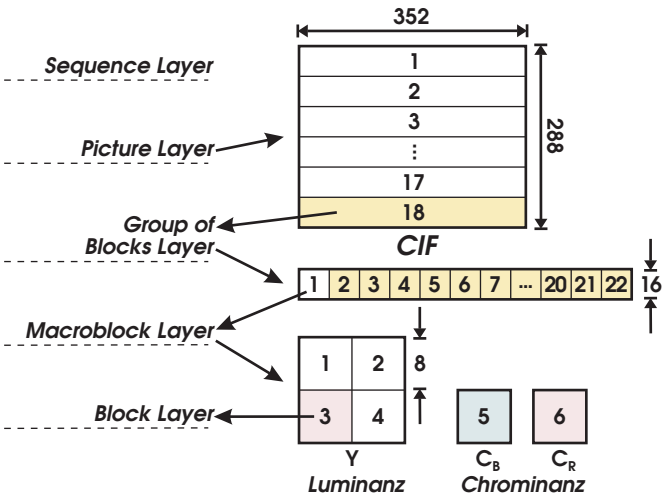


Abb. 7.1-1: Hierarchische Aufteilung des Bildes in GOBs, Makroblöcke und 4:2:0-Farbunterabtastung nach *ITU-T H.263*

PLUSTYPE

Die hierarchische Struktur setzt sich in der Syntax des Videodatenstroms fort. Abb. 7.1-2 zeigt die Datenstruktur des H.263 Baseline-Algorithmus. Die entsprechenden Felder im *Picture Layer* (*PLUSTYPE*), die für die Signalisierung von zusätzlichen *Codieroptionen* (s. u.) zuständig sind, wurden der Übersichtlichkeit wegen weggelassen.

Im Datenstrom des *Picture Layers* sind folgende Felder definiert:

Picture Layer

- *Picture Start Code – PSC* (22 Bit),
- *Temporal Reference – TR* (8 Bit),
- *Type Information – PTYPE* (8 bzw. 13 Bit),
- *Quantizer Information – PQUANT* (5 Bit),
- *Continuous Presence Multipoint – CPM* (1 Bit),
- *Picture Sub Bitstream Indicator – PSBI* (2 Bit) – optional,
- *Temporal Reference for B-Pictures – TRB* (3 Bit) – optional,
- *Quantization Information for B-Pictures – DBQUANT* (2 Bit) – optional,
- *Extra Insertion Information – PEI* (1 Bit),
- *Spare Information – PSPARE* (0, 8, 16... Bit) – optional,
- *End of Sequence – EOS* (22 Bit) und
- *Stuffing – ESTUF* und *PSTUF* (variable).

Der Datenstrom des *Group of Blocks Layers* setzt sich aus folgenden Komponenten zusammen:

Group of Blocks Layer

- *Group of Block Start Code – GBSC* (17 Bit),
- *Group Number – GN* (5 Bit),
- *GOB Sub Bitstream Indicator – GSBI* (2 Bit) – optional,
- *GOB Frame ID – GFID* (2 Bit),

- *Quantization Information – GQUANT* (5 Bit) und
- *Stuffing – GSTUF* (variable).

Der *Makroblock Layer* enthält die Felder

Makroblock Layer

- *Coded Macroblock Indication – COD* (1 Bit) – nur bei Inter codierten Makroblöcken vorhanden,
- *Macroblock Type and Coded Block Pattern for Chrominance – MCBPC* (VLC, d .h. mit *variabler Lauflängencodierung* codiert),
- *Macroblock Mode for B-Blocks – MODB* (VLC) – optional,
- *Coded Block Pattern for B-Blocks – CBPB* (6 Bit) – optional,
- *Coded Block Pattern for Luminance – CBPY* (variable),
- *Qunatization Information – DQUANT* (2 Bit),
- *Motion Vector Data – MVD* (VLC) und *MVD2...4* (VLC) – optional und
- *Motion Vector Data for B-Macroblock – MVDB* (VLC) – optional.

Block Layer

Ist der *Block Layer* vorhanden, so wird dort zunächst mit 8 Bit der DC Koeffizient definiert (*DC Coefficient for INTRA Blocks – INTRADC*). Anschließend folgen die übrigen Transformationskoeffizienten, die mit variabler Lauflänge codiert sind (*Transform Coefficient – TCOEF*).

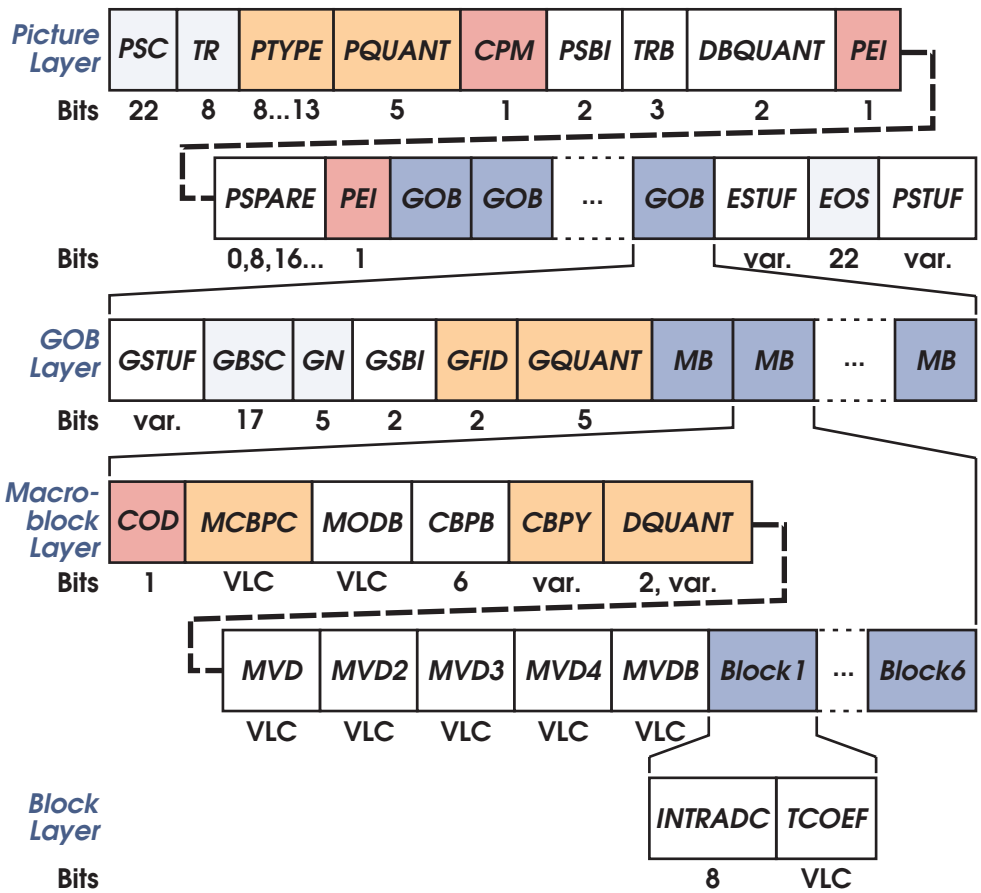


Abb. 7.1-2: Aufbau des Videodatenstroms beim ITU-T H.263 Baseline-Standards

Datenstruktur des Videodatenstrom H.263

Die Datenstruktur ist schon in der Grundversion (*ITU-T H.263 Baseline*) recht komplex und soll hier nicht weiter vertieft werden. Weiterführende Einzelheiten zu den Feldern lassen sich beispielsweise in [7.1] oder [7.2] nachlesen.

Erweiterungen für Codieroptionen

Umfangsreiche Erweiterungen erfährt die Datenstruktur mit der Einführung von *Codieroptionen* (s. u.). Diese werden durch ein neues Feld *PLUSTYPE* signalisiert, das im *Picture Layer* zwischen die Feldern *PTYPE* und *PQUANT* eingefügt wird. Einzelheiten hierzu sind in [7.4] aufgeführt.

7.1.3 ITU-T H.263, Version 1 – Codieroptionen

In Ergänzungen zum Standard ITU-T H.263 sind weitere optional unterstützte Funktionalitäten, so genannte *Codieroptionen* aufgeführt. Mit diesen lassen sich die eigentlichen Verbesserungen des Standards ITU-T H.263 gegenüber H.261 erzielen. Von diesen auch als *Optional Enhanced Modes* bekannten Codieroptionen gibt es in der ersten Version des Standards von 1995 insgesamt vier:

Codieroptionen H.263 V1

- *Unrestricted Motion Vector Mode – UMV* (Annex D),
- *Syntax-Based Arithmetic Coding Mode – SAC* (Annex E),
- *Advanced Prediction Mode – AP* (Annex F) und
- *PB-Frames Mode* (Annex G).

Im Feld *PTYPE* wird in den Bits 10 bis 13 angegeben, ob und welche Codieroptionen der Datenstrom beinhaltet.

Annex D Unrestricted Motion Vector Mode

Standardmäßig sind die Bewegungsvektoren beim Standard H.263 auf die Bildpunktverschiebung innerhalb des Intervalls [–16; 15.5] beschränkt. Im *Unrestricted Motion Vector Mode* (Annex D) ist der Bereich auf [–31.5; 31.5] erweitert worden. Zudem können Verschiebungsvektoren in diesem Mode auch auf virtuelle Bildbereiche außerhalb des aktiven Bildbereiches zeigen. Damit lassen sich Auf- und Verdeckungsfälle in Bildrandbereichen besser beschreiben.

Annex E Syntax-Based Arithmetic Coding Mode

Statt einer variablen Lauflängencodierung wird beim *Syntax-Based Arithmetic Coding Mode* (Annex E) eine so genannte arithmetische Codierung verwendet, die bei gleicher Rekonstruktionsqualität etwa 5% an Datenrate einspart.

Annex F Advanced Prediction Mode

Beim *Advanced Prediction Mode* (Annex F) darf die Anzahl der Bewegungsvektoren pro Makroblock eins oder vier betragen. So kann für jeden 8 x 8 Luminanzblock eines Makroblocks je ein eigener Verschiebungsvektor zur Verfügung stehen. Weiterhin findet eine *Block überlappende Bewegungskompensation* statt. Zu diesem Zweck können auch die Verschiebungsvektoren der örtlichen Nachbarblöcke mögliche Kandidaten für eine Bewegungskompensation im aktuellen Bildblock herangezogen werden (*Overlapped Block Motion Compensation – OBMC*). Mehrere Kandidatenvektoren werden gewichtet miteinander verrechnet und auf die aktuellen 8 x 8 Bildblöcke angewendet. In der Praxis lässt sich mit diesem Mode bei gleicher Datenrate ein etwa 1 bis 2 dB besserer PSNR erreichen.

Annex G PB-Frame Mode

Beim *PB-Frame Mode* (Annex G) wird eine *gemeinsame Codierung* von zwei Ein-

zelbildern vorgenommen. Die Bezeichnung „*PB*“ gibt an, dass es sich hierbei um P- bzw. B-Bilder handelt, wie sie schon bei MPEG definiert wurden. Abb. 7.1-3 zeigt die Prädiktionsbeziehungen eines PB-Frames.

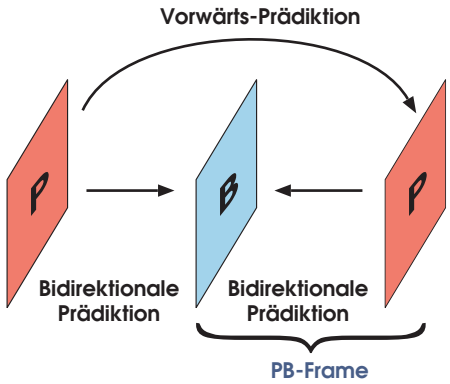


Abb. 7.1-3: Prädiktionsbeziehungen beim PB-Frame

7.1.4 ITU-T H.263, Version 2 (H.263+) – Codieroptionen

Die zweite Version von H.263 (H.263+) ist von der ITU-T Anfang 1998 verabschiedet worden [7.2]. In H.263+ sind 13 weitere Anhänge[67] dem H.263-Standard hinzugefügt worden, die oft die Codiereffizienz verbessern und die Benutzung weiterer Funktionalitäten ermöglichen:

Codieroptionen H.263 V2

- *Forward Error Correction for Coded Video Signal* (Annex H),
- *Advanced INTRA Coding Mode* (Annex I),
- *Deblocking Filter Mode* (Annex J),
- *Slice Structured Mode* (Annex K),
- *Supplemental Enhancement Information Specification* (Annex L),
- *Improved PB-Frames Mode* (Annex M),
- *Reference Picture Selection Mode* (Annex N),
- *Temporal, SNR, and Spatial Scalability Mode* (Annex O),
- *Reference Picture Resampling* (Annex P),
- *Reduced-Resolution Update Mode* (Annex Q),
- *Independent Segment Decoding Mode* (Annex R),
- *Alternative INTER VLC Mode* (Annex S) und
- *Modified Quantization Mode* (Annex T).

Annex H Forward Error Correction

Eine Besonderheit von H.263+ ist die verbesserte Unterstützung der Übertragung von komprimierten Videosignalen über verlustbehaftete, paketvermittelte Netze. Für eine *Forward Error Correction - FEC* kann ein BCH-Code[68] gemäß Annex H verwendet werden.

Annex I Advanced Intra Coding

Bei dem *Advanced Intra Coding Mode* (Annex I) wird eine Prädiktion von Intra-codierten Blöcken aus benachbarten Blöcken angewendet. Weiterhin wird eine veränderte inverse Quantisierung der Intra-codierten DCT Koeffizienten und eine eigene VLC-Tabelle für diese Koeffizienten benutzt. Ein zusätzliches, 1 oder 2 Bit langes Codewort zeigt auf der Makroblock-Ebene den entsprechenden Prädiktionsmode an.

Annex J Deblocking Filter

Das wieder optional mögliche Schleifenfilter (*Deblocking Filter Mode* - Annex J) dient dazu, gezielt an den Blockgrenzen Artefakte zu unterdrücken. Die Filterkoeffizienten werden in Abhängigkeit von den gewählten Quantisierungsstufen der angrenzenden Blöcke ausgewählt. Eine grobe Quantisierung führt zu einer stärkeren Verschleifung der Bildinhalte im Bereich der Blockgrenzen und verhindert so störende Blockartefakte.

Annex K Slice Structured Mode

Der *Slice Structured Mode* (Annex K) dient nicht zur Erhöhung der Codiereffizienz, sondern ermöglicht eine einfache Nutzung des Datenstroms im Zusammenspiel mit paketvermittelnden Netzen. Dazu wird der GOB-Layer durch einen Slice Layer ersetzt. Die Anordnung der Daten in der Slice Struktur können an die Übertragung angepasst werden (z. B. Wahl der Paketgröße).

Annex L Supplemental Enhancement Information

Mit der *Supplemental Enhancement Information* (Annex L) können wiedergabespezifische Informationen in den Datenstrom integriert werden. Es können zum Beispiel Teile des Bildes eingefroren oder ein Chroma-Key-Signal hinzugefügt werden.

Annex M Improved PB Frames Mode

Der *Improved PB Frames Mode* (Annex M) stellt eine Erweiterung des *PB Frames Modes* Option dar. Mit diesem Mode ist der Videodatenstrom weniger empfindlich gegenüber sprunghaften Szenenänderungen (z. B. Schnitt), da ein zusätzlicher vorwärts prädizierender Bewegungsvektor pro Makroblock zugelassen ist.

Annex N Reference Picture Selection Mode

Der *Reference Picture Selection Mode* (Annex N) dient dazu, die Fehlerempfindlichkeit des Datenstroms z. B. aufgrund von Paketverlusten zu minimieren. Dazu werden zusätzliche Referenzverweise eingefügt, die alternativ als Prädiktionsbilder herangezogen werden können.

Annex O Spatial Scalability Mode

Ähnlich wie bei entsprechenden MPEG-2 Profiles wird mit dem *Temporal, SNR, and Spatial Scalability Mode* (Annex O) die zeitliche, örtliche oder Signalpegel abhängige Skalierung der Codierqualität geregelt.

---

67 Annex H *"Forward Error Correction for Coded Video Signal"* gehört eigentlich schon zur Basisversion von H.263.

68 *FEC*– vgl. Abschnitt 6.4. BCH-Codes ist eine Gruppe von zyklischen Blockcodes. Der schon bekannte Reed-Solomon-Code gehört ebenfalls zu der Gruppe der BCH-Codes. Bei H.263 wird ein BCH(511,493) verwendet.

Annex P Reference Picture Mode

Mit dem *Reference Picture Resampling Mode* (Annex P) ist an Referenzbildern (I- oder P-Bilder) vor der Prädiktion eine örtliche Transformation (z. B. Änderung der Bildgröße, Rotation o. ä.) möglich. Damit lassen sich globale Bewegungen, wie Zoom oder Rotation mit ganz wenigen Transformationsparametern beschreiben, was für diese Fälle zu einer deutlichen Reduktion der erforderliche Datenrate führen kann.

Annex Q Reduced Resolution Update Mode

Beim *Reduced Resolution Update Mode* (Annex Q) werden die Makroblöcke in x- und y-Richtung jeweils doppelt so groß gewählt. Damit reduziert sich die Anzahl der zu übertragenden Makroblöcke auf ein viertel. Mit diesem Mode ist es möglich, bei Bedarf die zeitliche Auflösung für Szenen mit schnellen Bewegungen zu erhöhen. Im Gegenzug wird die örtliche Auflösung reduziert.

Annex R Independent Segment Decoding Mode

Im *Independent Segment Decoding Mode* (Annex R) können Bildsegmentgrenzen definiert werden, die wie Bildgrenzen behandelt werden: Keine Datenabhängigkeiten über die Bildsegmentgrenzen hinweg sind zugelassen. Dieser Mode macht den Datenstrom robuster gegenüber Übertragungsstörungen. Die Bildsegmentgrenzen werden dazu an die Paketgrößen für die Übertragung angepasst.

Annex S Alternate Inter VLC Mode

Der aktivierte *Alternate INTER VLC Mode* (Annex S) ermöglicht es, die VLC-Tabelle für Intraframe-codierte Blöcke auch für Interframe-codierte Blöcke verwenden zu können. In einigen Fällen führt diese Tabelle zu einem geringeren Datenaufkommen.

Annex T Modified Quantization Mode

Eine feinere Abstufung bei der Quantisierung geschieht beim *Modified Quantization Mode* (Annex T). Damit kann eine bessere Regelung der Datenrate erreicht werden. Zudem wird der Wertebereich für die DCT-Koeffizienten erweitert und auch unterschiedliche Quantisierungen sind für Y und C_B, C_R möglich. Das reduziert vor allem den Quantisierungsfehler bei der Farbartdarstellung.

H.263 V2-Level

In der Praxis ist es sinnvoll verschiedene Codieroptionen zu kombinieren und gemeinsam zu nutzen. Damit jedoch nicht jeder Decoder sämtliche Codieroptionen vorhalten muss, sind in der Version 2 des Standards drei Level[69] definiert, die sinnvolle Kombinationen von Codieroptionen festlegen.

Level 1

- *Advanced INTRA Coding*
- *Deblocking Filter*
- *Supplemental Enhancement Information* (nur *Full-Frame Freeze*)
- *Modified Quantization*

69 Diese Leveldefinition ist mit der 3. Version des Standards H.263 (H.263++) überflüssig geworden, da der Annex X eine komplette *Profiles and Levels Definition* für den Standard festlegt.

Level 2

- *Unrestricted Motion Vectors*
- *Slice Structured*
- *Reference Picture Resampling* (nur *Implicit Factor-of-4 Mode*)

Level 3

- *Advanced Prediction*
- *Improved PB-Frames*
- *Independent Segment Decoding*
- *Alternative INTER VLC*

7.1.5 ITU-T H.263, Version 3 (H.263++) – Codieroptionen

Version 3 des Standards H.263 (H.263++) wurde Ende 2000 von der ITU-T verabschiedet [7.4]. Auch hier geschah die Erweiterung des Standards durch das Hinzufügen von Ergänzungen:

Codieroptionen H.263 V3

- *Enhanced Reference Picture Selection Mode* (Annex U)
- *Data-Partitioned Slice Mode* (Annex V)
- *Additional Supplemental Enhancement Information Specification* (Annex W)

Annex U Enhanced Reference Picture Selection Mode

Im *Enhanced Reference Picture Selection Mode* (Annex U) wird im Decoder ein Speicher für mehrere Bilder vorgehalten. Damit ist es möglich, eine Bewegungskompensation mit mehreren Referenzbildern über eine *Bewegungsspur* hinweg durchführen zu können.

Annex V Data-Partitioned Slice Mode

Mit dem *Data-Partitioned Slice Mode* (Annex V) kann ein höherer Fehlerschutz erreicht werden. Dazu werden die Syntaxelemente, wie Daten-Header, Verschiebungsvektoren und DCT-Koeffizienten voneinander getrennt und mit unterschiedlichen Redundanzverfahren (RLC und VLC) codiert.

Annex W Additional Supplemental Enhancement Information Specification

Mit der *Additional Supplemental Enhancement Information Specification* (Annex W) werden Informationen in den Datenstrom eingeflochten, die eine Rückwärtskompatible Nutzung von Erweiterungen ermöglichen. Eine Spezifikation der inversen DCT auf Festkomma-Arithmetik ist ebenso enthalten wie auch *Picture Messages* (z. B. Beschreibung des Videos, Zeilensprungtyp etc.).

Annex X Profiles & Levels

Im April 2001 wurde eine Erweiterung hinzugefügt, *Profiles and Levels Definition* (Annex X), die aus den vielen Codieroptionen eine umfassende *Profile & Level* Struktur erzeugt, wie sie für vom MPEG-2 Standard her bekannt ist [7.5]. Es sind insgesamt 9 Profiles definiert:

- *Baseline Profile* (P0)
- *H.320 Coding Efficiency Version 2 Backward-Compatibility Profile* (P1)

- *Version 1 Backward-Compatibility Profile* (P2)
- *Version 2 Interactive and Streaming Wireless Profile* (P3)
- *Version 3 Interactive and Streaming Wireless Profile* (P4)
- *Conversational High Compression Profile* (P5)
- *Conversational Internet Profile* (P6)
- *Conversational Interlace Profile* (P7)
- *High Latency Profile* (P8)

Diese Profile sind mehr oder weniger frei mit maximal 7 Level-Stufen kombinierbar.

Der Standard H.263 wurde auch nach der dritten Version noch weiter verfeinert. Die Arbeiten an diesem Standard flossen aber mit der Zusammenlegung der beiden Video Expertengruppen der *ITU* und der *ISO/IEC* in den neuen Standard *ITU-T H.264 ein*, der so auch exakt dem *ISO/IEC* Standard *14496-10 AVC* (*MPEG-4 Part 10*) entspricht. Eine Beschreibung dieses Standards folgt im Abschnitt 7.3.

Selbsttestaufgabe 7.1-1:

Geben Sie die wesentlichen Neuerungen des ITU-T H.263 Baseline Standards gegenüber ITU-T H.261 an!

Selbsttestaufgabe 7.1-2:

Nennen Sie die Hierarchieebenen von ITU-T H.263!

Selbsttestaufgabe 7.1-3:

Nennen Sie die Codieroption, mit der bei ITU-T H.263 ein größerer Bereich für die Bewegungskompensation möglich wird, als bei ITU-T H.261!

Selbsttestaufgabe 7.1-4:

Was versteht man bei ITU-T H.263 unter einem PB-Frame?

Selbsttestaufgabe 7.1-5:

Wozu dienen die Leveldefinitionen bei ITU-T H.263 V2?

Selbsttestaufgabe 7.1-6:

Wie viele Profile- & Level-Stufen sind bei ITU-T H.263 V3 maximal möglich?

7.2 ISO/IEC 14496-2 (MPEG-4 Video)

Die *ISO/IEC-Norm 14496 – MPEG-4* ist kein Audio-Video-Kompressionsstandard im herkömmlichen Sinn. MPEG-4 ist vielmehr ein komplexer und umfassender Standard für interaktive Multimedia-Anwendungen. Nach der Einführung in das Grundkonzept von MPEG-4 (Abschnitt 7.2.1) beschränkt sich diese Kapitel auf die Beschreibung des MPEG-4 Teils, der sich mit der Videocodierung beschäftigt, MPEG-4 Video.

7.2.1 Konzept des MPEG-4 Standards

ISO/IEC 14496 – MPEG-4

Der Standard ISO/IEC 14496 – MPEG-4 – wurde in seiner ersten Version Anfang 1999 verabschiedet, Version 2 folgte Anfang 2000 [7.6]. Obwohl weitere Versionen in Planung sind, werden die bisher fertig gestellten Teile laufend ergänzt und erweitert. Ähnlich wie bei MPEG-2 geschieht die Erweiterung mit neuen Funktionalitäten über die Einführung neuer *Profiles*. Diese werden noch näher in Abschnitt 7.2.6 beschrieben.

Für den Standard sind 15 Teile vorgesehen. Die ersten 6 Teile sind seit 2000 integraler Bestandteil von MPEG-4:

Aufbau des MPEG-4 Standards

- Part 1: *Systems,*
- Part 2: *Visual,*
- Part 3: *Audio,*
- Part 4: *Conformance Testing,*
- Part 5: *Reference Software,*
- Part 6: *Delivery Multimedia Integration Framework* (*DMIF*),
- Part 7: *Optimised Reference Software of Audio-Visual Objects,*
- Part 8: *Carriage of MPEG-4 Contents over IP Networks,*
- Part 9: *Reference Hardware Description,*
- Part 10: *Advanced Video Coding – AVC* (identisch mit *ITU-T H.264/AVC*),
- Part 11: *Scene Description and Application Engine,*
- Part 12: *ISO Base Media File Format,*
- Part 13: *IPMP Extensions,*
- Part 14: *MP4 File Format,*
- Part 15: *AVC File Format.*

Im Rahmen dieses Kurses sind die Teile 2 (*Visual*) und 10 (*Advanced Video Coding – AVC*) von Interesse. Der Teil 10 ist identisch mit dem ITU-T Standard H.264/AVC und wird im Abschnitt 7.3 beschrieben.

Die wesentlichen Eigenschaften von MPEG-4 sind:

Eigenschaften von MPEG-4

- Die MPEG-4 Plattform ist übergreifend einsetzbar (Interoperabilität). MPEG-4 Decoder und -Player sind damit für beliebige Hard- und Softwareumgebungen konzipierbar.
- MPEG-4 arbeitet *objektorientiert*. Das bedeutet, dass eine *Szene* aus unterschiedlichen multimedialen Objektelementen zusammengesetzt sein kann. Mögliche Objekte sind Audio, Video, Bilder, Grafik, Interaktionselemente und Texteinblendungen.
- MPEG-4 ermöglicht erstmals auch die Beschreibung von synthetischen audiovisuellen Objekten in einem gemeinsamen Standard zusammen mit natürlichen, photografischen Objekten. Ein wichtiges Anwendungsgebiet hierfür ist die *Facial Animation* (vgl. Abschnitt 7.2.7).
- Das MPEG-4 Format enthält *Tools*, um skalierbare, beliebig geformte Video-Objekten (*Video-Shapes*) zu beschreiben. Je nach Anwendungszweck kann entweder eine binäre Form (*Maske – Binary Shape*) oder eine Form mit Graustufen (*Gray Scale, Alpha Shape*) verwendet werden. Diese Key- und Stanzfähigkeiten prädestinieren MPEG-4 auch für den Einsatz in virtuellen Studio-Umgebungen.
- Mit einer mächtigen Szenenbeschreibungssprache (*Binary Format for Scenes – BIFS*) innerhalb von MPEG-4 steht Autoren von Multimedia-Anwendungen eine Vielzahl von Gestaltungsmöglichkeiten zur Verfügung. Mit zahlreichen Interaktionselementen kann der Anwender das Erscheinungsbild einer *Szene* frei an seine Vorstellungen anpassen.
- MPEG-4 ist besonders für den Transport multimedialer Inhalte über heterogene Netze auch mit niedriger Bandbreite ausgelegt. Ein offener Kommunikationsstandard (*Delivery Multimedia Integration Framework – DMIF*) macht eine flexible Anpassung an die jeweiligen Netzwerkstrukturen möglich. Besonderes Augenmerk liegt auf den vielfältigen Multiplex- und Synchronisationsmechanismen.

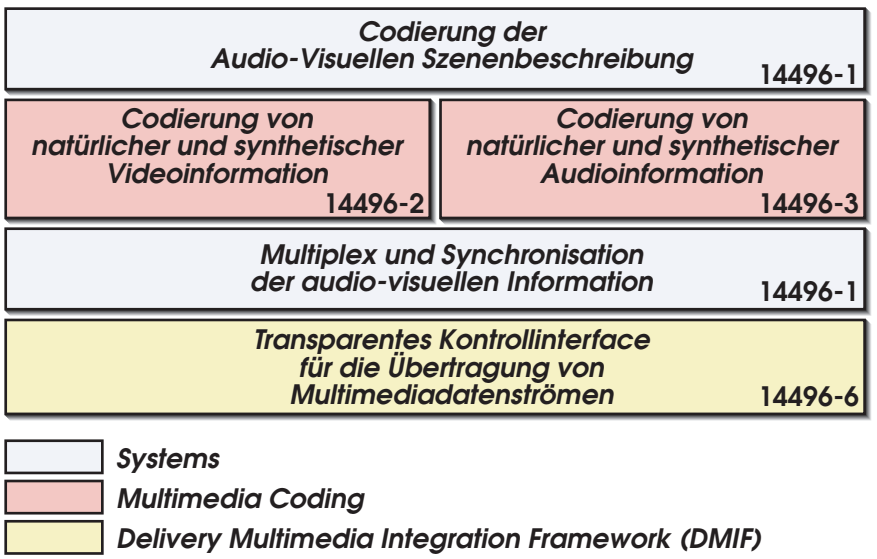


Abb. 7.2-1: Konzeptioneller Aufbau von MPEG-4

Konzeptioneller Aufbau von MPEG-4

Abb. 7.2-1 zeigt das konzeptionelle Zusammenspiel der 4 wichtigsten Teile des Standards. Der Systems-Teil ist für den Datenaustausch zwischen den *Codiermodulen* (Audio und Video) und der *Transportebene* (DMIF) sowie für die audio-visuelle Szenenbeschreibung verantwortlich. Die Codiermodule sorgen für die eigentliche Video- und Audiokompression. Der DMIF-Teil nimmt eine transparente Adaption des MPEG-4 Datenstroms für die unterschiedlichen Netzwerkstrukturen vor.

Komposition einer MPEG-4 Szene

In Abb. 7.2-2 ist die objektorientierte Komposition einer MPEG-4 Szene an einem Beispiel dargestellt. Die Szene wird zusammengesetzt aus (natürlichen und synthetischen) audio-visuellen Objekten, Objekt- und Szenenbeschreibung. Der Anwender hat über Interaktionselemente die Möglichkeit, auf das Szenengeschehen zu reagieren.

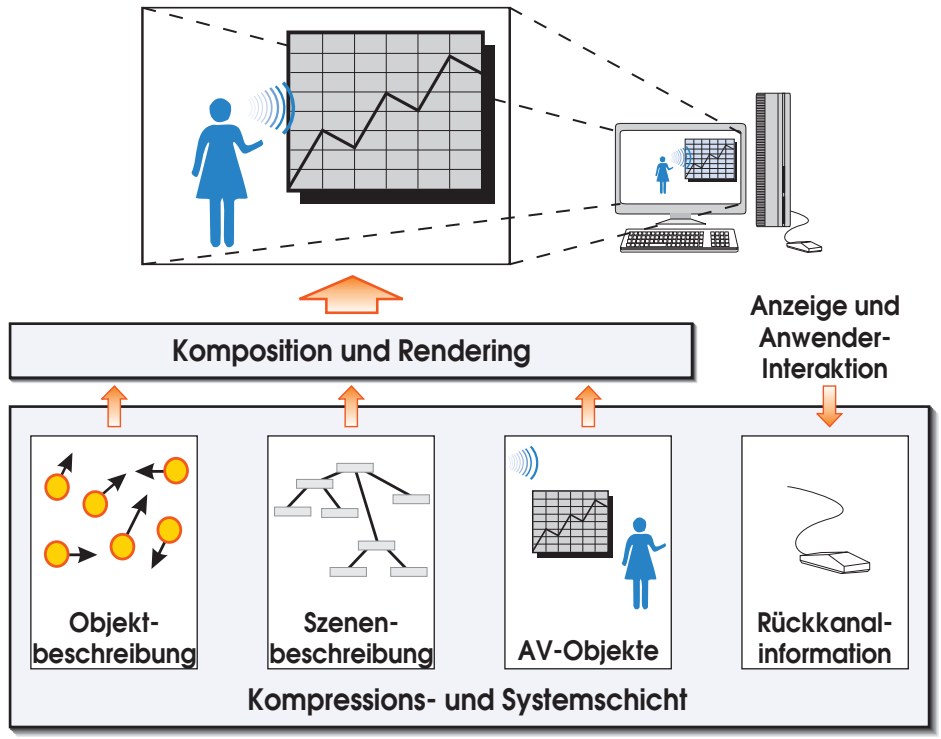


Abb. 7.2-2: Komposition einer MPEG-4 Szene

7.2.2 Leistungsmerkmale von MPEG-4 Video

MPEG-4 folgt in der Version 1 bezüglich des Konzepts zur Video-Datenkompression grob dem Design von ITU-T H.263 (vgl. Abschnitt 7.1). Ergänzt wird die Videocodierung durch die verschiedenen *Trickmodi* und eine Konturcodierung (*Shape Coding*). Weitere Leistungsmerkmale der MPEG-4 Videocodierung sind:

Leistungsmerkmale MPEG-4 Video

- MPEG-4 Video ist abwärtskompatibel zu MPEG-1 und MPEG-2. Damit kann ein MPEG-4 Decoder auch MPEG-1- oder MPEG-2-kompatible Datenströme auswerten und darstellen. MPEG-4 kann sowohl Progressiv- als auch Zeilensprungmaterial verarbeiten.
- Als Farbunterabtastungsformat hat sich auch hier die 4:2:0-Codierung für MPEG-4 Video als zweckmäßig erwiesen. Höherwertige Abtastungen sind nur in besonderen Profiles (z. B. *Studio Profile*) vorgesehen. Die Quantisierung der einzelnen Komponenten kann zwischen 4 und 12 Bit liegen (z. B. *N-Bit Profile*).
- Für die Übertragungsdatenraten von MPEG-4 haben sich drei Bereiche *weniger als 64 kbit/s*, *64 bis 384 kbit/s* und *384 kbit/s bis 4 Mbit/s* herauskristallisiert. Höhere Datenrate sind in besonderen Profiles möglich (z. B. Studio Profile).
- Die Genauigkeit der Bewegungskompensation kann bis zu einem viertel Bildpunkt gesteigert werden. Ein spezielles Interpolationsfilter sorgt für die gewichtete Zuordnung auf das Pixelraster.
- Einzelbildobjekte können nach dem standardmäßigen JPEG-Verfahren oder nach der *Zero-Tree-Wavelet* Methode codiert werden (vgl. Abschnitt 3.4).
- Für die Beschreibung von visuellen Konturen ist eine *segmentierte Codierung* vorgesehen. Die Codierung von synthetischen Videoobjekten geschieht mit eigenen Verfahren, auf die hier nicht weiter eingegangen wird.

7.2.3 Die *Shape-Adaptive DCT* bei MPEG-4 Video

SA-DCT

Die Codierung von beliebig umrandeten Videoobjekten zählt zu den besonderen Merkmalen der MPEG-4 Videocodierung. Mit Hilfe von Masken wird in den zu codierenden Bildblöcken festgelegt, ob bestimmte Bildpunkte zum Objekt gehören oder nicht. Diese Information muss bei der Transformationscodierung ebenfalls berücksichtigt werden. Dazu wurde die so genannte *Shape-Adaptive DCT* (*SA-DCT*) [7.7] entwickelt.

Bei der SA-DCT wird eine kaskadierte Zeilen- und Spaltentransformation angewendet. An einem Beispiel (vgl. Abb. 7.2-3) soll das Verfahren verdeutlicht werden. Abb. 7.2-3 zeigt ein kleines Bildsegment, dass in diesem Fall komplett innerhalb eines 8 x 8 Pixel großen Referenzbildblockes liegt. Die dunkel markierten Pixel sollen das Objekt, die hellen den Hintergrund darstellen. Zunächst werden die Vordergrundpixel vertikal an den oberen Rand des Referenzblockes geschoben (vgl. Abb. 7.2-3). Die einzelnen Spalten werden anschließend eindimensional entsprechend ihrer Ordnung N ($0 < N < 9$) transformiert. Dabei errechnet man die

orthonormierte Form der 1D-DCT für die *N*-langen Objektspalten mit Hilfe der Transformationsmatrix $\underline{A}$:

$$\underline{s}_k = \sqrt{\frac{2}{N}} \cdot \underline{A} \cdot \underline{x}_k \ , \qquad 7.2\text{-}1$$

mit $\underline{s}_k$ als die *k* transformierten Objektspaltenvektoren $\underline{x}_k$. Die Koeffizienten $a_{i,j}$ der Transformationsmatrix $\underline{A}$ errechnen sich zu

$$a_{i,j} = c_0 \cdot \cos\left(\frac{(2i+1)\pi j}{2N}\right), c_0 = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } j = 0 \\ 1 & sonst \end{cases}, \ 0 \le i,j \le N-1. \qquad 7.2\text{-}2$$

Der Vorfaktor

$$c_1 = \sqrt{\frac{2}{N}} \qquad 7.2\text{-}3$$

in Gl. 7.2-1 dient dazu, die Transformation zu *normalisieren*. Abb. 7.2-3 zeigt die Position der Koeffizienten nach der vertikalen SA-DCT. Man beachte, dass die DC-Koeffizienten nach den einzelnen Spaltentransformationen immer in der oberen Reihe des 8 x 8 Referenzblockes liegen (markiert).

Beispiel SA-DCT

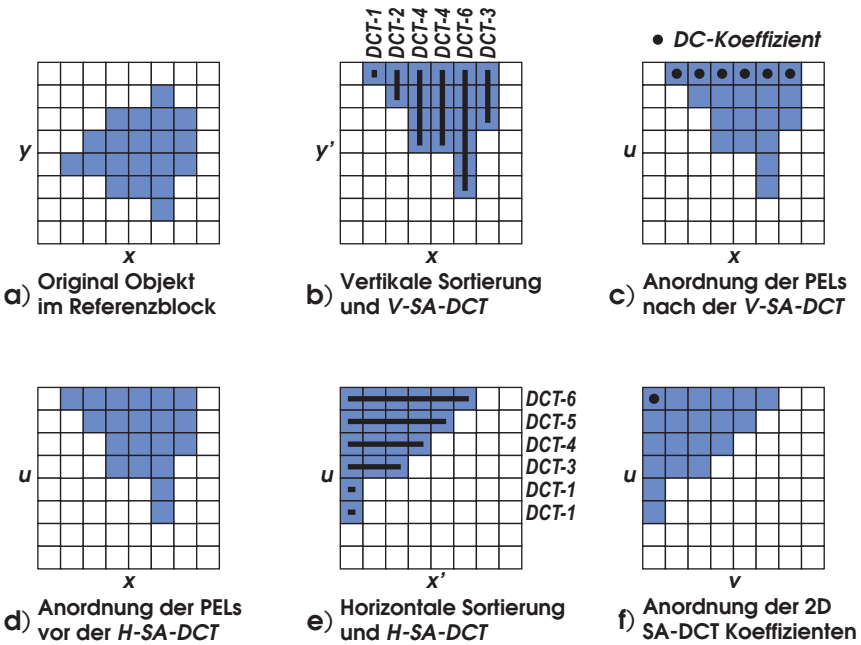


Abb. 7.2-3: Veranschaulichung der SA-DCT an einem Beispiel

Für die anschließende horizontale SA-DCT werden die transformierten Koeffizienten linksbündig an den linken Rand des Referenzblockes verschoben (Abb. 7.2-3). Mit der gleichen Formel werden jetzt auch alle Zeilen mit wenigstens einem Koeffizienten transformiert. Man beachte hier, dass in jeder Zeile die Koeffizienten gemeinsam transformiert werden, die schon den gleichen DCT-Index bei

der vertikalen SA-DCT erhalten haben. So wird beispielsweise die oberste Zeile mit den DC-Koeffizienten der vertikalen Transformation in einer eindimensionalen Zeilen-Transformation verarbeitet. Das Ergebnis der gesamten Transformation ist in Abb. 7.2-3 zu sehen. Man erkennt, dass der Algorithmus für eine ähnliche Anordnung der Koeffizienten sorgt, wie man es von einer normalen DCT-Transformation her gewohnt ist. Der DC-Koeffizient wird links oben positioniert. Bereiche, die nicht zum Objekt selber gehören werden in der SA-DCT Matrix explizit zu Null gesetzt, bevor eine Quantisierung und Zusammenfassung der Koeffizienten beispielsweise mit dem Zick-Zack-Scanning vorgenommen wird.

Rücktransformation SA-IDCT

Da die Kontur jedes Blockes mit übertragen wird, kann nach der zeilen- und spaltenweisen *SA-IDCT* mit der Rückverschiebung der Koeffizienten bzw. Bildpunkte eine exakte Rekonstruktion des Ausgangsbildes des Blockes erfolgen. Für die eindimensionale Rücktransformation benötigt man die Transponierte der Transformationsmatrix $\underline{A}$:

$$\underline{x}_k = \sqrt{\frac{2}{N}} \cdot \underline{A}^T \cdot \underline{s}_k \ . \qquad 7.2\text{-}4$$

Der Vektor $\underline{s}_k$ stellt dabei den horizontalen bzw. vertikalen Koeffizientenvektor der Größe *N* dar.

7.2.4 Strukturebenen bei MPEG-4 Video

Strukturebenen MPEG-4 Video

Visuelle Objekte werden in 5 Ebenen strukturiert.

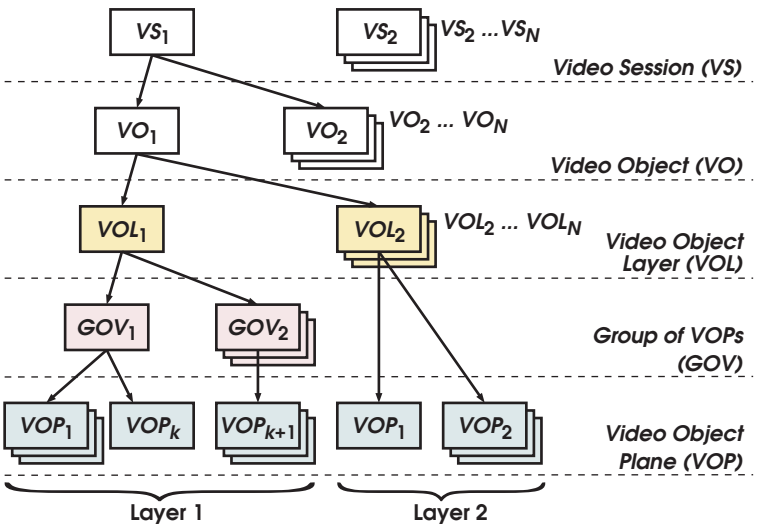


Abb. 7.2-4: Beispiel für die Strukturebenen bei der MPEG-4 Video-Codierung

In der obersten Ebene – *Video Session* (*VS*) – wird die komplette, photografische 2D- oder 3D- Szene dargestellt. Sie besteht aus einem oder mehreren *Video Objects* (*VO*), die mit den visuellen 2D-Objekten in der Szene korrespondieren. Diese Objekte können eine rechteckige oder beliebig umrandete Kontur annehmen.

Die mögliche Skalierbarkeit bei MPEG-4 erlaubt es, dass ein Videoobjekt selbst auf mehrere, unterschiedlich örtlich und/oder zeitlich skalierte visuelle Objekte der darunter liegenden Ebene verweist. Diese Ebene wird *Video Object Layer* (*VOL*) genannt. In der *Group of VOPs* Ebene (*GOV*) werden Zugriffspunkte auf den Datenstrom festgelegt. Damit ist die Auswahl von einzelnen Video Elementen möglich. Der *GOV-Layer* ist optional. So können die eigentlichen codierten Einzelbilder der *Video Object Plane* (*VOP*) auch direkt an den *Video Object Layer* weitergereicht werden. Abb. 7.2-4 zeigt den hierarchischen Aufbau der Strukturebenen für visuelle Objekte an einem Beispiel.

7.2.5 Funktionsdiagramm des MPEG-4 Videoencoders

Generischer MPEG-4 Encoder

MPEG-4 Basis Encoder

Eine wesentliche Besonderheit von MPEG-4 Video ist die objektbasierte Videocodierung. Dieser so genannte *generische MPEG-4 Encoder* lässt sich mit Hilfe eines *MPEG-4 Basis Encoders* realisieren, der durch eine formangepasste Codierung (*Konturcodierung*) ergänzt wird. Der MPEG-4 Basis Encoder entspricht im wesentlichen der schon von MPEG-1 bekannten konventionellen Hybridcodierung, bei der die Textur (*Bildinhalt*) (I-Bild) bzw. das Prädiktionsfehlerbild (B- oder P-Bild) und die zugehörige Bewegungsinformation übertragen werden. Der MPEG-4 Basis Encoder verarbeitet konventionelles Videomaterial in rechteckigem Format.

Die vorgeschaltete Konturcodierung erweitert den MPEG-4 Basis Encoder, so dass auch die Verarbeitung von beliebig umrandeten Objekten möglich wird (vgl. Abb. 7.2-5).

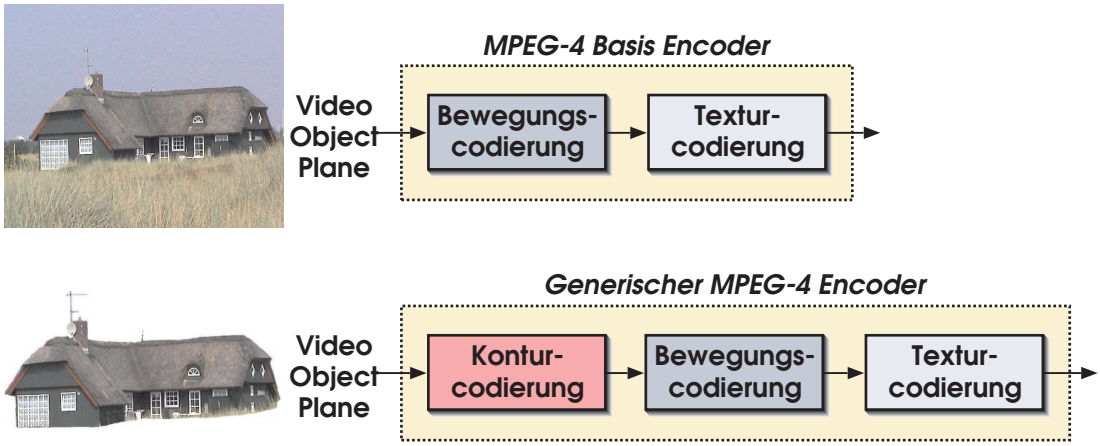


Abb. 7.2-5: Erweiterung des MPEG-4 Basis Encoders durch eine Konturcodierung

Das nachfolgende Funktionsdiagramm (Abb. 7.2-6) zeigt den generischen MPEG-4 Encoder im Detail, bestehend aus den wesentlichen Elementen, *Konturcodierung*, *Bewegungskompensation*, *Codierung des Prädiktionsfehlers* und einem abschließenden *Datenmultiplexer*.

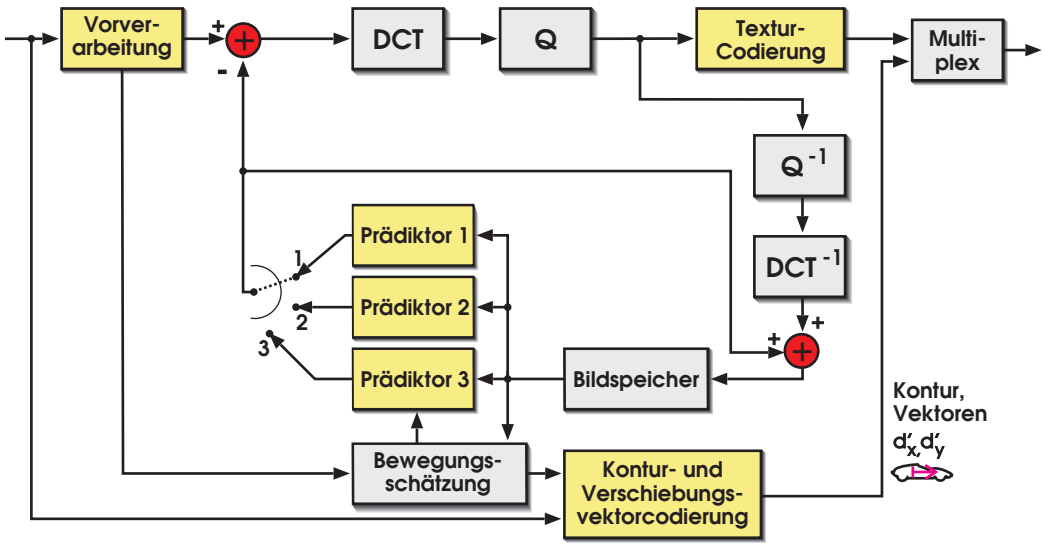


Abb. 7.2-6: Funktionsdiagramm für die formangepasste MPEG-4 Video-Codierung

Formcodierung

Segmentierte Codierung

In der Regel bestimmt eine *Binärmaske* die Form eines Videoobjekts. Für die Codierung dieser *Masken* haben sich verschiedene Verfahren bewährt. In der Regel wird ein kontextadaptives arithmetisches Verfahren (*segmentierte Codierung*) verwendet, bei dem auch zwischen Intra- und zeitlich prädiktiver Codierung unterschieden werden kann. Alternativ zur Binärmaske ist auch eine 8 Bit aufgelösende Transparenzmaske möglich. Diese wird wie der Bildinhalt selbst als Textur codiert.

Bewegungsschätzung und Kompensation

Wie bei bisherigen Codierverfahren wird auch bei MPEG-4 das Videoobjekt in ein regelmäßiges Blockraster eingeteilt. Damit kann die bekannte blockbasierte Bewegungsschätzung und -kompensation angewendet werden, um die zeitliche Korrelationen zwischen aufeinanderfolgenden VOPs zu nutzen. Der Prädiktionsfehler wird ähnlich wie bei ITU-T H.263 codiert: Es sind 8 x 8 oder 16 x 16 Pixel große Makroblöcke möglich. Auch eine Bewegungskompensation mit überlappenden Nachbarblöcken (*Overlapped Block Motion Compensation*) kann durchgeführt werden, wie sie schon aus Annex F des ITU-T H.263 Standards her bekannt ist.

Polygonmatching

Da auch nichtrechteckige Bildobjekte zugelassen sind, muss die herkömmliche Bewegungskompensation etwas modifiziert werden: Zum einen ist das Videoobjekt im Außenbereich vor der Bewegungskompensation mit Bildinformation zu ergänzen, damit auch auf dem Rand des Objekts liegende Bildblöcke als sinnvolle Referenz verwendet werden können. Als zweites hat sich bei der Bewegungsschätzung für die Randblöcke das so genannte *Polygonmatching* bewährt, bei dem für die Bestimmung eines geeigneten Verschiebungsvektors nur diejenigen Bildpunkte innerhalb des Blocks herangezogen werden, die gleichzeitig auch zum Videoobjekt gehören. Die drei bekannten Codierverfahren (*Prädiktionsmodi*) von MPEG-2 für I-, P- und B-Blöcke gibt es auch bei der MPEG-4 Videocodierung.

Texturcodierung

Intraframe-codierten VOPs und der Prädiktionsfehler in P- und B-Blöcken werden auf der Basis von 8 x 8 Bildblöcken DCT-transformiert. Wegen der Farbunterabtastung von 4:2:0 werden normalerweise in einem Makroblock vier benachbarte 8 x 8 Luminanzblöcke und zwei Chrominanzblöcke (C_B, C_R) zusammengefasst. Videoobjekte mit nichtrechteckiger Kontur werden mit der in Abschnitt 7.2.3 beschriebenen *SA-DCT* transformiert.

Multiplexer

Die komprimierte Kontur, die geschätzten Verschiebungsvektoren und die DCT-Koeffizienten des Prädiktionsfehlers mit den zugehörigen Lauflängen werden in einem gemeinsamen Datenstrom zusammengefasst (*Multiplex*). Die zugrunde liegende Syntax hierfür ist auf Makroblock-Ebene der Syntax des ITU-T H.263 Standards sehr ähnlich. Auch die dort bereits bewährten Codewort-Tabellen für die DCT-Koeffizienten und Verschiebungsvektoren werden so auch bei MPEG-4 verwendet. Es werden sogar einige Codieroptionen von ITU-T H.263++ unterstützt, so beispielsweise die Möglichkeit, Verschiebungsvektoren und DCT-Koeffizienten getrennt voneinander zu codieren, um einen erhöhten Fehlerschutz zu erreichen (äquivalent zu ITU-T H.263, Annex V *Data-Partitioned Slice Mode*).

7.2.6 Visuelle Profiles und Levels von MPEG-4

Die jeweils unterstützte Leistungsfähigkeit wird auch bei MPEG-4 in *Profiles* und *Levels* spezifiziert. Da MPEG-4 Video in zwei Stufen standardisiert wurde, existieren – ähnlich wie bei den verschiedenen ITU-T H.263-Entwicklungen – auch bei MPEG-4 Video zwei Versionen. Nachfolgend werden die wesentlichen Parameter der wichtigsten Profiles und Levels vorgestellt.

Für MPEG-2 gibt es eine recht überschaubare Anzahl von Profiles und Levels (vgl. Abschnitt 5.3.2). Bei MPEG-4 Video liegen die Verhältnisse insofern anders, als dass auch die verschiedenen Objekttypen (synthetisch generierte Objekte, beliebige Kontur etc.) bei den Funktionalitäten der Profiles mit berücksichtigt werden müssen.

MPEG-4 – Version 1

MPEG-4 Version 1 Profiles

Beschränkt man sich auf natürliche Videoobjekte, dann unterscheidet man bei *MPEG-4 Profiles for Natural Video, Version 1* zwischen fünf verschiedenen Profiles:

Simple Visual Profile (SP)

Simple Der MPEG-4 SP ist im wesentlichen zu ITU-T H.263 Baseline kompatibel. Der *Picture Header* wird im Datenstrom geringfügig verändert, damit beliebige Bildauflösungen codiert werden können. Weiterhin werden folgende Codieroptionen bzw. Äquivalente des ITU-T H.263 Standards verwendet:

- Erweiterung des Bewegungsvektorraums (vgl. *Annex D – Unrestricted Motion Vector Mode*),
- Überlappende Bewegungskompensation (vgl. *Annex F – Advanced Prediction Mode*),
- Schleifenfilter (vgl. *Annex J – Deblocking Filter*),
- örtliche Intraframe-Prädiktion (ähnlich *Annex I – Advanced Intra Coding*),
- nichtlineare Quantisierungsstufen für den DC-Koeffizienten,
- Möglichkeit, eine örtlich relative Platzierung von rechteckigen Bildern vornehmen zu können (vergleichbar mit *Annex R – Independent Segment Decoding Mode*),
- wahlweise Gruppierung von Makroblöcken (wie *Annex K – Slice Structured Mode*),
- wahlweise Strukturierung und Gruppierung der Syntaxelemente (ähnlich wie *Annex V – Data-Partitioned Slice Mode*).

Der *SP* eignet sich für die Codierung und die Übertragung von rechteckig begrenztem Videomaterial. Ausgelegt ist dieses Profile für Anwendungen in mobilen Netzen.

Core Visual Profile (CP)

Core Baiserend auf dem SP ist beim CP zusätzlich die Codierung von zeitlich variablen Objekten mit beliebiger Kontur möglich. Diese Eigenschaft ist das besondere Merkmal von MPEG-4 und unterscheidet sich darin auch von den sehr beschränkten Möglichkeiten bei H.263, mit der Hilfe von *Chroma Keying* (vgl. *Annex L – Supplemental Enhancement Information*) eine beliebige Objektkontur beschreiben zu können. Die Codierung von zeitlich skalierbaren B-Bildern und P-Bildern wurden dem Standard ITU-T H.263 (Annex O und N) entnommen.

Main Visual Profile (MP)

Main Über die Leistungsfähigkeit des CP hinaus lässt sich mit dem MP auch Videomaterial verarbeiten, dass aus Zeilensprungmaterial entstanden ist. Weiterhin können halbtransparente Objekte codiert werden. Neu ist auch die Möglichkeit, statische Hintergrundbilder (auch skaliert oder transformiert) zusätzlich einbetten zu können. Man spricht hierbei von *Static Sprite Coding*. Dieses Profile eignet sich für interaktive Anwendungen und bedingt auch für Broadcastdienste (Digital-TV, DVD-Video etc.).

Simple Scalable Profile (SSP)

Simple Scalable

Beim *SSP* ist zusätzlich zum SP eine zeitliche und örtliche Skalierbarkeit der Videodaten möglich, ähnlich wie sie auch schon bei MPEG-2 in zwei Profiles vorgesehen ist. Die Skalierbarkeit ist beispielsweise für Dienste geeignet, die unterschiedliche Qualitätsstufen anbieten können.

N-Bit Visual Profile (N-Bit)

N-Bit

Ergänzend zum CP können die Komponenten *Y*, $C_B$und C_R mit einer höheren Bittiefe (bis 12 Bit) oder einer niedrigeren Bittiefe (mindestens 4 Bit) codiert werden. Die höhere Bittiefe kann beispielsweise für bildgebende Verfahren in der Medizin verwendet werden.

MPEG-4 – Version 2

MPEG-4 Version 2 Profiles

In der Version 2 des MPEG-4 Video Standards wurde die Codiereffizienz noch einmal gesteigert. Dies konnte dadurch erreicht werden, indem die Genauigkeit der Bewegungskompensation auf ein viertel Pixel erhöht wurde, eine Bewegungskompensation für globale Bewegungen eingeführt wurde (Schwenk, Zoom etc.), die SA-DCT für die Transformation von beliebigen Objektkonturen verwendet wird (vgl. Abschnitt 7.2.3) und eine optionale Reduktion der örtlichen oder zeitlichen Auflösung von Makroblöcken (ähnlich wie H.263, *Annex Q – Reduced Resolution Update Mode*) möglich ist. Für die Verbesserung des Fehlerschutzes kann darüber hinaus mit einer Prädiktion auf der Basis von mehreren Referenzbildern geschehen. Dieses Verfahren existiert so auch im Standard ITU-T H.263, Annex N (*Reference Picture Selection Mode*).

Mit diesen Erweiterungen wurden sieben neue *Visual Profiles* zum MPEG-4 Standard Version 2 hinzugefügt (vgl. [7.8], [7.9]).

Advanced Real-Time Simple Profile (ARTS)

ARTS

Dieses Profile baut auf den Simple Profile auf und ist für Echtzeit Übertragungen ausgelegt. Dazu wird über eine Rückkopplung adaptiv die örtliche und zeitliche Auflösung an die Empfangsbedingungen am Decoder angepasst. So kann die Datenrate kontinuierlich an den momentanen Netzwerkbedingungen angepasst werden.

Core Scalable Profile (CSP)

Core Scalable

Das Core Visual Profile wird erweitert, in dem beliebige Objektkonturen zeitlich und örtlich skalierbar codiert werden können. Die Skalierbarkeit bezieht sich wie bei den entsprechenden Profiles von MPEG-2 auf den verfügbaren Störabstand (SNR) bzw. auf die verfügbare Übertragungsbandbreite.

Advanced Coding Efficiency Profile (ACE)

ACE Der ACE nutzt in besonderer Weise die o. g. Erweiterungen zur Verbesserung der Codiereffizienz von MPEG-4. Es können rechteckige oder beliebig umrandete Objektkonturen verarbeitet werden.

Studio Profiles (SStP) und Core Studio Profile (CStP) Diese beiden Profiles wurden 2001 dem MPEG-4 Standard hinzugefügt. Sie erweitern den Standard um anwendungsgerechte Funktionalitäten für den professionellen Schnitt (*Editing*) und die Postproduktion mit Datenraten bis zu 2 Gbit/s. Dabei ist eine reine I-Bild-Codierung möglich, die bei Bedarf mit einer P-Bild-Codierung, einer Konturcodierung und erstmals auch mit Transparenzkanälen (Alphakanäle) ergänzt werden kann.

Advanced Simple Visual Profile (ASP)

ASP Der ASP baut auf den Funktionalitäten des Simple Profiles auf. Mit den bekannten Maßnahmen der Version 2 des MPEG-4 Standards wird die Codiereffizienz gesteigert. Zusätzlich ist auch die Bewegungskompensation über B-Bilder möglich. Ausgeschlossen sind aber alle anderen besonderen Funktionalitäten von MPEG-4, wie die Konturcodierung und die Interaktion oder Manipulation von Objekten. Damit ist dieses Profile besonders als effizienter Codierersatz für Anwendungen zu sehen, die bisher MPEG-1 oder MPEG-2 verwendet haben. Als erfolgreichster Vertreter verwendet der *DivX5-Codec* den ASP [7.10]. Zudem hält der Trend an, DVD-Abspielgeräte mit der Fähigkeit auszustatten, *MPEG-4 ASP* codierte Videodaten decodieren können.

Fine Grain Scalability bzw. Streaming Profile (FGS)

FGS Der FGS ist besonders für Übertragungskanäle ausgelegt, bei denen mit stark schwankenden Datenraten zu rechnen ist. Dazu werden die Tools zur Skalierbarkeit von MPEG-4 angewendet. Dynamisch wird die örtliche und zeitliche Auflösung der Videodaten an die tatsächliche Übertragungskapazität angepasst. Das geschieht über entsprechende Rückmeldungen des Decoders (z. B. Paketverzögerungen und -verluste). Die prädestinierte Anwendung für dieses Profile ist das Internet-Streaming.

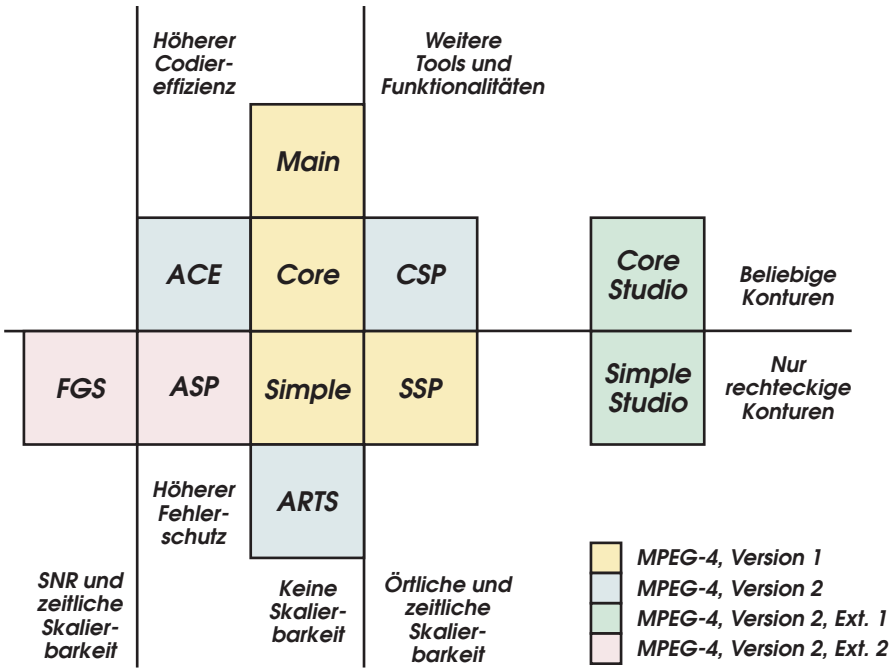


Abb. 7.2-7: Funktionalitäten bei den *MPEG-4 Profiles for Natural Video*

Abb. 7.2-7 visualisiert die Funktionalitäten der angesprochenen MPEG-4 Profiles. **Übersicht Profiles**

MPEG-4 – Levels

MPEG-4 kennt je nach Profile die Level-Stufen *L0* bis *L5*. Wie bei den bisherigen MPEG-Standards repräsentieren auch bei MPEG-4 die Levels verschiedenen Auflösung- bzw. Qualitätsstufen (z. B. maximale Datenrate). Tabelle 7.2-1 skizziert die Profile-Level Kombinationen für die Profiles *Simple*, *Core*, *Main* und *ACE* sowie die zugehörigen Auflösungen, die maximalen Datenraten und die Anzahl von möglichen Objekten im Datenstrom. **Levels für Profiles Simple, Core, Main und ACE**

Tab. 7.2-1: *Profile-Level Kombinationen für Simple, Core, Main und ACE Auflösungen:*
QCIF (176 x 144), CIF (352 x 288), ITU601 (720 x 576), ITU709 (1920 x 1080):
Funktionalitäten bei den MPEG-4 Profiles for Natural Video

Profile	Parameter	***Level***				
		L0	***L1***	***L2***	***L3***	***L4***
Simple	Auflösung	***QCIF***	***QCIF***	***CIF***	***CIF***	
	R_{max} [kbit/s]	64	64	128	384	
	max. Obj.	1	4	4	4	
Core	Auflösung		***QCIF***	***CIF***		
	R_{max} [kbit/s]		384	2000		
	max. Obj.		4	16		
Main	Auflösung			***CIF***	***ITU601***	***ITU709***
	R_{max} [kbit/s]			2000	15000	38400
	max. Obj.			16	32	32
ACE	Auflösung		***CIF***	***CIF***	***ITU601***	***ITU709***
	R_{max} [kbit/s]		384	2000	15000	38400
	max. Obj.		4	16	32	32

Ersichtlich sind die Parameter für einen Level bei verschiedenen Profiles nicht identisch. Beachtenswert ist auch, dass der Simple- und der Core-Profile nicht für die Standardauflösungen im Studio nach ITU-R BT.601 (720 x 576 Pixel) ausreichen. Die Main- und ACE-Profiles können jedoch auch mit HD-Bildmaterial nach ITU-R BT.709 zurecht kommen.

Levels für Profiles Studio, ASP und FGS

Tabelle 7.2-2 zeigt die Profiles für Studioumgebungen (*Simple Studio* und *Core Studio Profile*), für Standard Videoanwendungen mit hoher Codiereffizienz (*Advanced Simple Profile – ACE*) und für Internet-Streaming Anwendungen (*Fine Grain Scalability – FGS*).

Tab. 7.2-2: *Profile-Level Kombinationen für Simple Studio, Core Studio, ASP und FGS*
Auflösungen: QCIF (176 x 144), CIF (352 x 288), S-VCD (352 x 576), ITU601 (720 x 576), ITU709 (1920 x 1080), 2k x 2k (2000 x 2000)

Profile	Parameter	***Level***					
		L0	***L1***	***L2***	***L3***	***L4***	***L5***
Simple Studio	Auflösung		***ITU601***	***ITU709***	***ITU709***	2k x 2k	
	R_{max} [kbit/s]		180	600	900	1800	
	max. Obj.		1	1	1	1	
Core Studio	Auflösung		***ITU601***	***ITU709***	***ITU709***	2k x 2k	
	R_{max} [kbit/s]		90	300	450	900	
	max. Obj.		4	4	8	16	
ASP/FGS	Auflösung	QCIF	QCIF	***CIF***	***CIF***	***S-VCD***	***ITU601***
	R_{max} [kbit/s]	128	128	384	768	3000	8000
	max. Obj.	1	4	4	4	4	4

Die Studio Profiles können beim Level *L1* neben dem 4:2:2 Studio Signal ein zusätzliches Signal mit voller Auflösung übertragen (4:2:2:4), beispielsweise ein Masken- oder Alphakanalsignal. Beim Level *L2* sind alternativ zum HD-Signal sogar zwei 4:4:4 RGB-Signale verarbeitbar. Damit ist die Übertragung eines stereoskopischen Videosignals mit hoher Ortsauflösung möglich. Der Level *L3* kann entsprechend dem Level *L1* auch einen zusätzlichen Alphakanal verarbeiten, dieses aber auch mit HD-Auflösung. Entsprechendes gilt für den Level *L4*, bei dem bis zu sechs hochauflösende Kanäle in HD-Auflösung untergebracht werden können.

Für die *ASP*- und *FGS*-Profiles sind jeweils 6 Levelstufen definiert. Dieses Profiles eignen sich auch für ein breites Anwendungsfeld.

7.2.7 Beispiel für synthetisch generierte visuelle Objekte bei MPEG-4

In Abschnitt 7.2.1 wurde ein kurzer Überblick über die Besonderheiten des MPEG-4 Standards gegeben. Dabei wurde darauf hingewiesen, dass MPEG-4 sich in besonderer Weise dazu eignet, multimediale Inhalte in interaktiven Umgebungen zu beschreiben. Hierfür existiert u. a. eine Szenenbeschreibungssprache, welche die Beziehung und die möglichen Interaktionen zwischen den MPEG-4 Objekten koordiniert. Eine solche Beschreibung ist aber nicht allein auf die örtlich-zeitliche Korrelation zwischen Objekten (*Elementary Streams*) beschränkt. Vielmehr besteht bei MPEG-4 Video die Möglichkeit, bewegte Objekte synthetisch selbst zu generieren und ihre zeitliche Veränderung zu beschreiben. Auch eine Kombination aus synthetischen und natürlichen Objekten ist denkbar.

Synthetic and Natural Hybrid Coding - SNHC

Sehr häufig wird die Kombination von natürlichen und synthetischen Objekten für die *Animation* von *natürlichen Objekten* verwendet (*Synthetic and Natural Hybrid*

Coding – SNHC). Ein wichtiges Beispiel ist die synthetische örtlich-zeitliche Beschreibung der menschlichen Gesichtszüge in Kombination mit synthetisch generierter Sprache. Dieses Beispiel wird insbesondere bei der Konzeption von *Avataren* zu Informationszwecken oder für gestalterische Anwendungen genutzt. Eine besonders datensparende Konstruktion dieses Beispiels wird möglich, wenn es gelingt, mit möglichst wenigen Parametern ein am Empfänger vorliegendes natürliches Gesicht in seiner Mimik hinreichend gut zu beschreiben. Ein paar Grundzüge dieser so genannten *Facial Animation – FA* von MPEG-4 sind im folgenden Abschnitt kurz skizziert.

Facial Animation – FA

Facial Animation

In der Computeranimation ist die 3D-Beschreibung von Objekten über *Drahtgittermodelle* gebräuchlich. Bei diesem als *Triangulierung* bezeichnete Vorgang wird das natürliche Objekt mit vielen im Raum angeordneten Dreiecksflächen angenähert. Je nach Beschaffenheit und geforderter Genauigkeit der Approximation an die 3D-Kontur des Objekts ist eine mehr oder minder große Anzahl von Dreiecken zur Beschreibung des Objekts notwendig. Abb. 7.2-8 zeigt hierzu ein Beispiel.

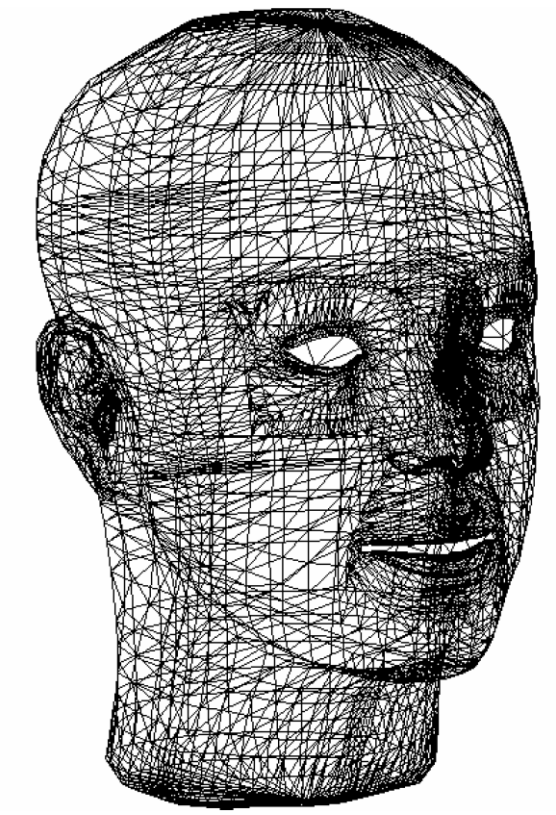


Abb. 7.2-8: Beispiel für die Triangulierung des menschlichen Kopfes

Triangulierung

Für die Bewegungsanimation einer solchen Triangulierung ist es erforderlich, den Eckpunkten der Dreiecke eine zeitlich-örtliche Verschiebungsinformation zuzuordnen. Eine Beschreibung bei komplexen Objekten mit vielen tausend Dreiecken ist aber nicht effizient codierbar, wenn die Verschiebung jedes Eckpunktes übertragen

werden soll. Daher wird bei MPEG-4 die Beschreibung der Mimik durch eine Verschiebungsinformation nur für maximal 84 so genannte *Feature Points – FP* definiert. Zudem sind einige FPs grundsätzlich fest, andere können ihre Lage je nach Mimik verändern. Abb. 7.2-9 zeigt die bei MPEG-4 verwendeten FPs beim menschlichen Gesicht.

Feature Points

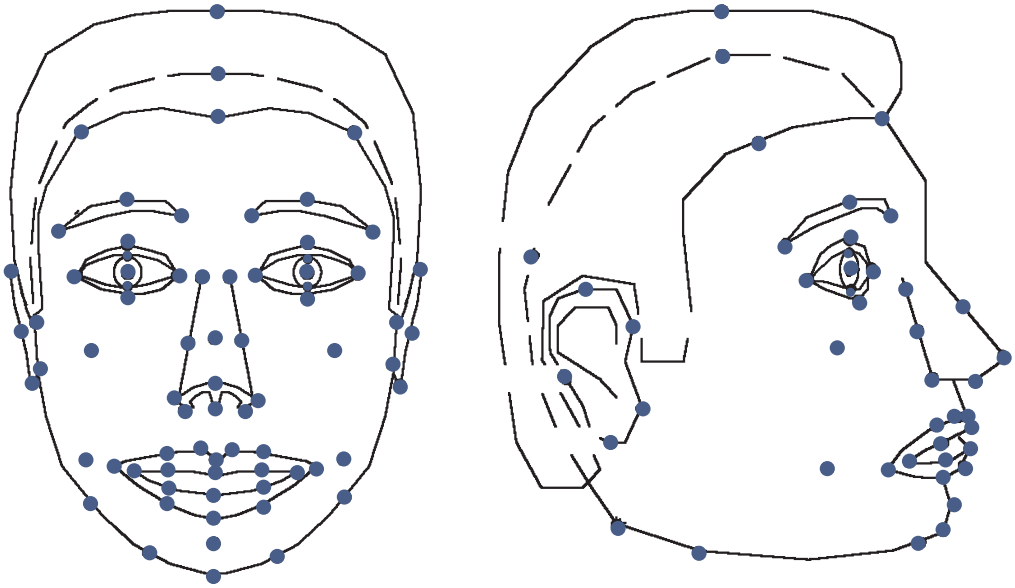


Abb. 7.2-9: *Feature Points* der *Facial Animation* von MPEG-4

Die Textur[70] des Gesichts und die FPs werden weiterhin zu so genannten *Facial Definition Parameters – FDPs* zusammengefasst. Da es für die unterschiedlichen Gesichtsausdrücke nicht unbedingt erforderlich ist, immer für jeden FP eine Bewegungsbeschreibung zu codieren, sind in MPEG-4 auch für die Animation insgesamt 68 so genannte *Facial Animation Parameter – FAP* definiert. Diese repräsentieren auf unterschiedlichen Abstraktionsebenen den Gesichtsausdruck. Ein Lächeln ist damit mit sehr wenigen Parametern eindeutig beschreibbar. Verständlicherweise müssen Abstriche bei der Individualität des Gesichtsausdrucks gemacht werden. Wenn man allerdings an die drei Haupteinsatzgebiete der Face Animation denkt, nämlich

Facial Definiton Parameter – FDP
Facial Animation Parameter – FAP

- Darstellung und Animation von Avataren,
- Text-zu-Sprache Umsetzung mit visueller Unterstützung,
- Analyse-Synthese-Codierung zur Datenreduktion durch Parametrisierung realer Personenbewegungen (z. B. für die Spieleentwicklung),

dann erscheint der Qualitätsverlust vertretbar.

70 Eine Textur für eine Facial Animation lässt sich beispielsweise aus einer 360° Abbildung eines Kopfes und einer entsprechenden Texture Map errechnen.

MPEG-4 Profiles mit synthetischen Objekten

Im Abschnitt 7.2.6 wurden die *Visual Natural Profiles* von MPEG-4 vorgestellt. Mit diesen können natürliche, visuelle Objekte mit rechteckigen oder beliebigen Konturen beschrieben werden. Weitere Profiles sind in MPEG-4 erforderlich, wenn zusätzlich auch synthetische, visuelle Objekte berücksichtigt werden sollen.

Weitere MPEG-4 Profiles

Der *Simple Facial Animation Profile* ist der einfachste Profile, der mit der Animation eines FAP-Datenstroms umgehen kann. Im Level *L1* kann ein Gesicht bei einer Renderfrequenz von 15 Hz mit nur 16 kbit/s beschrieben werden. Bis zu 4 Gesichter sind im Level 2 möglich. Die weiteren Profiles auf diesem Anwendungsgebiet sind das *Scalable Texture Visual Profile*, das *Basic Animated 2D Texture Visual Profile* und das *Hybrid Visual Profile*. In der Version 2 von MPEG-4 wurde die Ganzkörperanimation im *Simple Face and Body Animation Profile* hinzugefügt. Überdies gibt es seit dem auch zwei Profiles, die eine Einzelbildcodierung mit der Wavelet-Transformation (vgl. Kapitel 3) ermöglichen, das *Advanced Scalable Texture Profile* und das *Advanced Core Profile*. Eine Beschreibung dieser Profiles findet sich beispielsweise in [7.11] oder [7.12].

7.2.8 Abschließende Bemerkungen

MPEG-4 wurde ursprünglich ausschließlich für Anwendungen mit niedrigen Datenraten ausgelegt. Wie aber nicht zuletzt die Definition der *Studio Profiles* zeigt, wird MPEG-4 zunehmend auch für höhere Datenraten und für eine höhere Bildqualität in Standardanwendungen eingesetzt. Ansätze, MPEG-4 beispielsweise für die nächste DVD-Generation (*HD-DVD*) zu nutzen, sind recht vielversprechend, da HD-Videomaterial auf den bisherigen Datenträgern vertrieben werden kann. Die Beliebtheit des DivX-Codecs (MPEG-4 ASP kompatibel) zeigt, dass hierfür Märkte vorhanden sind. Insgesamt ist der mögliche Einsatzbereich für MPEG-4 tatsächlich recht vielfältig, wie die nachfolgende Zusammenfassung der Anwendungsbeispiele zeigt:

- Virtuelle Telekonferenz,
- Digitales Fernsehen in Standard- oder HD-Auflösung,
- Mobile Multimediaanwendungen (geringe Datenrate),
- Spiele (Interaktion, multimediale Elemente),
- Streaming Video, z. B. für das Internet (Anpassung an Netzstruktur, hohe Codiereffizienz, Interaktion),
- Ergänzte Realität (*Augmented Reality* z. B. für den Einsatz in der Lehre),
- synthetische Generierung einer begleitenden Gebärdensprache für Hörgeschädigte,
- Einsatz im Bereich der Verkehrsinformation.

Obwohl die Liste sicherlich noch vielfältig erweitert werden könnte, hat sich MPEG-4 als Standard bislang nur in wenigen Bereichen durchsetzen können. Das

Hauptproblem bei der Verbreitung von MPEG-4 ist wohl darin zu sehen, dass proprietäre Multimedia-Standards die Einführung der besonderen MPEG-4 Funktionalitäten erschweren. Das Hauptaugenmerk vieler Firmen in Bezug auf MPEG-4 ist die hohe Effizienz der Videocodierung. Damit wird MPEG-4 oft auf diesen Punkt reduziert. Der Trend bei der Weiterentwicklung der Videocodierung zeigt aber, dass noch weiter Bedarf an einer höheren Codiereffizienz besteht. Das folgende Kapitel beschäftigt sich mit der nächsten Generation der Videocodierung: *MPEG-4/AVC* bzw. *ITU-T H.264/AVC*.

Selbsttestaufgabe 7.2-1:

Welches ist die besondere Eigenschaft von MPEG-4 Video in Bezug auf die Codierung von Objektkonturen?

Selbsttestaufgabe 7.2-2:

Was versteht man unter BIFS?

Selbsttestaufgabe 7.2-3:

Welche Genauigkeit kann die Bewegungskompensation von MPEG-4 maximal erreichen?

Selbsttestaufgabe 7.2-4:

Nennen Sie die Strukturebenen von MPEG-4 Video!

Selbsttestaufgabe 7.2-5:

Nennen Sie die Visual Profiles von MPEG-4 Version 1!

Selbsttestaufgabe 7.2-6:

Welches Visual Profile von MPEG-4 wird gerne bei der DivX-Videocodierung verwendet? Kann dieses Profile beliebige Objektkonturen verarbeiten?

Selbsttestaufgabe 7.2-7:

Was versteht man bei MPEG-4 unter SNHC? Nennen Sie ein Beispiel, bei dem SNHC eingesetzt wird!

7.3 Advanced Video Coding - AVC

Joint Video Team – JVT

Hinter dem Kürzel *H.264/AVC* steht der derzeit jüngste Videocodierstandard, der vom *Joint Video Team* (*JVT*), einer gemeinsamen Expertengruppe der ITU-T (*Video Coding Experts Group – VCEG*) und ISO/IEC (*Moving Picture Experts Group – MPEG*) entwickelt wurde. Er ist auf hohe Codiereffizienz getrimmt, verfügt über eine verbesserte Anpassung an Netzwerkstrukturen und besitzt gegenüber H.263 mit seinen vielen Ergänzungen (*Annex D* bis *X*) über eine vereinfachte Syntax.

7.3.1 Überblick

MPEG-4 Part 10/AVC H.264/AVC

Im Zuge der zahlreichen Ergänzungen zu ITU-T H.263 stieß im Jahr 1997 die *Video Coding Experts Group* (*VCEG*) das Projekt *H.26L* mit dem Ziel an, einen neuen Codierstandard zu entwickeln, mit dem gegenüber den bisherigen Verfahren eine deutliche Steigerung der Codiereffizienz möglich sein sollte. Ende 2001 schlossen sich die beiden Expertengremien für die Videocodierung der ISO/IEC und der ITU-T zu einer neuen Gruppe zusammen, dem *Joint Video Team* (*JVT*). Diese Gruppe hatte fortan die Aufgabe, den neuen Videostandard unter dem Titel *H.264/AVC* bei der ITU-T bzw. in identischer Form bei der ISO/IEC als Teil 10 des existierenden MPEG-4 Standards (*MPEG-4 Part 10/AVC*) fort zu entwickeln und zu verabschieden. Die Standardisierung von *H.264/AVC* fand 2003 in der ersten Fassung statt [7.13]. Drei zentrale Ziele verfolgt der Standard:

Verbesserte Codiereffizienz

Ziele des Standards

Die mittlere erforderliche Datenrate soll gegenüber den bisherigen Videocodieralgorithmen (z. B. ITU-T H.263++) bei gleicher Bildqualität halbiert werden können. Dabei soll die zusätzlich notwendige Komplexität des Decoders in einer akzeptablen Relation zur Steigerung der Codiereffizienz stehen.

Verbesserte Netzwerkanpassung

Die Datenstruktur von H.264/AVC wurde von vornherein so geplant, dass eine Übertragung über verschiedene Netzwerkstrukturen möglich ist. Hauptsächlich ist bei H.264/AVC an paketvermittelte Übertragungen im Internet und an Mobilfunkanwendungen mit *UMTS*-Technologie gedacht. Verschiedene Techniken sind bei der Videocodierung von H.264/AVC integriert worden, um die Qualitätseinbußen bei Paketverlusten zu minimieren.

Einfache Videocodiersyntax

H.264/AVC ist mit dem Anspruch gestartet, eine einfache und übersichtliche Videocodiersyntax zu besitzen. Umfangreiche Syntaxoptionen wie sie beim ITU-T H.263- oder MPEG-4-Standard existieren, sollten nach Möglichkeit vermieden werden. Allerdings galt es hier, einen Kompromiss zu finden, da sich ein Codierstandard oft nur dann gut für ein spezielles Anwendungsgebiet eignet, wenn eigens zugeschnittene Codieroptionen die jeweiligen Anforderungen besonders berücksichtigen.

7.3.2 Eigenschaften

Generischer Codierstandard

Ähnlich wie bei der Standardisierung der bisherigen Videocodierverfahren bei ISO/IEC oder ITU-T wird auch bei H.264/AVC nur der Datenstrom, die Syntax und der Decoder selbst im Standard festgelegt. Mit diesem Ansatz sind für die Implementierung des Encoders unterschiedliche Varianten realisierbar; die Komplexität des Coders und des Decoders selbst sind frei wählbar. Zudem bleibt der Standard für künftige Optimierungen offen.

H.264/AVC ist nicht eine komplette Neuentwicklung. Vielmehr wurden erprobte Elemente von anderen Codierstandards (z. B. ITU-T H.263 oder MPEG-4 Video) neu kombiniert, verbessert oder erweitert und an einigen Stellen durch neue Ansätze ergänzt. Bewährt und deshalb übernommen wurden unter anderem folgende Elemente:

Elemente aus bisherigen Standards

- Hybride DCT mit Makroblöcken (16 x 16 Pixel Luminanz, 2 Blöcke für die Chrominanz mit jeweils 8 x 8 Pixeln),
- Farbunterabtastung 4:2:0,
- Bewegungskompensation auf der Basis von Blockverschiebungen,
- Verschiebungsvektoren dürfen über die Bildgrenzen hinauszeigen,
- Intraframe-Codierung (I-Blöcke), einseitig und zweiseitig prädiktive Interframe-Codierung (P- und B-Blöcke) sind möglich,
- die Intraframe-Codierung wird durch eine örtliche Prädiktion ergänzt.

Neue Elemente

Folgende neue bzw. verbesserte Elemente ergänzen den Standard:

- Verbesserungen bei der Bewegungskompensation,
- die Transformationscodierung wird vorzugsweise mit kleinen Blöcken (4 x 4 Pixel) durchgeführt,
- das *Deblocking Filter* als Schleifenfilter arbeitet adaptiv an den Blockgrenzen,
- die Entropiecodierung, bisher bestehend aus VLC und RLC wird durch eine kontext-adaptive arithmetische Codierung (*Context-adapted Binary Arithmetic Coding – CABAC*) und eine kontext-adaptive variable Lauflängencodierung (*Context-adaptive Variable-length Coding – CAVLC*) ersetzt,

- größere Flexibilität bei der Zusammenfassung von Makroblöcken (*Flexible Slice Size*) und bei der Anordnung von Makroblöcken (*Flexible Macroblock Ordering – FMO*) und Slices (*Arbitary Slice Ordering – ASO*) im Datenstrom.

H.264/AVC Codierkonzept

Das Blockschaltbild des Codierprozesses ist in seiner Grundversion in Abb. 7.3-1 dargestellt. Erkennbar ist der Grundaufbau der hybriden DCT. Die Ziffern geben die Positionen im Blockschaltbild an, bei denen sich Veränderungen gegenüber früheren Codiersystemen ergeben.

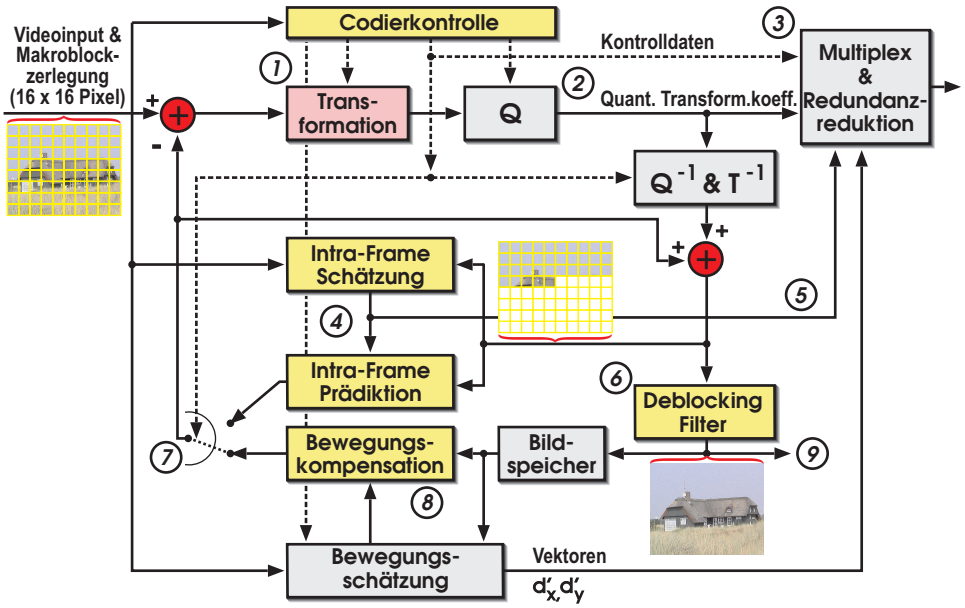


Abb. 7.3-1: Blockschaltbild des *H.264/AVC* Codierers in der Basisversion

Als Transformation (1) wird nicht mehr die DCT verwendet, sondern die so genannte *Integer-Transformation*, die im Abschnitt 7.3.6 eingehend vorgestellt wird. Für die Quantisierung (2) existieren 52 verschiedene Quantisierungsparameter, die eine proportionale Änderung der Quantisierungsstufen bewirken (vgl. Abschnitt 7.3.7). Für die Chrominanz werden dabei durchweg kleinere Stufenhöhe verwendet, als bei der Luminanz. Ein ähnlicher Ansatz wurde bereits als Codieroption *Annex T* zum ITU-T H.263+ Standard definiert. Bei der Redundanzreduktion (3) werden die Transformationskoeffizienten mit einer *inhaltsabhängigen VLC* zusammengefasst. Überdies erfolgt die Codierung mit einer ebenfalls *kontextsensitiven binären arithmetischen Codierung*. Eine Erhöhung der Codiereffizienz für Intraframe-codierte Makroblöcke wird durch eine *örtliche Prädiktion* (4) möglich, die in Abschnitt 7.3.5 näher beleuchtet wird. Die entsprechenden örtliche Prädiktionsdaten werden ebenfalls in den Datenstrom integriert (vgl. (5)). Das Schleifenfilter (6) (*Deblocking Filter*) arbeitet ähnlich wie in ITU-T H.263+, *Annex J* beschrieben. Mit dem Schalter (7) wird zwischen einer Intraframe- und Interframe-Codierung umgeschaltet. Die Bewegungskompensation (8) findet wie bisher auf Makroblockebene statt, und hat eine Genauigkeit von 1/4 Pixel. Dabei können, ähnlich wie bei ITU-T H.263++, *Annex U*, mehrere Blöcke aus verschiedenen Referenzbildern für die lokale Prädiktion herangezogen werden. Zudem können die

16 x 16 Bildpunkte großen Makroblocke und die 8 x 8 Bildpunkte großen Blöcke auch in ihrer Größe an die lokalen Gegebenheiten im Bild angepasst werden (*Partitionierung*, *Segmentierung*). Dieser Sachverhalt wird in Abschnitt 7.3.4 dargestellt. Am Punkt (9) liegt das Bild an, dass so auch der Decoder rekonstruieren kann; lediglich mögliche Fehler des Übertragungskanals selbst sind nicht berücksichtigt.

7.3.3 Profiles und Levels

Es sind 3 Profiles definiert: *Baseline Profile*, *Main Profile* und *Extended Profile*. Wie bei der Definition in anderen Standards, sind auch in den einzelnen Profiles jeweils Untermengen von Codieroptionen[71] des Standards möglich.

Baseline Profile

Profiles

Das *Baseline Profile* ist für den Einsatz bei Videokonferenzsystemen und einfachen Internet Streaming Anwendungen ausgelegt. Es unterstützt alle im Standard definierten Codieroptionen mit der Ausnahme von:

- B-Bildtypen,
- den so genannten *Switching BildtypenSI* und *SP*[72],
- gewichteten Prädiktionen zwischen mehreren Bildern,
- Verarbeitung von Zeilensprungmaterial,
- Redundanzreduktion nach dem *CABAC*-Verfahren.

Main Profile

Das *Main Profile* eignet sich für Broadcast- und DVD-ähnliche Anwendungen, bei denen TV-Qualität erzielt werden soll. Alle Codieroptionen bis auf die folgenden zwei werden unterstützt:

- flexible Makroblock Anordnung,
- *Switching BildtypenSI* und *SP.*

Extended Profile

Das *Extended Profile* ist am besten für das drahtgebundene oder mobile Internet-Streaming ausgelegt. Es unterstützt alle Codieroptionen, lediglich die Redundanzreduktion kann nicht mit dem *CABAC*-Verfahren durchgeführt werden.

Komplexität und Codiereffizienz

Die Unterstützung der Profiles stellt an die Komplexität des Decoder mehr oder weniger hohe Anforderungen.

---

71 Codieroptionen sind vergleichbar mit der Definition von Annexen bei H.263(+) (s. o.)

72 Die beiden Slice-Typen *SP* (*Switching P*) und *SI* (*Switching I*) ermöglichen ein effizientes Umschalten zwischen Bitströmen, die bei verschiedenen Bitraten codiert wurden. Zur Erhöhung des Fehlerschutzes können *SP*- und *SI*-Bildtypen eingesetzt werden.

Tab. 7.3-1: *Komplexität und Steigerung der Codiereffizienz des H.264 / AVC Codierkonzepts gegenüber MPEG-2*

H.264 / AVC Profile	Komplexität gegenüber MPEG-2	Steigerung der Codiereffizienz gegenüber MPEG-2
Baseline Profile	2.5-fach mehr komplex	1.7-fach besser
Main Profile	4.0-fach mehr komplex	2.0-fach besser
Extended Profile	3.5-fach mehr komplex	2.7-fach besser

In [7.14] wurde eine grobe Abschätzung für die verschiedenen Profiles im Verhältnis zu vergleichbaren Profiles beim MPEG-2 Standard durchgeführt. Überdies wurde ermittelt, mit welcher Steigerung der Codiereffizienz zu rechnen ist. Tabelle 7.3-1 führt die Ergebnisse der Untersuchungen tabellarisch auf.

Levels

Levels

Für die verschiedenen Profiles gibt es zum einen die Möglichkeit, jeweils identische Levels zu verwenden. Darüber hinaus existieren aber auch speziell definierte Implementierungen für die verschiedenen Profiles, bei denen auch unterschiedliche Levels zur Anwendung kommen können. In 11 Levelstufen variieren die Bildauflösungen von QCIF bis hin zu HD-Formaten. Mit den Levelstufen werden dabei nicht nur die Bereiche der zulässigen Bildauflösung sondern auch eine Reihe interner Codierparameter festgelegt. Einzelheiten finden sich in [7.13].

7.3.4 Blockbildung für die Bewegungskompensation

Bildfeldzerlegung in 4 x 4 Pixel große Blöcke

Bei H.264/AVC sind die kleinsten Bildblöcke, die transformiert werden, nur 4 x 4 Pixel groß (vgl. Abschnitt 7.3.5), im Gegensatz zu den 8 x 8 Pixel großen Blöcken, die bei den bisherigen Codierverfahren mit einer hybriden DCT immer verwendet wurden. Die kleinere Blockgröße führt dazu, dass lokale Bildstrukturen und Objektränder besser angenähert werden können.

Partitionierung

Ein Codiergewinn lässt sich mit diesem Verfahren aber nur erreichen, wenn nicht das ganze Bild pauschal in viele kleine 4 x 4 Pixel große Blöcke eingeteilt wird. Deshalb wird ein adaptives Verfahren für die Bildfeldzerlegung in Makroblöcke und Blöcke gewählt. Diese *Partitionierung* ist vergleichbar mit der Quadtree-Partitionierung, die in Abschnitt 3.5.3 (vgl. Abb. 3.5-4) behandelt wurde. Es ist hierbei jedoch zu beachten, dass die Partitionierung bei H.264/AVC vorzugsweise dazu dient, die örtliche Korrelation zwischen den Blöcken und den tatsächlichen Grenzen der bewegten Objekte zu verbessern, damit der Prädiktionsfehler nach der Bewegungskompensation noch weiter minimiert werden kann (Interframe-Prädiktion).

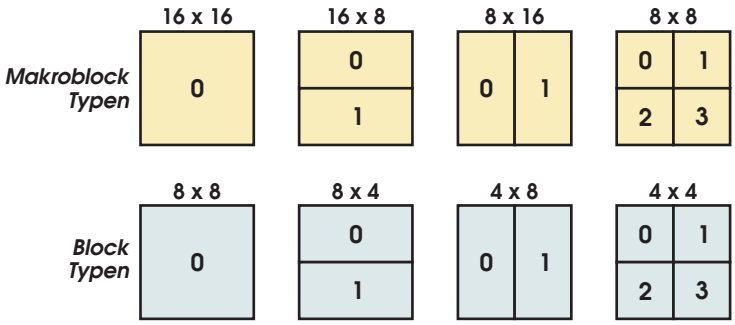


Abb. 7.3-2: Blocktypen und Segmentierungen für die Bewegungskompensation: Makroblöcke (oben) und Blöcke (unten)

Für die weiterhin 16 x 16 Pixel großen Makroblöcke und die 8 x 8 Pixel großen Blöcke stehen jeweils vier verschiedene Möglichkeiten der Segmentierung zur Verfügung. Abb. 7.3-2 visualisiert die verschiedenen Möglichkeiten. Wird eine maximal feine Segmentierung gewählt und benötigt jeder 4 x 4 Pixel große Transformationsblock einen eigenen Verschiebungsvektor, dann müssen insgesamt 16 verschiedene Verschiebungsvektoren für einen Makroblock berücksichtigt werden. Glücklicherweise müssen die Bewegungsvektoren nicht immer jeweils einzeln übertragen werden, sondern können, wenn dies für die Codiereffizienz eine Verbesserung bringt, auch aus benachbarten Makroblöcken gewichtet prädiziert werden (*Interframe-Prädiktion*).

7.3.5 Örtliche Intra-Prädiktion

Die örtliche *Intra-Prädiktion* erhöht vor allem die Codiereffizienz für Intraframe-codierte Makroblöcke und Blöcke[73]. Bei H.264/AVC unterscheidet man 9 Intra-Prädiktionsmodi für 4 x 4 Pixel große Intra-Blöcke und 4 für die 16 x 16 Pixel große Intraframe-codierte Makroblöcke der Luminanz. Für die Chrominanz steht ein Prädiktionsmodus zur Verfügung, der dem 16 x 16 Pixel-Modus der Luminanz ähnlich ist. Einige Beispiele sollen nachfolgend das Prinzip der örtlichen Prädiktion verdeutlichen.

16 x 16 Intra-Prädiktion

16 x 16 Intra-Prädiktion

Für die örtliche Prädiktion wird auf bereits decodierte benachbarte Makroblöcke zurückgegriffen. Da das Bild in *Makroblock-Zeilen* von oben nach unten abgetastet wird, kann für die Pradiktion des aktuellen Makroblockes $M(n,m)$ bestenfalls auf den links anschließenden Makroblock $M(n,m-1)$ und die Makroblöcke $M(n-1,m-1)$, $M(n,m-1)$ und $M(n+1,m-1)$, die sich in der vorherigen Makroblock-Zeile befinden, zurückgegriffen werden (vgl. Abb. 7.3-3).

73 In MPEG-4 Video wurden ähnliche Prädiktionsarten eingeführt.

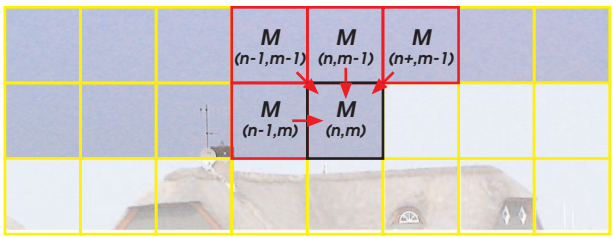


Abb. 7.3-3: Mögliche Intra-Prädiktionen für Makroblöcke

4 Prädiktionsmodi

In der Praxis werden für die örtliche Prädiktion von Makroblöcken nur *M*(*n-1*,*m*) und *M*(*n*,*m-1*) herangezogen. Makroblöcke außerhalb der eigenen *Slice* oder außerhalb des Bildes können nicht mit in die Prädiktion einbezogen werden. Bei der 16 x 16 Intra-Prädiktion gibt es die Möglichkeit der *rein vertikalen Prädiktion* (*Intra 16 x 16, Mode 0*), der *rein horizontalen Prädiktion* (*Intra 16 x 16, Mode 1*), der *Prädiktion des Gleichwertanteils* horizontal und vertikal (*Intra 16 x 16, Mode 2*) und der *diagonale Prädiktion* (*Intra 16 x 16, Mode 3 – Plane*).

Mode 0

Für den ***Mode 0*** (*vertical*) wird vertikal für alle Spalten des aktuellen Makroblocks *M*(*n*,*m*) der Wert aus der entsprechenden Spalte in der letzten Zeile des darüber liegenden Makroblocks *M*(*n*,*m-1*) übernommen:

$$\text{Präd}_{Lum.,M0}[x,y] = p[x,-1, \text{ mit } x,y = 0...15, \qquad 7.3\text{-}1$$

Mode 1

wobei p[x,y] dem Luminanzwert an der Stelle (*x*,*y*) entspricht. Entsprechend wird horizontal die Prädiktion im ***Mode 1*** (*horizontal*) vorgenommen:

$$\text{Präd}_{Lum.,M1}[x,y] = p[-1,y], \text{ mit } x,y = 0...15. \qquad 7.3\text{-}2$$

Mode 2

Bei ***Mode 2*** (*DC*) ist die Berechnung davon abhängig, ob der jeweils vertikal bzw. horizontal benachbarte Makroblock für die Prädiktion zur Verfügung steht. Man unterscheidet drei Fälle. Sind beide erforderlichen Makroblöcke nutzbar, errechnet sich die Prädiktion zu[74]:

$$\text{Präd}_{Lum.,M2}[x,y] = \frac{16 + \sum_{x'=0}^{15} p[x',-1] + \sum_{y'=0}^{15} p[-1,y']}{32}, \; x,y = 0...15. \qquad 7.3\text{-}3$$

Steht nur der linke Makroblock zu Verfügung vereinfacht sich die Gleichung zu:

$$\text{Präd}_{Lum,.M2}[x,y] = \frac{8 + \sum_{y'=0}^{15} p[-1,y']}{16}, \text{ mit } x,y = 0...15. \qquad 7.3\text{-}4$$

74 Die Divisionsfaktoren werden mit Bit-Schieberegistern realisiert. Hier und in allen weiteren Gleichungen ist demnach immer die Integerdivision zu verwenden!

Kann nur auf den darüber liegenden Makroblock zugegriffen werden ergibt sich:

$$\text{Präd}_{Lum.,M2}[x,y] = \frac{8 + \sum_{x'=0}^{15} p[x',-1]}{16}, \text{ mit } x,y = 0...15. \qquad 7.3\text{-}5$$

Mode 3

Der ***Mode 3*** (*Plane*) wird nur verwendet, wenn beide erforderlichen Makroblöcke vorhanden sind. Dann ergibt sich die Prädiktion zu:

$$\text{Präd}_{Lum.,M3}[x,y] = \frac{a + b(x-7) + c(y-7) - 16}{32}, \qquad 7.3\text{-}6$$

$$a = 16 \cdot (p[-1,15] + p[15,-1]), \qquad 7.3\text{-}6$$

$$b = \frac{5H+32}{64},\ c = \frac{5V+32}{64}, \qquad 7.3\text{-}6$$

$$H = \sum_{x'=0}^{7} (x'+1) \cdot (p[8+x',-1] - p[6-x',-1]), \qquad 7.3\text{-}6$$

$$V = \sum_{y'=0}^{7} (y'+1) \cdot (p[-1,8+y'] - p[-1,6-y']), \qquad 7.3\text{-}6$$

Clipping

Ergibt sich ein Prädiktionswert, der außerhalb des Wertebereichs [0,255] liegt, wird er auf einen Wert von 0 bzw. 255 geklippt.

7.3.5.1 4 x 4 Intra-Prädiktion

4 x 4 Intra-Prädiktion

Bei den insgesamt 9 Prädiktionsmodi für die 4 x 4 Intra-Prädiktion wird jeweils auf bis zu 4 benachbarte 4 x 4 Blöcke zurückgegriffen. Abb. 7.3-4 zeigt die Bildpunkte des aktuellen Blockes (Kleinbuchstaben) und die für die Prädiktion herangezogenen Bildpunkte der benachbarten Blöcke (Großbuchstaben).

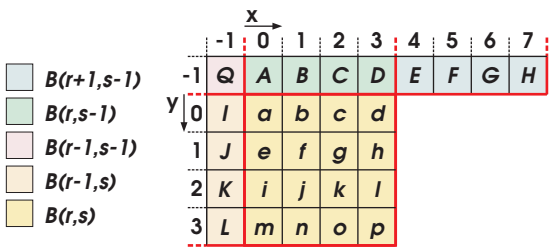


Abb. 7.3-4: Mögliche Bildpunkte für eine Intra-Prädiktion eines 4 x 4 Blocks

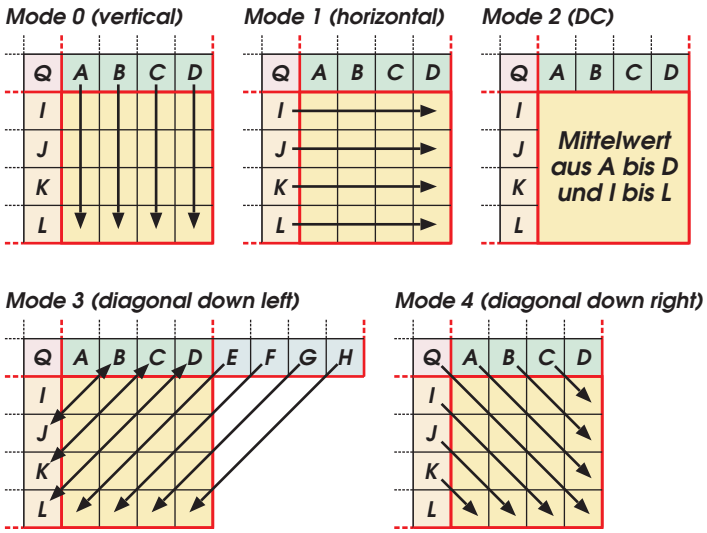


Abb. 7.3-5: Prädiktionsrichtungen 4 x 4 Intra Blöcke (*Mode 0* bis *4*)

Ist der Block *B*(*r+1*,*s-1*) nicht verfügbar werden die Pixelwerte *E*, *F*, *G* und *H* in einigen Fällen durch den Wert *D* aus Block *B*(*r*,*s-1*) ersetzt. Die Prädiktionsrichtungen der einzelnen Modi sind in Abb. 7.3-5 und Abb. 7.3-6 dargestellt.

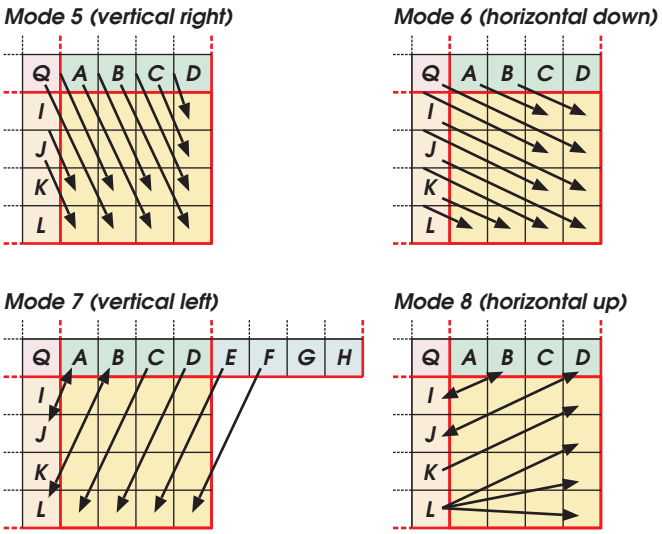


Abb. 7.3-6: Prädiktionsrichtungen 4 x 4 Intra Blöcke (*Mode 5* bis *8*)

Um eine Prädiktion im jeweiligen Mode durchführen zu können, müssen wieder die entsprechenden Pixel der benachbarten Blöcke verfügbar sein und im selben Slice liegen. Einige Modi können auch noch angewendet werden, wenn einer von mehreren Blöcken nicht mit in die Prädiktion einbezogen werden kann. Der DC-Prädiktionsmodus (*Mode 2*) kommt, wenn erforderlich, sogar ganz ohne Beziehungen zu benachbarten Blöcken aus. Er „prädiziert“ dann für den ganzen Block ein mittleres grau (Wert: 128, s. u.).

Mode 0 (*vertical*) ermittelt sich formal: **Mode 0**

$$\text{Präd}_{4x4Lum.,M0}[x,y] = p[x,-1], \text{ mit } x,y = 0...3, \qquad 7.3\text{-}7$$

Mode 1 ***Mode 1*** (*horizontal*) errechnet sich zu:

$$\text{Präd}_{4x4Lum.,M1}[x,y] = p[-1,y], \text{ mit } x,y = 0...3. \qquad \text{7.3-8}$$

Mode 2 Beim ***Mode 2*** (*DC*) werden vier Fälle in Abhängigkeit davon unterschieden, ob auf die Bildpunkte *I* bis *L* und *A* bis *D* zugegriffen werden kann. Es ergibt sich dann:

$$\text{Präd}_{4x4Lum.,M2}[x,y] = \begin{cases} \frac{1}{8}(\sum_{x'=0}^{3} p[x',-1] + \sum_{y'=0}^{3} p[-1,y'] + 4), & A...D \text{ und } I...L \\ \frac{1}{4}(\sum_{y'=0}^{3} p[-1,y'] + 2, & \text{nur } I...L \\ \frac{1}{4}(\sum_{x'=0}^{3} p[x',-1] + 2, & \text{nur } A...D \\ 128, & \text{sonst} \end{cases} . \qquad \text{7.3-9}$$

Mode 3 Der ***Mode 3*** (*down left*) wird nur angewendet, wenn die Bildpunkte *A* bis *H* vorhanden sind. Die Prädiktion errechnet sich dann zu:

$$\text{Präd}_{4x4Lum.,M3}[x,y] =$$

$$\begin{cases} \frac{1}{4}(G+3H+2), & x=y=3 \\ \frac{1}{4}(p[x+y,-1+2p[x+y+1,-1]+p[x+y+2,-1]+2), & sonst \end{cases} . \qquad \text{7.3-10}$$

Mode 4 ***Mode 4*** (*down right*) ist möglich, wenn auf die Bildpunkte *A* bis *D*, *I* bis *L* und *Q* zugegriffen werden kann:

$$\text{Präd}_{4x4Lum.,M4}[x,y] =$$

$$\begin{cases} \frac{1}{4}(p[+-y-2-1]+2p[x-y-1,-1]+p[x-y,-1]+2), & x>y \\ \frac{1}{4}(A+2Q+I+2), & x=y \\ \frac{1}{4}(p[-1,y-x-2]+2p[-1,y-x-1]+p[-1,y-x]+2), & x<y \end{cases} . \qquad \text{7.3-11}$$

Mode 5 Für den ***Mode 5*** (*vertical right*) sind die Bildpunkte *A* bis *D*, *Q* und *I* bis *L* erforderlich:

$$\text{Präd}_{4x4Lum.,M5}[x,y] =$$

$$\begin{cases} \frac{1}{2}(p[x-\frac{y}{2}-1,-1]+p[x-\frac{y}{2},-1]+1), & VR=0,2,4,6 \\ \frac{1}{4}(p[x-\frac{y}{2}-2,-1]+2p[x-\frac{y}{2},-1-1]), & VR=1,3,5 \\ \frac{1}{4}(I+2Q+A+2), & VR=-1 \\ \frac{1}{4}(p[-1,y-1]+2p[-1,y-2]+p[-1,y-3]+2), & VR=-2,-3 \end{cases} .$$

7.3-12

mit

$$VR = 2x - y. \tag{7.3-13}$$

Wie bei *Mode 5* sind auch beim ***Mode 6***(*horizontal down*) die Bildpunkte *A* bis *D*, *Q*, und *I* bis *L* erforderlich: **Mode 6**

$$\text{Präd}_{4x4Lum.,M6}[x,y] =$$

$$\begin{cases} \frac{1}{2}(p[-1,y-\frac{x}{2}-1]+p[-1,y-\frac{x}{2}]+1), & HD=0,2,4,6 \\ \frac{1}{4}(p[-1,y-\frac{x}{2}-2]+2p[-1,y-\frac{x}{2}-1]+p[-1,y-\frac{x}{2}]+2), & HD=1,3,5 \\ \frac{1}{4}(I+2Q+A+2), & HD=-1 \\ \frac{1}{4}(p[x-1,-1]+2p[x-2,-1]+p[x-3,-1]+2), & HD=-2,-3 \end{cases} \tag{7.3-14}$$

$$HD = 2y - x. \tag{7.3-15}$$

Bei ***Mode 7*** (*vertical left*) werden die Bildpunkte *A* bis *H* und *I* bis *L* benötigt: **Mode 7**

$$\text{Präd}_{4x4Lum.,M7}[x,y] =$$

$$\begin{cases} \frac{1}{2}(p[x+\frac{y}{2},-1+p[x+\frac{y}{2}+1,-1]+1), & y=0,2 \\ \frac{1}{4}(p[x+\frac{y}{2},-1]+2p[x+\frac{y}{2}+1,-1]+p[x+\frac{y}{2}+2,-1]+2)), & sonst \end{cases} \tag{7.3-16}$$

Für den ***Mode 8*** (*horizontal up*) werden die Bildpunkte *I* bis *L* gebraucht: **Mode 8**

$$\text{Präd}_{4x4Lum.,M8}[x,y] =$$

$$\begin{cases} \frac{1}{2}(p[-1,y+\frac{x}{2}]+p[-1,y+\frac{x}{2}+1]+1), & HU=0,2,4 \\ \frac{1}{4}(p[-1,y+\frac{x}{2}]+2p[-1,y+\frac{x}{2}+1]+p[-1,y+\frac{x}{2}+2]+2), & HU=1,3 \\ \frac{1}{4}(K+3L+2), & HU=5 \\ L, & HU>5 \end{cases} \tag{7.3-17}$$

mit

$$HU = x + 2y. \tag{7.3-18}$$

Intra-Prädiktion für Chrominanzbildpunkte

Intra-Prädiktion für Chrominanz

Bei der Chrominanz werden ähnlich wie bei 16 x 16 Intra-Prädiktion (s. o.) 4 verschiedene Modi unterschieden:

- *Mode 0 – Chroma DC,*
- *Mode 1 – Chroma Horizontal,*
- *Mode 2 – Chroma Vertikal,*
- *Mode 3 – Chroma Diagonal* (*Plane*).

Auf die entsprechenden Berechnungsformeln soll hier nicht weiter eingegangen werden. Einzelheiten finden sich in [7.13].

7.3.6 Integer-Transformation

Integer-Transformation auf 4 x 4 Blöcke

In Abb. 7.3-2 wurden verschiedene Segmentierungen der Makroblöcke und Blöcke dargestellt. Da die feinste Zerlegung eines 16 x 16 Bildpunkte großen Makroblockes unter Umständen in bis zu 16 Blöcke mit je 4 x 4 Bildpunkten geschehen kann, wird bei ITU-T H.264 keine klassische DCT mehr angewendet, sondern die so genannte *Integer-Transformation*. Diese ist der DCT recht ähnlich, macht aber einige Vereinfachungen, so dass sämtliche Koeffizienten als Integerwerte errechnet werden können, was die algorithmische Realisierung entscheidend beschleunigt.

Herleitung aus DCT

Die Integer-Transformation wird aus der DCT (vgl. Gl.3.1-4) hergeleitet. Gl.3.1-4 lässt sich für

$$M = N = 4 \tag{7.3-19}$$

in eine Matrixschreibweise umformen:

$$\underline{Y}_{DCT} = (\underline{A} \cdot \underline{X})\,\underline{A}^T = \left[\begin{pmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{pmatrix} \cdot \underline{X} \right] \cdot \begin{pmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{pmatrix}, \tag{7.3-20}$$

mit

$$\underline{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix}, \tag{7.3-21}$$

$$a = \frac{1}{2},\ b = \sqrt{\frac{1}{2}} \cdot \cos\left(\frac{\pi}{8}\right),\ c = \sqrt{\frac{1}{2}} \cdot \cos\left(\frac{3\pi}{8}\right).$$

$\underline{X}$ repräsentiert den zu transformierenden 4 x 4 Bildblock. Erkennbar erzeugen die Konstanten *b* und *c* immer rationale Anteile in den transformierten Koeffizienten. Dieses gilt es zu verhindern. Dazu soll die Matrixmultiplikation in Gl. 7.3-20 in eine faktorisierte Form überführt werden:

$$\underline{Y}_{DCT} = \underline{W} \otimes \underline{C} = \left(\left(\underline{B} \cdot \underline{X}\right) \underline{B}^T\right) \otimes \underline{C}. \qquad 7.3\text{-}22$$

Hierin ist

$$\underline{B} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{pmatrix}, \; \underline{C} = \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix}, \; d = \frac{c}{b}. \qquad 7.3\text{-}23$$

core-Transformation

Die Matrix $\underline{W}$ wird als *core*-Transformation bezeichnet. Die Matrix $\underline{C}$ enthält die Skalierfaktoren aus den bekannten Konstanten *a* und *b*. Das Symbol $\otimes$ gibt an, dass jedes Element der Ergebnismatrix $\left(\left(\underline{B} \cdot \underline{X}\right) \underline{B}^T\right)$ mit dem jeweiligen Element (*Skalierfaktor*) multipliziert wird, der an der gleichen Position in der Matrix $\underline{C}$ steht[75]. Als erste Vereinfachung wird der Wert für *d* angenähert zu

$$d = \frac{1}{2} \qquad 7.3\text{-}24$$

Damit die neue Matrix $\underline{B}'$ weiterhin orthogonal bleibt, muss auch die Konstante *b* verändert werden:

$$b = \sqrt{\frac{2}{5}}. \qquad 7.3\text{-}25$$

Weiterhin wird die zweite und vierte Zeile der Matrix $\underline{B}'$ mit zwei multipliziert. Zum Ausgleich wird die *Skalierungsmatrix C* an den entsprechenden Stellen durch zwei bzw. vier[76] dividiert. Damit sind sämtliche rationale Zahlenanteile aus der Matrix $\underline{B}$ entfernt und eine Matrixmultiplikation im Integer-Zahlenraum ist für die modifizierte *core*-Transformation $\underline{W}'$

$$\underline{W}' = \left(\underline{B}' \cdot \underline{X}\right) \underline{B}'^T \qquad 7.3\text{-}26$$

möglich. Insgesamt ergibt sich damit für die Integer-Transformation folgendes Ergebnis:

$$\underline{Y}_{Int} = \underline{W}' \otimes \underline{C}' = \left(\left(\underline{B}' \cdot \underline{X}\right) \underline{B}'^T\right) \otimes \underline{C}', \qquad 7.3\text{-}27$$

75 Es wird also eine *skalare Multiplikation* mit der Matrix $\underline{C}$ durchgeführt.

76 Der Faktor vier ergibt sich, da die transponierte Matrix $\underline{B}^T$ in den Spalten zwei und vier mit dem Faktor zwei multipliziert wurde.

mit

$$\underline{B}' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \; \underline{C}' = \begin{pmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{pmatrix}. \qquad 7.3\text{-}28$$

Diese Integer-Transformation stellt eine Näherung der klassischen 4 x 4 DCT dar. In der Praxis zeigt sich, dass einige Koeffizienten ähnlich sind, andere sich auch stärker unterscheiden. Der DC-Koeffizient ist in beiden Fällen immer exakt gleich.

Rücktransformation

Die *Rücktransformation* errechnet sich wie folgt:

$$\underline{X}' = \underline{B}_i^T \left((\underline{Y}_{Int} \otimes \underline{C}_i) \, \underline{B}_i \right) \qquad 7.3\text{-}29$$

mit den inversen Matrizen $\underline{B}'_i$ und $\underline{C}'_i$

$$\underline{B}_i = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{pmatrix}, \; \underline{C}'_i = \underline{C} = \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix}. \qquad 7.3\text{-}30$$

Bei der Rücktransformation wird die Koeffizientenmatrix Y_{Int} zunächst zurück skaliert. Man beachte überdies, dass die Matrix $\underline{B}'_i$ nun auch Werte von + 1/2 und -1/2 enthält. Diese Werte stellen jedoch kein Problem dar, da sie mit einer einfachen *Right-Shift-Operation* ohne signifikante Qualitätsverluste errechnet werden können.

Weitere Transformation für DC-Koeffizienten

Auf eine Besonderheit soll bei der Verarbeitung der DC-Koeffizienten hingewiesen werden. Wird eine Segmentierung gewählt, bei der in einem Makroblock 16 der 4 x 4 Blöcke vorhanden sind, dann werden die 16 DC-Koeffizienten der 16 Blöcke in einem eigenen virtuellen 4 x 4 Block zwischengespeichert (vgl. Abb. 7.3-7).

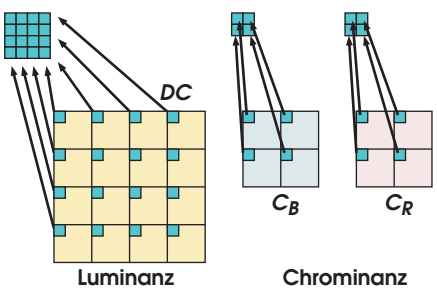


Abb. 7.3-7: Zusammenfassung der DC-Koeffizienten in neuen, virtuellen Blöcken für die Anwendung eine weiteren Transformation

Hadamard Transformation

Für diesen DC-Block wird dann eine weitere Transformation durchgeführt, allerdings nicht die oben angeführte Integer-Transformation, sondern die 4 x 4 *Hadamard Transformation* (vgl. Abb. 3.1-1). Diese ist leichter zu realisieren, da ihre Transformationsmatrix $\underline{B}_{Had}$ nur die Werte -1 und +1 enthält.

Es seien $\underline{W}_{DC}$ die Matrix der 4 x 4 Bildpunkte große Block mit den DC Koeffizienten, dann errechnet sich die Transformation des Blockes $\underline{Y}_{DC}$ mit Hilfe der Hadamard Transformation wie folgt:

$$\underline{Y}_{DC,Lum} = \frac{1}{2}\left(\left(\underline{B}_{Had} \cdot \underline{W}_{DC}\right)\underline{B_{Had}}^{T}\right), \qquad 7.3\text{-}31$$

mit

$$\underline{B}_{Had} = \underline{B_{Had}}^{T} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}. \qquad 7.3\text{-}32$$

Transformation DC-Koeffizienten Chrominanz

Für die DC-Koeffizienten der Chrominanz wird der gleiche Vorgang durchgeführt, nur dass hier mit 2 x 2 Bildpunkten großen Matrizen gearbeitet wird. Die entsprechende Transformation errechnet sich wie folgt:

$$\underline{Y}_{DC,Chr} = \left(\underline{B'}_{Had} \cdot \underline{W}_{DC}\right)\underline{B'_{Had}}^{T}, \qquad 7.3\text{-}33$$

mit

$$\underline{B}_{Had} = \underline{B_{Had}}^{T} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \qquad 7.3\text{-}34$$

7.3.7 Quantisierung und Auslesen der Koeffizienten

Quantisierungsparameter *QP*

Quantisierungsstufe Q_{step}

Für die Quantisierung kommen bei H.264/AVC insgesamt 52 verschiedene *QuantisierungsparameterQP* zum Einsatz. Mit diesen werden die eigentlichen *Quantisierungsstufen*Q_{step} bestimmt. Berechnet wird die Quantisierung wie bei den bisherigen Verfahren nach der Formel[77]

$$Z_{ij} = round\left(\frac{Y_{ij}}{Q_{step}}\right), \qquad 7.3\text{-}35$$

wobei Y_{ij} die ursprünglichen und Z_{ij} die quantisierten Transformationskoeffizienten darstellen. Der Zusammenhang zwischen *QP* und Q_{step} ist so ausgelegt, dass eine Erhöhung von *QP* um den Wert 6 genau zu einer Verdoppelung von Q_{step} führt, oder anders ausgedrückt: Q_{step} erhöht sich jeweils um 12.5 %, wenn *QP* um 1 größer gewählt wird. Mit diesem Verfahren lässt sich die Quantisierung über einen weiten Bereich sehr fein justieren. Tabelle 7.3-2 zeigt den logarithmischen Zusammenhang zwischen *QP* und Q_{step} für einige Zahlenwerte.

Zusammenhang *QP* und Q_{step}

77 Die Funktion *round(Argument)* rundet das Argument auf den nächstliegenden ganzahligen Wert auf bzw. ab.

Tab. 7.3-2: *Quantisierungsstufen Q_{step} in Abhängigkeit des Quantisierungsparameter QP*

QP	0	1	2	3	4	5	6
Q_{step}	0.625	0.6875	0.8125	0.875	1	1.125	1.25
QP	7	8	9	10	11	12	...
Q_{step}	1.375	1.625	1.75	2	2.25	2.5	...
QP	18	24	30	36	42	48	51
Q_{step}	5	10	20	40	80	160	224

Für die Chrominanz wird ein anderer Zusammenhang zwischen *QP* und Q_{step} gewählt. Q_{step} wird oberhalb von einem *QP*-Wert von 30 nur noch langsam stetig erhöht; die Quantisierungsstufen sind also für diesen Bereich feiner abgestimmt.

Parameter *PF* Die Integer-Transformation in Gl. 7.3-27 enthält die *Skalierungsmatrix C*, welche aus den nichtganzen Zahlen *a*2, *ab*/2 und *b*2/4 besteht. Der Aufwand für die Berechnung kann verringert werden, wenn diese Faktoren aus der Transformation selbst herausgenommen und in die Quantisierung integriert werden. Dazu führt man einen Faktor *PF* ein, der je nach Position in der Matrix einer der drei Faktoren *a*2, *b*2/4 oder *ab*/2 annimmt. Für die Quantisierung wird dann nur noch die *core*-Transformation $\underline{W'}$ nach Gl. 7.3-26 herangezogen. Die quantisierten Transformationskoeffizienten ergeben sich zu

$$Z_{ij} = round\left(W_{ij}\frac{PF}{Q_{step}}\right). \qquad 7.3\text{-}36$$

Die Werte für *PF* sind der nachfolgenden Tabelle 7.3-3 zu entnehmen.

Tab. 7.3-3: *Werte für den Skalierungsfaktor PF*

Position in Matrix $\underline{C}$'	*PF*
(0,0), (2,0), (0,2), (2,2)	a^2
(1,1), (1,3), (3,1), (3,3)	$b^2/4$
alle übrigen	*ab/2*

Parameter *MF* Um den Aufwand für die Division bei der Implementierung zu minimieren, wird oft statt der Division durch Q_{step} wieder eine Schieberegisteroperation zur Basis 2 gewählt. Dazu führt man einen weiteren Faktor *MF* ein, für den gilt[78]:

$$\frac{MF}{2^{qbits}} = \frac{PF}{Q_{step}}, mitqbits = 15 + floor\left(\frac{QP}{6}\right). \qquad 7.3\text{-}37$$

Für die Integerdarstellung des Faktors *MF* wird eine geringe Abweichung von dieser Gleichung bei der Annäherung der Faktoren $b^2/4$ und *ab*/2 in Kauf genommen.

78 Die Funktion *floor(Argument)* rundet das Argument auf den nächstliegenden ganzahligen Wert ab.

Tab. 7.3-4: *Werte für den Skalierungsfaktor MF*

QP	Positionen von $\underline{C}'$ (0,0), (2,0), (2,2), (0,2)	Positionen von $\underline{C}'$ (1,1), (1,3), (3,1), (3,3)	übrige Positionen
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

Für Werte von *QP* im Bereich zwischen 0 und 5 ist der Faktor *MF* in Tabelle 7.3-4 dargestellt. Für *QP* > 5 ändert sich der Faktor *MF* nicht, doch steigt der Divisionsfaktor 2^{qbits} wieder jeweils um den 2 bei einer Erhöhung von *QP* um 6. So ist beispielsweise *qbits* = 16 für 5 < *QP* < 12, *qbits* = 17 für 11 < *QP* < 18, usw.

Der Decoder verwendet eine inverse Quantisierung (*Rescaling*), die im Standard fest vorgeschrieben ist. Um Rundungsfehler möglichst zu vermeiden, wird der Faktor *PF* und Q_{step} mit 64 multipliziert. Zudem wird auch hier nur die inverse *core*-Transformation angewendet. Weitere Details finden sich in [7.13].

7.3.8 Weitere Elemente

Zick-Zack-Scanning

Die Zusammenfassung der quantisierten Transformationskoeffizienten wird nach dem bewährten ***Zick-Zack-Scannning*** durchgeführt. Dabei muss aber aufgrund der geänderten Blockstrukturen und der damit verbundenen Segmentierung zwischen 4 Blocktypen gemäß Abb. 7.3-2 unterschieden werden.

Deblocking Filter

Das Schleifenfilter (***Deblocking Filter***) verhindert subjektiv sehr effektiv die Bildung von Blockstrukturen bei geringen Datenraten. Dazu wird in der Rückkopplungsschleife ein lokales, nichtlineare Filter eingesetzt. Zunächst werden jeweils 3 Bildpunkte links und rechts der Blockgrenze in ihrem Luminanzverlauf analysiert. Der bekannte Quantisierungsparameter *QP* ist eine gute Hilfe, um eine Entscheidung darüber zu treffen, ob lokal im Blockgrenzenbereich eine örtliche Filterung durchgeführt werden muss oder nicht. Praxistests zeigen sehr deutlich, dass sowohl die subjektive als auch die objektive Bildqualität mit dieser Maßnahme steigt.

CAVLC und CABAC

Ein kleiner Anteil der Datenrate kann auch durch eine bessere Redundanzreduktion eingespart werden. Bei H.264/AVC wird die variable Lauflängencodierung wesentlich besser an die Statistik des Datenstroms angepasst. Das so genannte *Context-Adaptive Variable Length Coding* (***CAVLC***) und das *Context-Adaptive Binary Arithmetic Coding* (***CABAC***) wird hier verwendet. Eine Beschreibung dieser Algorithmen findet sich beispielsweise in [7.16].

7.3.9 Bewertung des Standards

Die vorherigen Kapitel haben gezeigt, dass der Standard H.264/AVC insgesamt recht komplex ist. Die Entwicklung von effektiv nutzbaren Hard- und Softwarelösungen schreitet aber unvermindert voran, so dass davon auszugehen ist, dass sich Codierverfahren nach dem H.264/AVC Standard über kurz oder lang in vielen Anwendungen wiederfinden werden.

Das Codierkonzept zeigt insgesamt, dass die zahlreichen Elemente wohlüberlegt mit dem Ziel aufeinander abgestimmt sind, die Codiereffizienz für Videomaterial entscheidend zu verbessern.

Vergleich der Codiereffizienz

Für Anwendungen mit sehr niedrigen Datenraten wurden Vergleichstests mit CIF-Testsequenzen durchgeführt. Ein solcher Vergleich mit den bisherigen Codierstandards MPEG-2, H.263 HLP und MPEG-4 ASP ist in Abb. 7.3-8 für die H.261-Testsequenz *Tempete* bei 30 Hz Vollbildfrequenz dargestellt[79].

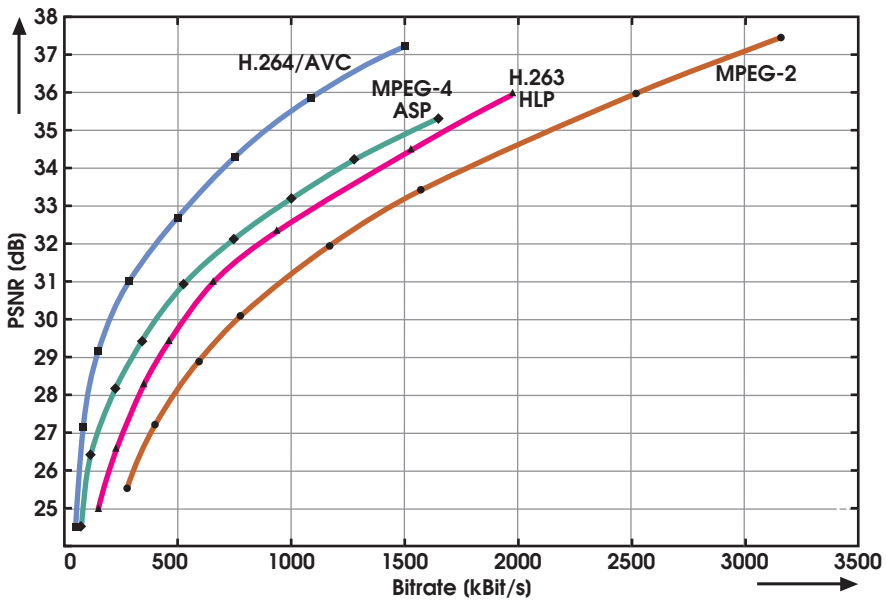


Abb. 7.3-8: Vergleich der Codiereffizienz für verschiedene Videocodierverfahren am Beispiel der Testsequenz *Tempete* (30 Hz)

Diese Ergebnisse lassen sich sicherlich nicht ohne weiteres auf hoch aufgelöstes Studiomaterial übertragen. Doch wird ersichtlich, dass eine deutliche Steigerung der Bildqualität durchaus noch erreicht werden kann. Um auch Videomaterial aus dem Studio möglichst optimal berücksichtigen zu können, wurde ein weiteres Profile im Rahmen des Standards H.264/AVC eingerichtet. Es muss sich noch erweisen, ob sich das Codierkonzept auch für dieses spezielle Anwendungsfeld bewährt.

79 Die Graphen wurden mit Messdaten aus [7.16] erstellt. Als Qualitätsmaß wurde der in Kapitel 3 in Gl.3.5-2 vorgestellte Peak-SNR (*PSNR*) für die Luminanz gewählt.

Eine sehr gute Eignung des Codierstandards H.264/AVC zeichnet sich aber schon jetzt für den Bereich der Bildkommunikation in heterogenen Netzen ab. Die Einfügung von gezielter Redundanz in den Datenstrom schon während der Quellencodierung ist eine besonders erfolgreiche Maßnahme zum Fehlerschutz.

7.4 Abschließende Bemerkungen

7.4.1 Entwicklung der digitalen Bildcodierung

Die Entwicklung der digitalen Bildcodierung hat in den letzten 5 bis 10 Jahren bemerkenswerte Fortschritte gemacht. Visionen der 70er Jahre, Videotelefonie über herkömmliche Telefonleitungen in guter Qualität abwickeln zu können, sind mit aktuellen Codierverfahren problemlos möglich. Die hierfür notwendigen Algorithmensysteme sind zwar im Vergleich zu den ersten MPEG-Standards hochkomplex. Trotzdem können die neuen Codiersysteme erfolgreich eingesetzt werden, da das Performanceverhältnis der heutigen Mikroelektronik zu der Komplexität der Codiersysteme größer ist, als das beispielsweise zu Beginn der 90er Jahre bei der Einführung von MPEG-1 der Fall war.

Entwicklung der Codierverfahren

Abb. 7.4-1 veranschaulicht noch einmal die Codiereffizienz der verschiedenen Codiersysteme anhand des Peak-SNRs[80].

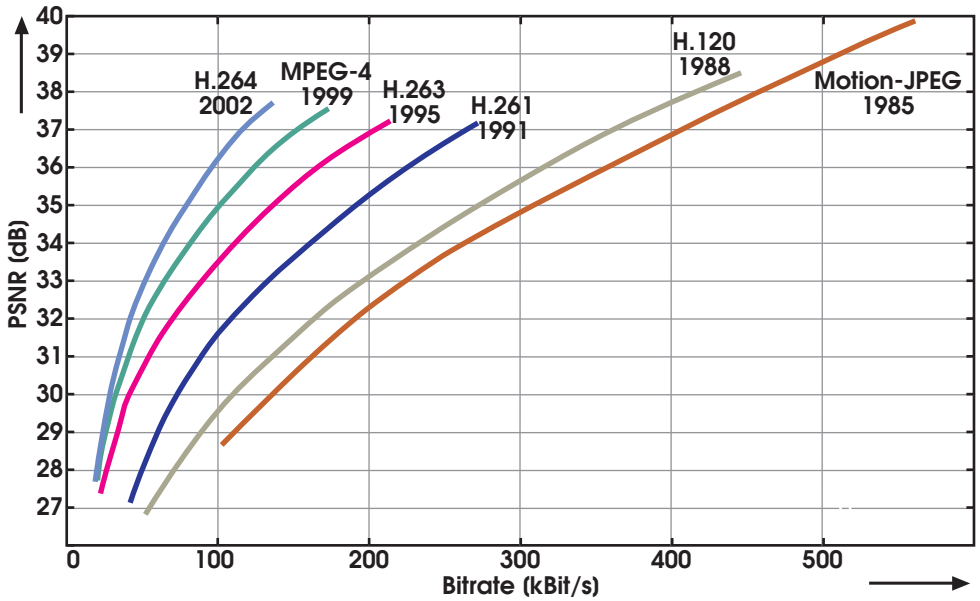


Abb. 7.4-1: Vergleich der Codiereffizienz für verschiedene Videocodierverfahren am Beispiel der Testsequenz *Foreman* (10 Hz)

80 Das hier gezeigte Beispiel bezieht sich auf die Codierung von 133 Bilder der QCIF-Testsequenz *Foreman* (allgemein anerkannte Testsequenz aus der ITU-T H.261 Entwicklung). Die Sequenz hat eine Bildwiederholrate von 10 Hz. Die Basisdaten wurden dem Beitrag [7.12] entnommen.

Sicherlich ist die Entwicklung der digitalen Bild- und Videocodierung mit den aktuellen Systemkonzepten noch nicht abgeschlossen. Auf verschiedenen Ebenen sind weitere Forschungen im Gange.

7.4.2 Motion-JPEG 2000

In Abschnitt 3.4 wurde die Wavelet-Codierung für Einzelbilder eingeführt. Der JPEG 2000 Standard beschreibt verschiedene Codiersysteme, die Wavelets verwenden. Der Teil 3 des JPEG 2000 Standards behandelt die Wavelet-basierte Videocodierung. Dieser Standard soll hier nur kurz angesprochen werden.

Intra-Frame Codierung

Im Unterschied zu den Codiersystemen der Abschnitt 7.1 bis Abschnitt 7.3 steht bei Motion JPEG 2000 nicht die Minimierung der Datenrate, sondern die möglichst hohe Bildqualität im Vordergrund. Im Teil 3 von JPEG 2000 ist zunächst eine Codierung nach dem Motion-JPEG Prinzip (vgl. Kapitel 4) vorgesehen. Damit ist JPEG 2000 prädestiniert für die Einsatzgebiete:

- Bildverarbeitung in der Medizintechnik,
- Hochauflösendes Fernsehen,
- Digitales Kino.

Für diese Anwendungen ist tatsächlich die Intraframe-Codierung von JPEG 2000, Teil 3 die erste Wahl, da eine quasitransparente Übertragung mit einer deutlich geringeren Datenrate möglich wird, als bei den DCT-basierte Verfahren.

Inter-Frame Codierung

Bei Motion JPEG 2000 ist aber überdies auch eine Inter-Frame Codierung in Anlehnung an das MPEG-Konzept denkbar. Bislang hat sich die Wavelet-Transformation mit Bewegungskompensation nicht auf breiter Front durchsetzen können, obwohl ein deutlicher Anstieg der Codiereffizienz auch bei Bewegtbildern zu erwarten ist.

7.4.3 Abschluss

Der Kurs *Digitale Bildcodierung* hat ausgehend von den Grundlagen der Bildtechnik die wichtigsten Codierverfahren für die Einzelbild- und Bewegtbildcodierung vorgestellt. Zwei zentrale Anwendungsbereiche der MPEG-2 Codierung, die Speicherung von digitalem Videomaterial auf CD oder Video DVD und das Codierkonzept für das digitale Europäische Fernsehen (*DVB*) wurden in einem separaten Kapitel detaillierter dargestellt. Trotzdem kann der Kurs nur einen kleinen Einblick in die verschiedenen Verfahren zur digitalen Bildcodierung geben. So mussten insbesondere folgende verwandte Themen in diesem Kurs ausgeklammert werden:

- Verfahren zur digitalen, dreidimensionalen Bilddarstellung und zur stereoskopischen Bildverarbeitung (*JP3D*),
- spezielle Codierverfahren für die Bildkommunikation im mobilen Einsatz (*DVB-H*),
- das Zusammenspiel zwischen Systemen der Bildkommunikation und den Netzwerkprotokollen (z. B. *Videostreaming*),

- Videokommunikation und Datenaustausch in professionellen Studioumgebungen (z. B. Datenformate wie *Multimedia Exchange Format - MX F*).

Selbsttestaufgabe 7.4-1:

Geben Sie die Profiles von H.264/AVC an!

Selbsttestaufgabe 7.4-2:

Geben Sie die kleinste Bildblockgröße an, die bei H.264/AVC transformiert werden kann!

Selbsttestaufgabe 7.4-3:

Welche Blocktypen sind bei H.264/AVC für die Bewegungskompensation möglich?

Selbsttestaufgabe 7.4-4:

Um für den H.264/AVC Standard eine Integer-Transformation anwenden zu können wird die DCT-Transfromation äquivalent nach Gl. 7.3-22 und Gl. 7.3-23 umgeformt. Wie groß ist hierin die Konstante d vor der Vereinfachung? Geben Sie d mit 3 Stellen Genauigkeit nach dem Komma an!

Selbsttestaufgabe 7.4-5:

Der Quantisierungsparameter QP sei für einen Block H.264/AVC wie folgt gegeben:

$$QP = 16.$$

Geben Sie den zugehörigen Wert für Q_{step} an!

Selbsttestaufgabe 7.4-6:

Die Testsequenz Tempete werde mit 500 kbit/s übertragen. Um wie viel differieren die Codierverfahren H.264/AVC und MPEG-2 beim Peak-SNR für diese Randbedingung?

Lösungshinweise zu Selbsttestaufgaben

Lösung zu Selbsttestaufgabe 1.2-1:

Der ungefähre Wellenlängenbereich für die sichtbare optische Strahlung liegt zwischen 380 nm und 760 nm.

Lösung zu Selbsttestaufgabe 1.2-2:

Die Sensortypen auf der menschlichen Netzhaut heißen Stäbchen und Zapfen. Die Stäbchen sind für die Helligkeitswahrnehmung, die Zapfen für die Farbwahrnehmung zuständig.

Lösung zu Selbsttestaufgabe 1.2-3:

Der Mach-Effekt äußert sich in der gegenseitigen Beeinflussung örtlich benach-barter Sinneszellen. Es handelt sich hierbei um eine Bandpasscharakteristik. Wird eine Sehzelle durch einen Lichtreiz aktiv, hemmt diese Aktivität gleichzeitig die Aktivität unmittelbar benachbarter Sinneszellen. Der Mach-Effekt ist auch an Objektkanten zu beobachten und führt in der Regel subjektiv zu einer Kontrasterhöhung im Kantenbereich.

Lösung zu Selbsttestaufgabe 1.2-4:

Die Bewegungsverschmelzungsfrequenz ist die minimale Bildwiederholfrequenz, die erforderlich ist, um in der dargestellten Folge von Einzelbildern einen kontinuierlichen Bewegungsverlauf wahrnehmen zu können.

Die Flimmerverschmelzungsfrequenz ist die minimale Bildwiederholfrequenz, die erforderlich ist, um bei vorgegeben Beleuchtungs- und Betrachtungsbedingungen kein Flimmern oder Flackern mehr wahrzunehmen.

Lösung zu Selbsttestaufgabe 1.2-5:

Werden die einzelnen Kinobilder mehrfach hintereinander projiziert (Realisierung: Shutterblende) kann der Flimmereindruck vermindert werden.

Da diese Maßnahme faktisch einer Bildwiederholung gleichkommt, kann es bei Bewegungen im Bild (Objekt- und/oder globale Bewegungen) zu Ruckelstörungen kommen.

Lösung zu Selbsttestaufgabe 1.2-6:

In Abb. 1.2-6 ist die Kontrastempfindlichkeit des visuellen Systems für farbart- und helligkeitsmodulierte räumliche Gitter dargestellt. Daraus ist, dass die Kontrast-empfindlichkeit für Farbartübergänge zu höheren Frequenzen früher abfällt als für Helligkeitsübergänge. So ist das Auflösungsvermögen des menschlichen Gesicht-sinns für Farbartübergänge (bei gleicher Helligkeit) wesentlich geringer als für Helligkeitsübergänge.

Lösung zu Selbsttestaufgabe 1.3-1:

Additive Farbmischung: Hier vermischen sich die Farbanteile von selbst leuch-tende Farbpunkten. Aus großer Entfernung betrachtet werden die einzelnen Farbpunkte nicht mehr separat aufgelöst und verschmelzen damit additiv zu einer neuen Farbe. Beispiel für dieses Farbmischprinzip ist die Farbdarstellung eines Farb-Kathodenstrahlmonitors.

Subtraktive Farbmischung: Hierbei durchdringt ein spektral weißes Licht mehrere hintereinander liegende Farbstoffe (quasi als Farbfilter). Die verschiedenen Farbstoffe besitzen unterschiedliche spektrale Absorptionseigenschaften und entziehen somit dem Lichtstrahl verschiedene spektrale Anteile. Übrig bleibt der spektrale Anteil des Lichtes, der von keinem Farbstoff absorbiert wurde. Ein Beispiel für die subtraktive Farbmischung ist das Zusammenführen verschiedenfarbiger Farb-stoffe, wie es bei der Malerei geschieht. Die subtraktive Farbmischung wird vielfach auch bei Farbdruckern verwendet, die mit drei oder mehr Grundfarben arbeiten. Als Grundfarben werden oft gelb, magenta, cyan und ggf. weitere ver-wendet.

Lösung zu Selbsttestaufgabe 1.3-2:

Metamere Farben sind Farbvalenzen, die gleich aussehen, nicht aber unbedingt die gleiche spektrale Verteilung besitzen.

Lösung zu Selbsttestaufgabe 1.3-3:

Mit einem EBU- oder SMPTE-Farbmonitor lassen sich nicht alle in der Natur vorkommenden Farbarten darstellen, da nur solche dargestellt werden können, die innerhalb des jeweiligen Farbdreiecks liegen (vgl. Abb. 1.3-5). Es lassen sich Farborte außerhalb des jeweiligen Farbdreiecks zwar in Bezug auf den Farbton kor-rekt darstellen, nicht aber mit der vollständigen Farbsättigung. Die maximal mögliche Farbsättigung liegt auf der Begrenzung des jeweiligen Farbdreiecks selbst.

Lösung zu Selbsttestaufgabe 1.3-4:

Als praktischen Unbuntpunkt wird der Weißpunkt D65 für beide Farbdreiecke verwendet: D65(x, y, z) = (0.3127, 0.3290, 0.3582).

Lösung zu Selbsttestaufgabe 1.4-1:

Die Bildvorlage wird in horizontale, vertikal aneinander angrenzende Zeilen zerlegt. Dazu wird der Bildinhalt die Zeilen von oben links anfangend zeilenweise nacheinander bis zum Ende der letzten Zeile unten rechts ausgelesen.

Lösung zu Selbsttestaufgabe 1.4-2:

Die nominelle Zeilenzahl ist größer als die aktive Zeilenzahl. Die Differenz ergibt sich aus dem Wunsch, eine gewisse Zeit zwischen der letzten aktiven Zeile des Bildes n bis zum Beginn der ersten Zeile des Bildes n+1 zu lassen. Diese Zeit wird empfängerseitig für den vertikalen Rücklauf eines Wiedergabesystems mit Kathodenstrahlröhre benötigt, der nicht beliebig kurz sein kann (vgl. Abb. 1.4-2 roter Sägezahn).

Lösung zu Selbsttestaufgabe 1.4-3:

Ähnlich wie beim vertikalen Rücklauf (vgl. Selbsttestaufgabe 1.4-2) wird auch Zeit für einen horizontalen Zeilenrücklauf vorgesehen, da der für die Ansteuerung notwendige Sägezahn (in Abb. 1.4-2 blau eingezeichnet) in der Rücklaufphase nicht beliebig steil sein kann.

Lösung zu Selbsttestaufgabe 1.4-4:

Die erforderliche Transformationsgleichung ist in Gl. 1.4-9 dargestellt:

$$\begin{pmatrix} U \\ V \\ Y \end{pmatrix} = \begin{pmatrix} -0.147 & -0.289 & +0.436 \\ +0.615 & -0.515 & -0.100 \\ +0.299 & +0.587 & +0.114 \end{pmatrix} \cdot \begin{pmatrix} E'_R \\ E'_G \\ E'_B \end{pmatrix}$$

Lösung zu Selbsttestaufgabe 2.2-1:

Für eine fehlerfreie Rekonstruktion eines abgetasteten Signals muss die Abtastfrequenz f_T mindestens doppelt so hoch sein, wie die höchste vorkommende Frequenz des Eingangssignals. Dann überlappen sich die Spektren nicht und das analoge Ausgangssignal kann theoretisch exakt wieder durch ein geeignetes Rekonstruktionsfilter zurück gewonnen werden.

Formal gilt also:

$$f_T \geq 2 \cdot f_{max}.$$

Lösung zu Selbsttestaufgabe 2.2-2:

Unter der Diskretisierung versteht man eine örtliche und/oder zeitliche Aufteilung des Signals in eine beschränkte Anzahl von (kontinuierlichen) Werten, die das Ausgangssignal repräsentieren.

Unter der Quantisierung versteht man das Runden eines analogen Ausgangssignals in eine beschränkte Anzahl von Wertestufen. In der Regel wird die Quantisierung in 2N verschiedene Stufen vorgenommen. In der digitalen Bildtechnik sind Wortbreiten von N = 8 und N = 10 gebräuchlich.

Lösung zu Selbsttestaufgabe 2.2-3:

Die Abtastformate *M*:*N*:*O*, (*M*, *N*, *O* natürliche Zahlen) geben das Abtastverhältnis zwischen *Y*, $C_B$und C_R, bezogen auf eine Basisfrequenz von $f_{\perp,Basis} = 3.375$ MHz an. Für 4:4:4 ergibt sich ein Abtastfrequenz von je 13.5 MHz, der Studiostandard arbeitet mit einem Abtastverhältnis von 4:2:2. Hier sind also die Farbdifferenzsignale horizontal um den Faktor 2 niedriger abgetastet (6.75 MHz) als das Helligkeitssignal. Bei 4:1:1 sinkt die Abtastfrequenz für die Farbdifferenzsignale noch einmal um den Faktor 2 auf eine Frequenz von 3.375 MHz ab. Es handelt sich insgesamt also um eine Farbunterabtastung in horizontaler Richtung um den Faktor 4.

Das Abtastformat 4:2:0 gibt nicht mehr das Abtastverhältnis zwischen *Y*, $C_B$und C_R an. Vielmehr handelt es sich hierbei um eine horizontale und vertikale Unterabtastung der Farbdifferenzsignal jeweils um den Faktor 2.

Lösung zu Selbsttestaufgabe 2.2-4:

Beim ITU-R Standard BT.601 wird die Luminanz bei einem 625-Zeilen-System mit 13.5 MHz abgetastet. Hier kommt das Abtastformat 4:2:2 zum Einsatz, welches sich aus dem vierfachen der Basisfrequenz $f_{\perp,Basis} = 3.375\ MHz$ errechnet .

$$f_{\perp} = 4 \cdot 3.375\ MHz = 13.5\ MHz.$$

Lösung zu Selbsttestaufgabe 2.2-5:

Beim abgetasteten Luminanzsignal werden die Bereiche zwischen 1 und 15 (8-Bit-Codierung) bzw. 1 und 63 (10-Bit-Codierung) und die Bereiche zwischen 236 und 254 (8-Bit-Codierung) bzw. 941 und 1022 (10-Bit-Codierung) nicht für das eigentliche Videosignal verwendet. Diese Bereiche dienen als Über- bzw. Untersteuerungsreserve für Überschwinger des Analogsignals. Bei den Chrominanzsignalen sind dies die Bereiche zwischen 1 und 15 bzw. 1 und 63 sowie zwischen 241 und 254 bzw. 961 und 1022. Die Werte 0 und 255 (8-Bit-Codierung) bzw. 0 und 1023 (10-Bit-Codierung) sind für spezielle Meldungen, die Synchronisationszwecken dienen reserviert.

Lösung zu Selbsttestaufgabe 3.2-1:

Die Transformation dient der Dekorrelierung des Bildsignals.

Lösung zu Selbsttestaufgabe 3.2-2:

- Zu a. Oben links ist der DC-Koeffizient (Gleichanteil des Bildes).
- Zu b. In der ersten Zeile ganz rechts wird die höchste rein horizontale Orts-Frequenz für den Block dargestellt.
- Zu c. In der ersten Spalte ganz unten wird die höchste rein vertikale Orts-Frequenz für den Block dargestellt.

Lösung zu Selbsttestaufgabe 3.2-3:

Mit M = N = 2 ergibt sich:

$$F_{DCT}(u,v) = C(u) \cdot C(v) \sum_{x=0}^{1} \sum_{y=0}^{1} f(x,y) \cdot \cos\left[\frac{(2x+1)u \cdot \pi}{16}\right] \cos\left[\frac{(2y+1)v \cdot \pi}{16}\right] \qquad 3.2\text{-}4$$

$$C(u) = \begin{cases} \sqrt{\frac{1}{2}} & \text{für } u = 0 \\ 1 & \text{für } u \neq 0 \end{cases};$$

$$C(v) = \begin{cases} \sqrt{\frac{1}{2}} & \text{für } v = 0 \\ 1 & \text{für } v \neq 0 \end{cases}$$

Die Tabelle kann so aussehen:

u, v	$C_u \cdot C_v$	x = 0, y = 0	x = 1, y = 0	x = 0, y = 1	x = 1, y = 1	Summe
u = 0, v = 0	$\frac{1}{2}$	10	15	12	9	46
u = 1, v = 0	$\frac{1}{\sqrt{2}}$	$\frac{10}{\sqrt{2}}$	$-\frac{15}{\sqrt{2}}$	$\frac{12}{\sqrt{2}}$	$-\frac{9}{\sqrt{2}}$	$-\sqrt{2}$
u = 0, v = 1	$\frac{1}{\sqrt{2}}$	$\frac{10}{\sqrt{2}}$	$\frac{15}{\sqrt{2}}$	$-\frac{12}{\sqrt{2}}$	$-\frac{9}{\sqrt{2}}$	$12 \cdot \sqrt{2}$
u = 1, v = 1	1	5	–7.5	–6	4.5	–4

Damit ergibt sich:

$$F(u,v) = \begin{pmatrix} 23 & -1 \\ 12 & -4 \end{pmatrix}$$

Lösung zu Selbsttestaufgabe 3.2-4:

Die Quantisierung der DCT-Koeffizienten dient der eigentlichen Redundanzreduktion: Ähnliche Werte der Koeffizientenmatrix werden durch die Rundung einem identischen Wert zugeordnet. Gleiche Werte in Folge können effektiv zusammengefasst werden. In der Praxis sind viele höherwertige DCT-Koeffizienten nahe Null und werden bei der Quantisierung zu Null gesetzt.

Mit dem Grad der Quantisierung wird die Qualität und damit auch das erforderliche Datenaufkommen reguliert.

Lösung zu Selbsttestaufgabe 3.3-1:

Der Zick-Zack-Scan sorgt für eine möglichst optimale Zusammenfassung der quantisierten DCT-Koeffizienten. Er berücksichtigt die erwartete Verteilung von Koeffizienten, die in langen *Run*s anschließend gut zusammengefasst werden können.

Lösung zu Selbsttestaufgabe 3.3-2:

Die Lauflängencodierung fasst gleiche Zahlenwerte in sogenannten Runs zusammen.

Lösung zu Selbsttestaufgabe 3.3-3:

Die variable Lauflängencodierung ordnet häufig vorkommenden Runs kurze Codeworte zu und seltenen lange.

Lösung zu Selbsttestaufgabe 3.4-1:

Für jeden DCT-Block ist eine Mindestanzahl an Daten für die Beschreibung notwendig.

Lösung zu Selbsttestaufgabe 3.4-2:

Die Basisfunktionen für die Wavelet-Transformation sind besser auf die Bildstrukturen angepasst. Es werden Basisfunktionen mit endlicher Ausdehnung verwendet. Damit lassen sich beispielsweise Kanten besser annähern.

Lösung zu Selbsttestaufgabe 3.4-3:

Die Basisfunktionen sind die sogenannten *Scaling-Funktionen* (*Mutter-Wavelets*), die eine *Tiefpass-Funktion* übernehmen. Die *Wavelets* haben *Hochpass-Funktion*.

Lösung zu Selbsttestaufgabe 3.4-4:

Die Basisfunktionen sind die sognennten *Scaling-Funktionen* (*Mutter-Wavelets*), die eine *Tiefpass-Funktion* übernehmen. Die *Wavelets* haben eine *Hochpass-Funktion*.

Lösung zu Selbsttestaufgabe 3.5-1:

Erlaubte Operationen sind Stauchen, Verschieben, Vervielfältigen, Drehen und Änderung der Luminanzwerte von ausgewählten Bildbereichen.

Lösung zu Selbsttestaufgabe 3.5-2:

Bekannt sind die Quadtree-Zerlegung und die Triangular-Zerlegung.

Lösung zu Selbsttestaufgabe 3.5-3:

Der modifizierte Kopierer führt eine oder mehrere Transformationen auf ein Ausgangsbild aus, und setzt die Ergebnisse zusammen.

Lösung zu Selbsttestaufgabe 3.5-4:

Die Transformationen müssen *kontraktiv* sein, d. h. die Transformationen dürfen bei wiederholter Anwendung immer nur kleinere Bildstrukturen erzeugen, als vorher vorhanden sind.

Lösung zu Selbsttestaufgabe 3.5-5:

Für die Messung der Bildqualität wird der *Peak-SNR* (*PSNR*) herangezogen und in dB angegeben:

$$PSNR_A(s, s') = 10 \log_{10} \left(\frac{255^2}{MSE_A(s, s')} \right), \qquad 3.5\text{-}3$$

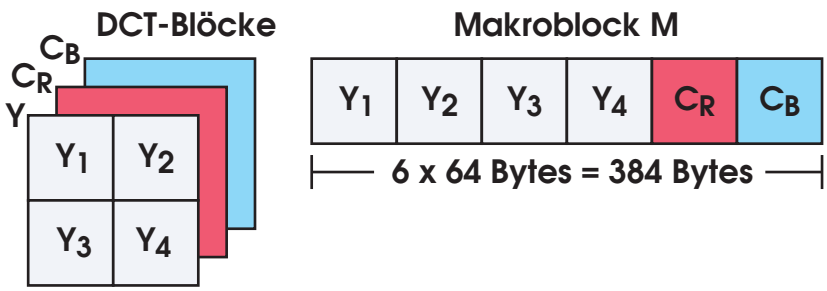
$$\text{mit } MSE_A(s, s') = \frac{1}{|A|} \sum_{(x,y) \in A} [s(x, y) - s'(x, y)]^2.$$

Lösung zu Selbsttestaufgabe 4.1-1:

Beim DV25 Algorithmus das zu erwartende Datenaufkommen im *vorhinein* ermittelt, damit die erforderliche Datenrate nicht nur im Mittel, sondern exakt für jedes Videosegment (bestehend aus 5 Makroblöcken, vgl. Abb. 4.1-3) erreicht wird. Ein exakt konstantes Datenaufkommen auf Videosegmentebene ist für die Magnetbandaufzeichnung notwendig.

Lösung zu Selbsttestaufgabe 4.1-2:

In Abb. 4.1-3 ist das Makroblockbild für die DV25 Codierung nach 4:1:1 Abtastung dargestellt. Bei der 4:2:0 Abtastung ist zu beachten, dass sowohl horizontal, als auch vertikal eine Unterabtastung der Farbdifferenzsignale geschieht. Daher muss die Makroblockbildung wie folgt geschehen:



Lösung zu Selbsttestaufgabe 4.1-3:

Das Block-Shuffling führt dazu, dass jeder der fünf Makroblöcke eines Videosegments in seinem Gehalt an Redundanz und Irrelevanz sehr unterschiedlich sein kann. Bezogen auf alle Videosegmente eines einzelnen Videobildes können mit diesem Verfahren die Redundanzunterschiede minimiert werden. So kann die Forderung leichter erfüllt werden, für jedes Videosegment eine konstante Datenkompression zu erreichen, ohne dass die korrespondierende Bildqualität in „schwierigen“ Bildbereichen schlechter sein muss.

Lösung zu Selbsttestaufgabe 4.1-4:

Die Einteilung in eine der vier Gruppen charakterisiert grob die spektrale Energieverteilung im betrachteten Einzelblock. Die Zuordnung eines Blockes in eine der vier Gruppen geschieht anhand des betragsgrößten AC-DCT-Koeffizienten. Gruppe 3 ist für sehr fein strukturierten, Gruppe 0 für groben bzw. homogenen Bildinhalt optimiert. Die Gruppen 1 und 2 decken die dazwischen liegenden Fälle ab. Die 16 Quantisierungsstufen dienen der Einstellung der Datenrate.

Lösung zu Selbsttestaufgabe 4.2-1:

Die Feed-Forward DPCM (D*PCM) ist in Abb. 4.2-1, die Feed-Backward DPCM (DPCM) in Abb. 4.2-3 dargestellt.

Bei der DPCM wird der Prädiktor des Empfängers möglichst exakt bereits auf der Coderseite nachgebildet. Bei der DPCM muss daher im Unterschied zur D*PCM der Prädiktor auf der Coderseite ein brauchbares Prädiktorsignal aus dem quantisierten Signal erzeugen können. Man kann theoretisch zeigen, dass die DPCM für die spektrale Energieverteilung von natürlichen Bildvorlagen einen Vorteil bringt. Bei entsprechender Wahl des DPCM-Decoders kann zudem gewährleistet werden, dass im Unterschied zur D*PCM alle Übertragungsfehler asymptotisch abklingen. Damit ist bei der DPCM ein wichtiges Stabilitätskriterium erfüllt.

Lösung zu Selbsttestaufgabe 4.2-2:

Liegt eine unbewegte Bildszene vor (Stillbild), dann kann als Prädiktor für eine DPCM ein Bildspeicher verwendet werden, da dann eine sehr gute Prädiktion gelingt. Im Gegensatz dazu ist der Bildspeicher ungeeignet, wenn große Bildbereiche einer starken zeitlichen Veränderung unterworfen sind. Dies ist insbesondere dann der Fall, wenn Kamerabewegungen (Schwenk, Zoom) vorliegen. Dann ändern sich alle Bildbereiche gleichzeitig und aus dem Bildspeicher kann kein Bildbereich direkt prädiziert werden.

Lösung zu Selbsttestaufgabe 4.3-1:

In Analogie zu Gl. 4.3-5 kann die diskrete MAD wie folgt angegeben werden:

$$MAD_{\perp}(d_x, d_y) = \frac{1}{m \cdot n} \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} |s_k(x,y) - s_{k-1}(x-i-d_x, y-j-d_y)|.$$

Lösung zu Selbsttestaufgabe 4.3-2:

Anhand eines einfachen Beispiels (Verschiebung einer Objektkante) kann gezeigt werden, dass die Zielfunktion mehrere Minima hat. Überstreicht der Referenzblock des Blockmatching Verfahrens einen Teil der Kante, kann die Verschiebung senkrecht zur Kantenrichtung exakt bestimmt werden. Längs zur Kantenrichtung ist aber eine Positionierung des Referenzblockes nicht exakt möglich, da diese für den Referenzblock überall gleich aussieht.

Beim Vorliegen von periodische Strukturen, Helligkeitsschwankungen und Auf- und Verdeckungseffekte sind absolute Minima der Zielfunktion mit Hilfe des Blockmatching Verfahrens ebenfalls oft nicht erreichbar. Entweder liegen Mehrdeutigkeiten vor (z. B. bei periodische Strukturen) oder der Referenzblock besitzt keine Korrespondenz im Suchbereich (z. B. in Auf- und Verdeckungsbereichen).

Lösung zu Selbsttestaufgabe 4.3-3:

Gl. 4.3-6 (Full-Search – FS) und Gl.4.3-8 (Three-Step Search – TSS, Cross-Search Algorithms – CSA, 2D Logarithmic Search – TDL), geben die zugehörigen Formeln für die Berechnung an:

$$\begin{aligned} N_{FS} &= (2d_{x,\max} + 1) \cdot (2d_{y,\max} + 1), \\ N_{TSS} &= 1 + 8 \cdot \log_2(d_{\max}), \\ N_{TDL} &= N_{CSA} = 5 + 4 \cdot \log_2(d_{\max}). \end{aligned} \qquad \text{4.3-9}$$

Mit $d_{\max}$ = 8 ergibt sich:

$$\begin{aligned} N_{FS} &= (2 \cdot 8 + 1) \cdot (2 \cdot 8 + 1) = 289, \\ N_{TSS} &= 1 + 8 \cdot \log_2(8) = 1 + 8 \cdot 3 = 25, \\ N_{TDL} &= N_{CSA} = 5 + 4 \cdot \log_2(8) = 5 + 4 \cdot 3 = 17. \end{aligned} \qquad \text{4.3-10}$$

Lösung zu Selbsttestaufgabe 5.2-1:

Unter dem *Prädiktionsfehlersignal* oder *Prädiktionsfehlerbild* versteht man das verbleibende Restbild bzw. Restsignal, welches sich aus der Differenz des aktuell zu codierenden Originalbildes und einem prädizierten Bild errechnet. Das prädizierte Bild kann dabei aus einem oder mehreren benachbarten Bildern mit zusätzlicher Verschiebungsinformation ermittelt worden sein.

Lösung zu Selbsttestaufgabe 5.2-2:

Ein Makroblock hat 16 x 16 = 256 Bildpunkte. Ein ITU-R BT.601 Bild hat 720 x 576 Bildpunkte. Pro Sekunde werden 25 Vollbilder übertragen. Dann ergibt sich *N* als die Anzahl der gesuchten Makroblöcke pro Sekunde zu:

$$N = \frac{720 \cdot 576}{256} \cdot 25 = 40500 \text{ Makroblöcke.}$$

Lösung zu Selbsttestaufgabe 5.2-3:

Nachfolgende Abb. 5.2-15 zeigt die Umsortierung der *closed* 10er GOP aus Abb. 5.2-10.

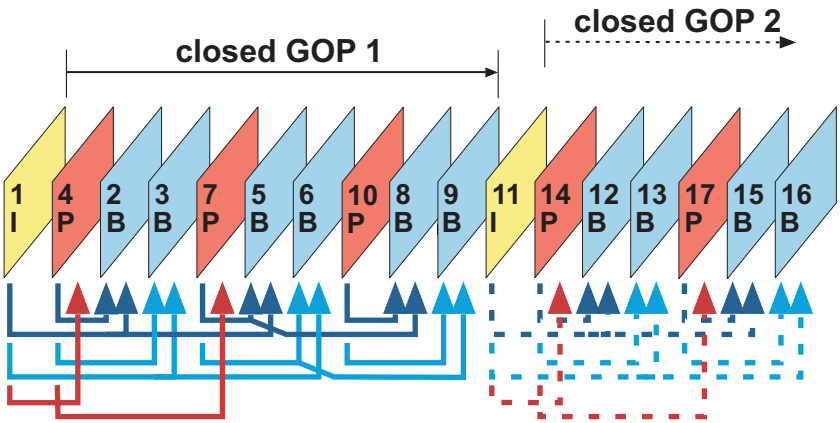


Abb. 5.2-15: Bild-Umsortierung am Beispiel einer *closed* 10er-GOP mit Prädiktionsbeziehungen zwischen *I*-, *P*-, und *B-Bildtypen*

Lösung zu Selbsttestaufgabe 5.3-1:

Die sieben MPEG-2 Profiles lauten: *Simple*, *Main*, *High*, *4:2:2*, *SNR*, *Spatial* und *Multiview*.

Quellmaterial von ITU-R BT.601 besitzt eine Auflösung von 720 x 576 Bildpunkten. Diese Auflösung ist im *Main-Level* verfügbar. Dieser Level ist für alle Profiles außer *Spatial* möglich. Üblich ist für *DVD* und *DVB MP@ML*; für Studioanwendungen ist es *422P@ML*.

Lösung zu Selbsttestaufgabe 5.3-2:

Es ist die *Frame-Codierung* und die *Field-Codierung* möglich. Zusätzliche *Modes* sind der *16 x 8 Mode* und die *Dual Prime Modes*.

Lösung zu Selbsttestaufgabe 5.3-3:

Beim Alternate Scanning wird die Zusammenfassung im Gegensatz zum Zick-Zack-Scan mehr vertikal durchgeführt.

Bei der vertikalen Zusammenfassung von 16 x 8 Bildmaterial auf einen 8 x 8 DCT-Block, werden die auftretenden vertikalen Frequenzen verdoppelt, wobei zu beachten ist, dass aufgrund der fehlenden Vorfilterung vertikaler Alias möglich ist. Als Ergebnis verbleiben auf jeden Fall mehr vertikal hochfrequente AC-Koeffizienten im 8 x 8 DCT-Block nach der Quantisierung. Um diese Koeffizienten möglichst früh in die Zusammenfassung integrieren zu können, muss die das *Alternate Scanning* vertikale AC-Koeffizienten vor den entsprechenden horizontalen AC-Koeffizienten berücksichtigen.

Lösung zu Selbsttestaufgabe 5.3-4:

Die Paketlänge des PS ist variable und sehr lang. Fehler führen dann zu erheblichen Störungen im Datenmaterial. TS-Pakete haben eine feste Länge von 188 Bytes, lassen sich daher besser in die Paketstruktur des Internets integrieren.

Lösung zu Selbsttestaufgabe 5.3-5:

Die *System Clock Reference – SCR* leitet sich beim *Programmstrom* (*PS*) aus der *System Time Clock – STC* ab. Es handelt sich hierbei demnach um eine zentrale Zeitbasis. Im *Transportstrom* (*TS*) wird für jedes einzelne Programm eine eigene Zeitbasis berücksichtigt, die als *Program Clock Reference – PCR* bezeichnet wird und für jedes Programm jeweils aus der individuellen *STC* abgeleitet wird. Die in den einzelnen Programmen entstehenden *PCRs* müssen nicht zueinander synchron sein.

Lösung zu Selbsttestaufgabe 6.2-1:

Nach der *CD-I* Spezifikation ist das spezielle Betriebssystem *OS-9* notwendig.

Lösung zu Selbsttestaufgabe 6.2-2:

Die Audio- und Videopakete sind immer in den *VOBU*-Einheiten miteinander verschachtelt. Es existiert kein getrennter übergeordneter Audiodatenstrom, zu dem mehrere Videoströme über ILVU Zugriff hätten.

Lösung zu Selbsttestaufgabe 6.2-3:

Ein Pack (*V_PCK*, *A_PCK*, *SP_PCK*) belegt genau 2048 Byte. Es wird ein Standard MPEG-2 Programmstrom Multiplexverfahren angewendet.

Lösung zu Selbsttestaufgabe 6.3-1:

Bei MPEG-2 sind folgende Tabellen definiert:

- *Program Association Table – PAT*,
- *Program Map Table – PMT*,
- *Conditional Access Table – CAT*,
- *Network Information Table – NIT*.

Die wichtigsten SI-Tabellen bei DVB sind:

- *Bouquet Association Table* (*BAT*),
- *Service Description Table* (*SDT*),
- *Event Information Table* (*EIT*),
- *Time and Data Table* (*TDT*),
- *Running Status Table* (*RST*).

Lösung zu Selbsttestaufgabe 6.3-2:

Aus Abb. 5.3-11 ist zu ersehen, dass für die PID 13 Bit vorgesehen sind. Damit lassen sich *8191* verschiedene Werte adressieren. Der Wert *0* ist reserviert für die PAT, der Wert *8191* wird oft für ein Datenpaket mit Stopfbits (freie Kapazität) verwendet.

Lösung zu Selbsttestaufgabe 6.5-1:

Die Energieverwischung ist in Animation 6.4-1 dargestellt. Die Energieverwischung wird über den Enable-Eingang ein- und ausgeschaltet. Dieser ist über eine UND-Schaltung mit dem Ausgang des Pseudozufallszahlengenerator zusammengeschaltet. Eine logische „1“ am Enable-Eingang schaltet die Energieverwischung ein, eine logische „0“ schaltet sie wieder aus.

Lösung zu Selbsttestaufgabe 6.5-2:

10% fehlerhafte Symbole lassen sich mit 20 Symbolen korrigieren. Es entsteht ein RS(120,100) Code.

Lösung zu Selbsttestaufgabe 6.5-3:

$G_x = 1110110_{BIN} = 166_{OCT}$, $G_y = 1010111_{BIN} = 127_{OCT}$.

Lösung zu Selbsttestaufgabe 6.5-4:

Es ist die Gl. 6.4-9 für die Kanalcodierung und die Gl. 6.5-4 für die Leitungscodierung heranzuziehen. Der Roll-Off-Faktor der Basisbandfilterung sei ideal: r_{DVB-S} = 0.35. Zunächst wird aus Gl. 6.5-4 mit r_{DVB-S} die Bruttodatenrate berechnet:

$$R_{Brutto} = \frac{2\,Bit/Symbol}{1.35\,\frac{Hz}{Symbole/s}} \cdot 46\ MHz = 68.15\ Mbit/s$$

In Gl. 6.4-9 stellt R_1 die Coderate des RS-Codes dar. Ein *RS(204,188)* besitzt die Coderate:

$$R_1 = \frac{188}{204} \approx 0.92157.$$

Die Coderate des Faltungscodes ist in der Aufgabenstellung angegeben. Es ergeben sich damit folgende Ergebnisse:

Coderate R_2	1/2	2/3	3/4	5/6	7/8
Nettodatenrate [Mbit/s]	31.40	41.86	47.10	52.34	56.95

Lösung zu Selbsttestaufgabe 6.5-5:

Es gilt:

$$d = T_G \cdot c_0 = \frac{T_G}{T_S} \cdot T_S \cdot c_0. \qquad 6.5\text{-}16$$

Aus Tabelle 6.5-2 ergibt sich: T_S = 224 µs (*2k-Modus*) und T_S = 896 µs (*8k-Modus*). Damit lassen sich die Werte errechnen:

Genutzte Symboldauer T_S	224 µs				896 µs			
T_G / T_S	1/4	1/8	1/16	1/32	1/4	1/8	1/16	1/32
Max. Senderabstand [km]	16.8	8.4	4.2	2.1	67.2	33.6	16.8	8.4

Lösung zu Selbsttestaufgabe 7.1-1:

- Die *Genauigkeit der Bewegungskompensation* wurde von einem auf einen halben Bildpunkt angehoben.
- Zusätzlich zu *CIF* (352 x 288) und *QCIF* (176 x 144) sind auch *SQCIF* (128 x 96), *4CIF* (704 x 576) und *16CIF* (1408 x 1152) möglich.
- Die *Effizienz der variablen Lauflängencodierung* wurde gesteigert.

Lösung zu Selbsttestaufgabe 7.1-2:

Von oben nach unten sind folgende Hierarchieebenen bei H.263 zu unterscheiden: *Sequence Layer*, *Picture Layer*, *Group of Blocks Layer*, *Macroblock Layer* und *Block Layer*. Der *Group of Blocks Layer* wird bei H.263 oft auch *Slice Layer* genannt.

Lösung zu Selbsttestaufgabe 7.1-3:

Im *Unrestricted Motion Vector Mode* (Annex D) von ITU-T H.263 ist der Bereich der Verschiebungsvektoren auf [-31.5; 31.5] erweitert worden. Damit sind bei ITU-T H.263 schnellere Bewegungen im Bild mit Hilfe der Bewegungskompensation darstellbar.

Lösung zu Selbsttestaufgabe 7.1-4:

Beim *PB-Frame Mode* (Annex G) von ITU-T H.263 wird eine gemeinsame Codierung von zwei aufeinanderfolgenden Einzelbildern vorgenommen. Die Bezeichnung „*PB*“ gibt an, dass es sich hierbei um P- bzw. B-Bilder handelt, wie sie schon bei MPEG definiert wurden.

Lösung zu Selbsttestaufgabe 7.1-5:

ITU-T H.263 V2 besitzt eine sehr große Anzahl von verschiedenen Codieroptionen. Damit der Decoder nicht sämtliche Codieroptionen vorhalten muss, gibt es in der Version 2 des Standards drei verschiedene Levels, welche sinnvolle Kombinationen von Codieroptionen festlegen.

Lösung zu Selbsttestaufgabe 7.1-6:

Im Annex X „Profiles and Levels Definition“ wird aus den vielen Codieroptionen eine neue Profile und Level Struktur zusammengestellt. Hierin sind 9 Profiles definiert, die sich mit maximal 7 Levelstufen kombinieren lassen.

Lösung zu Selbsttestaufgabe 7.2-1:

Mit MPEG-4 Video lassen sich nicht nur normale rechteckig begrenzte Objekte codieren, sondern auch beliebig umrandete Objektkonturen.

Lösung zu Selbsttestaufgabe 7.2-2:

BIFS ist die mächtige Szenenbeschreibungssprache (*Binary Format for Scenes – BIFS*) von MPEG-4. Mit ihr stehen Autoren von Multimedia-Anwendungen zahlreiche Gestaltungsmöglichkeiten und Interaktionselemente zur Verfügung. Darüber hinaus kann der Anwender das Erscheinungsbild einer *Szene* frei an seine Vorstellungen anpassen.

Lösung zu Selbsttestaufgabe 7.2-3:

Die Verschiebungsvektoren haben maximal eine Genauigkeit von 1/4 Bildpunkt.

Lösung zu Selbsttestaufgabe 7.2-4:

Die Strukturebenen von MPEG-4 Video lauten von oben nach unten *Video Session* (*VS*), *Video Object* (*VO*), *Video Object Layer* (*VOL*), *Group of VOPs* (*GOV*) und *Video Object Plane* (*VOP*).

Lösung zu Selbsttestaufgabe 7.2-5:

Bei MPEG-4 Version 1 unterscheidet man 5 verschiedene Profiles: *Simple Visual Profile* (*SP*), *Core Visual Profile* (*CP*), *Main Visual Profile* (*MP*), *Simple Scalable Profile* (*SSP*) und *N-Bit Visual Profile*.

Lösung zu Selbsttestaufgabe 7.2-6:

Bei der DivX-Videocodierung wird das MPEG-4 Profile *Advanced Simple Profile* (*ASP*) verwendet. Es kann nur rechteckige Bilder verarbeiten.

Lösung zu Selbsttestaufgabe 7.2-7:

SNHC bedeutet *Synthetic and Natural Hybrid Coding*. Man versteht darunter die gemeinsame Codierung von natürlichen und synthetischen Objekten bei MPEG-4. Ein Beispiel ist die Facial Animation, bei der ein natürliches Gesicht mit Mimik-Parametern animiert wird.

Lösung zu Selbsttestaufgabe 7.4-1:

Bei H.264/AVC unterscheidet man 3 Profiles: *Baseline Profile, Main Profile* und *Extended Profile*.

Lösung zu Selbsttestaufgabe 7.4-2:

Die kleinste Bildblockgröße bei H.264/AVC ist 4 x 4 Bildpunkte groß.

Lösung zu Selbsttestaufgabe 7.4-3:

Die vier verschiedenen Bildtypen bei H.264/AVC sind: 8 x 8 Pixel, 8 x 4 Pixel, 4 x 8 Pixel und 4 x 4 Pixel (vgl. Abb. 7.3-2 unten).

Lösung zu Selbsttestaufgabe 7.4-4:

Mit Gl.7.3-21 und Gl. 7.3-23 ergibt sich

$$d = \frac{c}{b} = \frac{\cos\left(\frac{3\pi}{8}\right)}{\cos\left(\frac{\pi}{8}\right)} \approx 0.414$$

Lösung zu Selbsttestaufgabe 7.4-5:

Eine Erhöhung von *QP* um 6 führt genau zu einer Verdoppelung von Q_{step}. Man nehme den nächst niedrigen Wert von *QP* aus Tabelle 7.3-2, der mit 6er-Schritten erreichbar ist, also:

$$QP = 16 - 6 = 10.$$

Für diesen Wert von *QP* kann Q_{step} abgelesen werden:

$$Q_{step}(QP = 10) = 2.$$

Dann wird Q_{step} verdoppelt, um Q_{step} für einen *QP*-Wert von 16 zu erhalten:

$$Q_{step}(QP = 16) = 2 \cdot Q_{step}(QP = 10) = 2 \cdot 2 = 4.$$

Lösung zu Selbsttestaufgabe 7.4-6:

Die PSNR-Werte können aus Abb. 7.3-8 abgelesen werden. MPEG-2 besitzt bei 500 kbit/s ein PSNR von etwa 28 dB. H.264/AVC bringt es für diese Datenrate etwa auf 32.7 dB. Es ergibt sich daraus eine Differenz von etwa 4.7 dB.

Übungsaufgaben

Übungsaufgaben zu Kapitel "Einführung in die Bildtechnik"

Aufgabe 1: Ein Kinofilm (24 Hz) soll zur Flimmerreduktion mit der dreifachen Bildwiederholfrequenz wiedergegeben werden. Dazu wird jedes Kinoeinzelbild bei der Projektion mit Hilfe einer Umlaufblende dreimal dargestellt. **15 Pkt**

Es soll die maximale Darstellungsdifferenz eines bewegten Objektes berechnet werden, wenn dieses innerhalb einer Sekunde horizontal die Hälfte der Bildschirmbreite überstreicht!

a) Geben Sie zunächst die Bildwiederholfrequenz der Bildwiedergabe in [Hz] an! **3 Pkt**

b) Geben Sie die Zeitspanne *T* zwischen zwei Bildern der Bildwiedergabe in [ms] (2 Stellen Genauigkeit nach dem Komma) an! **3 Pkt**

c) Geben Sie den prozentualen Anteil der Bildbreite an (2 Stellen Genauigkeit nach dem Komma), den das bewegte Objekt während der Zeitspanne *T* zurücklegt! **3 Pkt**

d) Wie groß ist die maximale Darstellungsdifferenz in Prozent (2 Stellen Genauigkeit nach dem Komma) bezogen auf die Bildbreite? **3 Pkt**

e) Die gleiche Darstellung werde mit einem progressiv arbeitenden HD-Videosystem vorgenommen. Das Display besitze das 1920 x 1080 aktive Bildpunkte. Geben Sie die maximale Darstellungsdifferenz bei sonst gleichen Parametern in Bildpunkten an (aufrunden)! **3 Pkt**

Aufgabe 2: Charakterisieren Sie die CIE-Normspektralwertkurven X(λ), Y(λ) und Z(λ) **6 Pkt**

a. Die Kurven können positive und negative Werte annehmen.

b. Die Kurven können nur positive Werte annehmen.

c. Die Z(λ)-Kurve besitzt zwei Maxima.

d. Die Y(λ)-Kurve entspricht der Hellempfindlichkeitsverteilung des menschlichen Gesichtssinns für das Nachtsehen.

e. Die Y(λ)-Kurve ist so normiert, dass sie bei = 550 nm exakt den Wert 1 annimmt.

f. Die Summe von X(λ), Y(λ) und Z(λ) ist immer gleich 1.

57 Pkt **Aufgabe 3**: Gegeben ist der Verlauf des Spektralfarbenzugs in der CIE-Normfarbtafel mit den x-y-Koordinaten, die in der nachfolgenden Tabelle auszugsweise eingetragen sind:

λ [nm]	x(λ)	y(λ)
440	0,16441176	0,01085756
441	0,16382843	0,01138487
442	0,1632099	0,01193739
443	0,16255214	0,01252003
444	0,16185144	0,01313731
445	0,16110458	0,01379336
446	0,1603096	0,01449138
447	0,15946595	0,01523206
448	0,15857311	0,01601516
449	0,15763117	0,01683987
450	0,15664093	0,0177048
451	0,1556051	0,01860861
452	0,15452461	0,0195557
453	0,15339723	0,02055373
454	0,15221924	0,02161171
455	0,15098541	0,02274019
456	0,14969056	0,02395033
457	0,14833682	0,0252474
458	0,14692823	0,02663519
459	0,14546837	0,02811843
460	0,1439604	0,02970297
461	0,14240509	0,03139358
462	0,14079565	0,03321315
463	0,13912068	0,03520057
464	0,13736376	0,03740309
465	0,13550267	0,03987912
466	0,13350934	0,04269239
467	0,13137064	0,04587598
468	0,12908579	0,04944981
469	0,12666216	0,05342592
470	0,12411848	0,05780251
471	0,12146858	0,06258767
472	0,11870128	0,06783044
473	0,11580736	0,07358071
474	0,11277605	0,07989582
475	0,10959432	0,08684251
476	0,10626074	0,09448607
477	0,10277586	0,10286374

Fortsetzung auf der naechsten Seite

Fortsetzung von der vorhergehenden Seite

λ [nm]	x(λ)	y(λ)
478	0,0991276	0,11200703
479	0,09530406	0,12194486
480	0,09129351	0,13270204
481	0,08708243	0,14431658
482	0,08267953	0,15686596
483	0,07811599	0,17042049
484	0,07343726	0,18503188
485	0,06870592	0,20072322
486	0,06399302	0,21746761
487	0,05931583	0,23525374
488	0,05466652	0,25409559
489	0,0500315	0,2740018
490	0,04539073	0,29497596
491	0,04075732	0,31698108
492	0,03619511	0,33989993
493	0,03175647	0,36359769
494	0,02749419	0,38792133
495	0,02345994	0,41270348
496	0,01970464	0,43775589
497	0,01626847	0,46295451
498	0,01318304	0,48820707
499	0,0104757	0,51340425
500	0,00816803	0,53842307
501	0,00628485	0,56306846
502	0,00487543	0,58711644
503	0,00398243	0,6104475
504	0,00363638	0,63301138
505	0,00385852	0,65482315
506	0,00464571	0,67589846
507	0,00601091	0,69612006
508	0,0079884	0,71534152
509	0,01060329	0,73341294
510	0,01387025	0,75018643
511	0,01776612	0,76561215
512	0,02224421	0,77962992
513	0,02727326	0,7921035
514	0,03282036	0,80292567
515	0,0388518	0,81201602
516	0,04532798	0,8193908
517	0,05217669	0,82516354

Fortsetzung auf der naechsten Seite

Fortsetzung von der vorhergehenden Seite

λ [nm]	x(λ)	y(λ)
518	0,05932553	0,82942578
519	0,06671589	0,83227374
520	0,07430242	0,83380309
521	0,0820534	0,83409031
522	0,08994174	0,83328892
523	0,09793975	0,83159267
524	0,10602111	0,82917819
525	0,11416072	0,82620696
526	0,12234737	0,8227704
527	0,13054567	0,81892785
528	0,13870235	0,81477438
529	0,14677322	0,81039461
530	0,15472206	0,80586355
531	0,16253542	0,80123848
532	0,1702372	0,79651854
533	0,17784953	0,79168658
534	0,18539076	0,78672777
535	0,1928761	0,78162922
536	0,2003088	0,77639942
537	0,20768999	0,7710548
538	0,21502955	0,7655951
539	0,2223366	0,76002
540	0,22961967	0,75432909
541	0,23688472	0,74852447
542	0,24413256	0,74261399
543	0,25136341	0,73660558
544	0,25857751	0,7305066
545	0,26577508	0,72432392
546	0,2729576	0,71806219
547	0,28012894	0,71172473
548	0,28729241	0,70531627
549	0,29445028	0,69884202
550	0,3016038	0,69230776
551	0,30875992	0,68571206
552	0,31591439	0,67906348
553	0,32306627	0,6723674
554	0,33021555	0,66562803
555	0,33736333	0,65884829
556	0,3445132	0,65202821
557	0,35166441	0,64517217

Fortsetzung auf der naechsten Seite

Fortsetzung von der vorhergehenden Seite

λ [nm]	x(λ)	y(λ)
558	0,35881369	0,63828734
559	0,36595936	0,63137908
560	0,37310154	0,62445086
561	0,38024384	0,61750215
562	0,38737898	0,6105418
563	0,39450655	0,60357134
564	0,40162592	0,59659242
565	0,40873626	0,58960687
566	0,41583577	0,58261797
567	0,42292093	0,57563069
568	0,42998863	0,56864889
569	0,43703642	0,56167577
570	0,44406246	0,5547139
571	0,45106494	0,54776604
572	0,45804067	0,54083663
573	0,46498633	0,53393005
574	0,47189874	0,52705057
575	0,47877479	0,52020231
576	0,48561159	0,51338866
577	0,49240498	0,50661492
578	0,49915067	0,49988734
579	0,50584528	0,49321118
580	0,51248637	0,48659079
581	0,51907251	0,48002861
582	0,52560049	0,47352737
583	0,5320656	0,46709136
584	0,53846276	0,46072525
585	0,54478651	0,45443411
586	0,55103105	0,4482245
587	0,55719291	0,44209914
588	0,56326931	0,43605806
589	0,56925682	0,43010197
590	0,57515131	0,42423223
591	0,58095261	0,41844688
592	0,58665019	0,41275842
593	0,5922248	0,40718953
594	0,59765816	0,40176193
595	0,60293279	0,39649663
596	0,60803511	0,39140915
597	0,612977	0,38648616

Fortsetzung auf der naechsten Seite

Fortsetzung von der vorhergehenden Seite		
λ [nm]	x(λ)	y(λ)
598	0,61777873	0,38170576
599	0,6224593	0,37704729
600	0,6270366	0,37249115
601	0,63152094	0,36802601
602	0,63589982	0,3636654
603	0,64015616	0,35942772
604	0,64427296	0,35533137
605	0,64823311	0,35139492
606	0,65202824	0,34762796
607	0,65566918	0,34401829
608	0,65916613	0,34055323
609	0,66252822	0,33722099
610	0,66576358	0,33401065
611	0,66887414	0,33091855
612	0,67185867	0,32794707
613	0,67471951	0,32509518
614	0,67745889	0,32236208
615	0,68007885	0,31974722
616	0,68258157	0,31724871
617	0,6849706	0,31486282
618	0,68725045	0,31258596
619	0,6894263	0,31041401
620	0,69150397	0,30834226

Nachfolgend soll die farbtongleiche Wellenlänge der EBU- und SMPTE-Phosphore geschätzt bzw. berechnet werden.

12 Pkt

a) Geben Sie zunächst einen 10 nm breiten Bereich für die farbtongleiche Wellenlänge der EBU- und SMPTE-Phosphore an, in dem Sie diese mit Hilfe der CIE-Normfarbtafel **Abb. 1.3-5** grafisch konstruieren.

EBU-Phosphore:
R_{min}:
R_{max}:
G_{min}:
G_{max}:
B_{min}:
B_{max}:

SMPTE-Phosphore:
R_{min}:
R_{max}:
G_{min}:

G_{max}:
B_{min}:
B_{max}:

b) Errechnen Sie nun auch nm genau die λ-Werte für das EBU-Farbdreiecks, in dem Sie die o. a. Tabelle verwenden und für die Koordinaten des EBU-Farbdreiecks die **Gl.1.3-3** benutzen. **45 Pkt**

R:
G:
B:

Aufgabe 4: Geben Sie den Signal-Störabstand für eine 8-Bit und 10-Bit Quantisierung an! **10 Pkt**

Aufgabe 5: **12 Pkt**

a) Geben Sie die Anzahl der Abtastwerte für die Luminanz- und Chrominanzkomponenten für die aktive Zeile nach ITU-R BT.601 an! **4 Pkt**

a: 640
b: 704
c: 720
d: 768

b) Wie viel Abtastwerte wären eigentlich notwendig, um eine 4:3-konforme Darstellung mit quadratischen Pixeln zu erreichen? **8 Pkt**

Übungsaufgaben zu Kapitel "Digitale Bilddarstellung und -übertragung"

Aufgabe 6: Errechnen Sie den Pixeltakt in MHz für ein hoch aufgelöstes digitales Bildsignal mit den Parametern: 50 Hz, Zeilensprung, 2640 Abtastwerte (nominell), 1125 Zeilen (nominell)! **5 Pkt**

Aufgabe 7: Errechnen Sie die notwendige Kompressionsrate (aufrunden) für ein reines SMPTE 296M Videosignal (HD-1080i 25 Hz, 10 Bit Quantisierung), das über ein 10 Mbit/s Datenkanal übertragen werden soll! **15 Pkt**

Hinweis: Berücksichtigen Sie nur den aktiven Bildinhalt ohne Audio etc.! Beachten Sie die Umrechnung: 1 Mbit = 1024 kbit, 1 kbit = 1024 Bit.

20 Pkt **Aufgabe 8**: Gegeben sei die Matrix $\underline{A}$:

$$\underline{A} = \begin{pmatrix} 1 & 0 & -2 \\ 4 & 1 & 0 \\ 1 & 1 & 7 \end{pmatrix}$$

Es soll überprüft werden, ob $\underline{A}$ orthogonal und sich damit als Transformationsmatrix für eine Transformationscodierung gemäß **Gl. 3.1-2**

$$\underline{A}^{-1} = \underline{A}^{T} \tag{12}$$

eignet.

5 Pkt a) Geben Sie zunächst die Transformierte von $\underline{A}$ an:

a.

$$\underline{A}^{T} = \begin{pmatrix} 1 & 1 & 7 \\ 4 & 1 & 0 \\ 1 & 0 & -2 \end{pmatrix} \tag{13}$$

b.

$$\underline{A}^{T} = \begin{pmatrix} 7 & 0 & -2 \\ 1 & 1 & 0 \\ 1 & 4 & 1 \end{pmatrix} \tag{14}$$

c.

$$\underline{A}^{T} = \begin{pmatrix} -2 & 0 & 1 \\ 0 & 1 & 4 \\ 7 & 1 & 1 \end{pmatrix} \tag{15}$$

d.

$$\underline{A}^{T} = \begin{pmatrix} 1 & 4 & 1 \\ 0 & 1 & 1 \\ -2 & 0 & 7 \end{pmatrix} \tag{16}$$

e.

$$\underline{A}^{T} = \begin{pmatrix} 7 & -2 & 2 \\ 1 & 9 & 0 \\ 3 & 4 & 1 \end{pmatrix} \tag{17}$$

b) Berechnen Sie die Inverse zu A, in dem Sie die fehlenden Koeffizienten angeben: **15 Pkt**

$$\underline{A} = \begin{pmatrix} 1 & 0 & -2 \\ 4 & 1 & 0 \\ 1 & 1 & 7 \end{pmatrix} \qquad 18$$

a. =

b. =

c. =

Aufgabe 9: Für den 3 x 3 Bildblock B **10 Pkt**

$$B(x,y) = \begin{pmatrix} 126 \\ 115 \\ 110 \\ 80 \\ c \quad 75 \\ 66 \\ 12 \\ 13 \\ 15 \end{pmatrix} \qquad 23$$

soll der DC-Koeffizient der diskreten Cosinus Transformation errechnet werden.

Hinweis: Geben Sie zunächst die formale Transformationsvorschrift für die diskrete Cosinus Transformation (DCT) für diesen Fall M = N = 3 an!

Aufgabe 10: Gegeben seien die digitalen Werte von 8 Bildpunkten. Es soll die 1D-DCT, die Werte nach der Quantisierung und Rekonstruktion nach der 1D-IDCT errechnet werden. **20 Pkt**

In der nachfolgenden Tabelle sind die fehlenden Werte zu ergänzen.

100	100	100	100	100	100	100	100	digitale Werte der 8 Bildpunkte
-28	-28	-28	-28	-28	-28	-28	-28	nominiert auf Werte von -128 bis +127
a	0	0	0	0	0	0	0	ermittelt DCT-Koeffizienten (gerundet)
16	11	10	16	24	40	51	61	Quantisierungsmatrix
b	*c*	0	0	0	0	0	0	nach Anwendung der Quantisierungsmatrix (gerundet)
d	100	*e*	100	100	100	100	100	reproduzierte digitale Werte nach IDCT

Hinweis: Verwenden Sie die Prozedur, die in **Abschnitt 3.2.4** beschrieben ist.

a.

b.

c.

d.

e.

30 Pkt **Aufgabe 11**: Der folgende DCT-Block sei gegeben:

20	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

10 Pkt a) Ermitteln Sie für diesen DCT-Block die RLC-Darstellung ermittelt werden!

a. (20) (1,–2) (0,1) EOB
b. (20) (0,2) (–2) (1) EOB
c. (20) (2,–2) (0,1) EOB
d. (20) (–2,0) (1,0) EOB
e. (20,1) (2,–2) (0,1) EOB

b) Unter Zuhilfenahme der zugehörigen Tabellen (**Tabelle 3.3-1**, **Tabelle 3.3-2** und **Tabelle 3.3-3**) soll aus der RLC für das Beispiel aus **a)** die VLC in Bitdarstellung errechnet werden, die sich für das JPEG-Verfahren ergibt. **20 Pkt**

Geben Sie unter Zuhilfenahme der zugehörigen Tabellen den Ergebniscode (Bitdarstellung) an, der sich mit der VLC für das JPEG-Verfahren ergibt!

Hinweis: Erstellen Sie zunächst für dieses Beispiel eine VLC-Tabelle (vgl. **Tabelle 3.3-4**).

a. 110101001111001010011010
b. 110101001011001010011010
c. 110101001111011010011010
d. 110101001111000101011010
e. 111101001111001010011010

Übungsaufgaben zu Kapitel "Digitale Quellencodierung für Einzelbilder"

Aufgabe 12: Gegeben sei die eindimensionale Wertefolge: **20 Pkt**

64 76 84 90 94 98 98 98 44 12 12 18 24 20 26 24

Errechnen Sie die Mittelwertsignal-Folge und die Differenzsignal-Folge mit der Haar-Transformation! Geben Sie die fehlenden Koeffizienten an!

Mittelwertsignal-Folge: *a b c* 98 28 15 *d e*

a.
b.
c.
d.
e.

Differenzsignal-Folge: *f g h* 0 16 *i* 2 *j*

f.
g.
h.
i.
j.

Aufgabe 13: Gegeben sei der 4 x 4 Bildpunkte große Bildausschnitt: **30 Pkt**

$$B(x,y) = \begin{pmatrix} 5 & 2 & 2 & 4 \\ 0 & 0 & 1 & 3 \\ 3 & 4 & 5 & 6 \\ 2 & 3 & 3 & 2 \end{pmatrix} \tag{31}$$

Für diesen Block soll mit Hilfe der Haar-Scaling Funktion und des Haar-Wavelets eine Stufe der 2D-Wavelet-Transformation durchgeführt werden.

Gehen Sie nach dem Prinzip vor, das in **Abb. 3.4-3** skizziert ist! Berechnen Sie zunächst spaltenweise die Anteile T und H und ermitteln dann die Anteile TT, HT, TH und HH. Ergänzen Sie dann die Koeffizienten der nachfolgenden Matrix:

$$B_{Haar}(u,v) = \frac{1}{q} \begin{pmatrix} a' & 10 & 3 & d' \\ 12 & f' & -2 & 0 \\ 7 & 2 & k' & 0 \\ 2 & n' & 0 & -2 \end{pmatrix} \tag{32}$$

26 Pkt **Aufgabe 14**: Gegeben sei der originale Bildblock *s(x,y)*:

$$s(x,y) = \begin{pmatrix} 160 & 159 & 160 & 159 & 158 & 158 & 159 & 159 \\ 161 & 160 & 160 & 159 & 156 & 156 & 153 & 153 \\ 163 & 160 & 160 & 153 & 155 & 155 & 156 & 153 \\ 159 & 159 & 155 & 156 & 158 & 160 & 160 & 158 \\ 160 & 159 & 156 & 158 & 159 & 159 & 159 & 159 \\ 159 & 156 & 158 & 158 & 158 & 158 & 158 & 158 \\ 158 & 155 & 158 & 156 & 156 & 155 & 155 & 156 \\ 158 & 155 & 156 & 155 & 155 & 154 & 154 & 155 \end{pmatrix}$$

Das codierte Ergebnisbild dieses Bildblocks sei *s'(x,y)*:

$$s'(x,y) = \begin{pmatrix} 160 & 159 & 160 & 159 & 158 & 158 & 159 & 159 \\ 161 & 160 & 160 & 159 & 158 & 157 & 155 & 153 \\ 163 & 162 & 161 & 155 & 155 & 155 & 156 & 154 \\ 159 & 159 & 156 & 156 & 158 & 160 & 160 & 158 \\ 160 & 159 & 158 & 158 & 159 & 159 & 159 & 159 \\ 159 & 156 & 158 & 158 & 158 & 158 & 158 & 158 \\ 158 & 157 & 158 & 157 & 156 & 155 & 155 & 156 \\ 158 & 156 & 156 & 155 & 155 & 154 & 154 & 155 \end{pmatrix}$$

16 Pkt a) Berechnen Sie den mittleren quadratischen Fehler bezogen auf den Bildblock.

MSE(s,s') =

b) Berechnen Sie als Maß der Codiereffizienz den Peak-SNR in dB bezogen auf den Beispielblock! **10 Pkt**

PSNR(s,s') =

Aufgabe 15: Gegeben sei der Eingangsdatenstrom *s(n)* für eine prädiktive Bildcodierung (D*PCM und DPCM). Als Prädiktor werde die einfache Wertewiederholung genommen. Die verwendete Quantisierung rundet auf die nächst kleineren geraden Zahlwert ab. Zu Beginn seien alle Zustände gleich 0. Der Übertragungskanal soll das Signal fehlerfrei übertragen. **24 Pkt**

$$s(n) = (7, 9, 10, 8, 8, 9, 7, 6, 5, 9, 0, 0, 0, 9, 8, 7, 6, 5, 4, 7, 8, 9, 9, 8, 7) \quad 47$$

a) Das Signal werde mit einer Feed-Forward DPCM (D*PCM) codiert und übertragen. Berechnen Sie die Wertefolge am Ausgang des Decoders $s_d(n)$ für das o. a. Eingangssignal! **12 Pkt**

Ergänzen Sie die fehlenden Werte:

$$s_d(n) = (6, 2, 6, 0, 4, 0, 4, 4, -4, 4, -4, 12, -4, 12, a, b, c, d, -4, 14, -4, 14, -4, 14, 4) \quad 48$$

b) Das aus **a)** bekannte Eingangssignal werde nun mit einer Feed-Backward DPCM codiert und übertragen. Berechnen Sie die Wertefolge am Ausgang des Decoders $s_d(n)$. **12 Pkt**

Ergänzen Sie die fehlenden Werte:

$$s_d(n) = (6, 8, 10, 8, 8, a, b, c, d, 8, 0, 0, 0, 8, 8, 8, 6, 8, 4, 6, 8, 8, 8, 8) \quad 49$$

Übungsaufgaben zu Kapitel "Prinzipien der digitalen Quellencodierung für Bewegtbilder"

Aufgabe 16: Anhand eines Beispiels soll der 3-schrittige 2D Logarithmic Search (TDL) durchgeführt werden (Startsuchweite: 4 Bildpunkte). Dazu ist der 3 x 3 Bildpunkte große Referenzblock **82 Pkt**

$$B(x,y) = \begin{pmatrix} 20 & 21 & 20 \\ 49 & 52 & 47 \\ 99 & 98 & 102 \end{pmatrix}$$

sowie ein Suchbereich *S(x,y)*, $x = 0 \ldots 10$, $y = 0 \ldots 13$ gegeben:

y \ x	0	1	2	3	4	5	6	7	8	9	10
0	34	35	20	34	40	34	45	47	48	50	44
1	62	57	54	30	23	21	21	20	21	18	15
2	59	55	57	51	50	49	52	47	60	62	65
3	97	99	98	102	97	99	98	102	102	105	117
4	102	103	105	65	49	60	48	76	126	127	129
5	111	107	107	109	107	109	110	115	117	119	121
6	97	88	87	85	85	90	97	86	86	80	79
7	70	73	70	65	67	68	70	71	70	74	78
8	50	54	52	53	56	56	55	52	51	49	49
9	34	32	29	20	25	23	26	27	23	20	21
10	49	50	51	49	48	47	49	50	52	53	51
11	80	81	83	84	90	93	95	98	99	100	102
12	85	89	90	101	102	100	99	98	97	95	99
13	87	86	85	90	97	87	90	93	95	97	99

Die Nullverschiebung liegt bei $x = 5$ und $y = 8$ (rot gekennzeichnet).

4 Pkt a) Für wieviele unterschiedliche Verschiebungen muss ein Match für den Referenzblock *B(x,y)* Suchbereich *S(x,y)* nach dem o. g. Suchalgorithmus mit Hilfe der MSE insgesamt berechnet werden?

N_{TDL} =

5 Pkt b) Wieviele MSE-Berechnungen sind im ersten Suchschritt nach dem o. g. Verfahren notwendig?

$N_{1,TDL}$ =

10 Pkt c) Geben Sie dx_1, dy_1 für die minimale MSE_1(dx_1, dy_1) des ersten Suchschrittes an!

dx_1 =
dy_1 =

7 Pkt d) Wie groß ist $MSE_1(dx_1, dy_1)$? Runden Sie auf den nächstliegenden ganzahligen Wert!

5 Pkt e) Wieviele MSE-Berechnungen sind im zweiten Suchschritt nach dem o. g. Verfahren notwendig?

10 Pkt f) Geben Sie dx_2, dy_2 für die minimale $MSE_2(dx_2,dy_2)$ des zweiten Suchschrittes an!

7 Pkt g) Wie groß ist $MSE_2(dx_2,dy_2)$? Runden Sie auf den nächstliegenden ganzahligen Wert!

h) Wieviele MSE-Berechnungen sind im dritten Suchschritt nach dem o. g. Verfahren notwendig? **5 Pkt**

i) Geben Sie dx_3, dy_3 für die minimale $MSE_3(dx_3, dy_3)$ des zweiten Suchschrittes an! **10 Pkt**

j) Wie groß ist $MSE_3(dx_3, dy_3)$? Runden Sie auf den nächstliegenden ganzahligen Wert! **7 Pkt**

k) Nachfolgend sei der Suchbereich des Bildes leicht modifiziert. Die Zeile 4 **12 Pkt**

$$z(4) = (102, 103, 105, 65, 49, 60, 48, 76, 126, 127, 129) \quad 52$$

werde durch die Zeile

$$z_{mod}(4) = (102, 103, 105, 102, 115, 120, 121, 122, 126, 129) \quad 53$$

ersetzt. Nachfolgend wird die Frage gestellt, ob die bekannten Suchverfahren das Minimum bei der gleichen *dx-dy*-Verschiebung finden, wie es in Aufgabe 4.1.10 ermittelt wurde. Welche der nachfolgenden Aussagen sind richtig?

a. Das o. g. TDL-Verfahren findet ein Minimum bei der gleichen *dx-dy*-Verschiebung.
b. Das TSS-Verfahren findet ein Minimum bei der gleichen *dx-dy*-Verschiebung.
c. Das CSA-Verfahren findet ein Minimum bei der gleichen *dx-dy*-Verschiebung.
d. Der Full-Search Algorithmus findet ein Minimum bei der gleichen *dx-dy*-Verschiebung.

Aufgabe 17: Diese Aufgabe beschäftigt sich mit den durchschnittlich erzielbaren Kompressionsraten bei der MPEG-Codierung in Abhängigkeit der GOP-Länge. Beispielhaft soll eine normale (nicht *closed*) 6er-GOP mit der Bildtypfolge I-B-B-P-B-B mit einer 13er-GOP, die die Bildtypfolge I B B B P B B B P B B B P besitzt, verglichen werden. **21 Pkt**

Das Ursprungsmaterial besitze 720 x 576 Bildpunkte und eine Farbunterabtastung von 4:2:2. Das I-Bild sei durchschnittlich 128 kByte groß (1 kByte = 1024 Bit) und es gelten die bei TM5 (vgl. **Abschnitt 5.2.7**, **Gl. 5.2-5**) gemachten Annahmen bezüglich des Datenaufkommens der Bildtypen I, P und B.

a) Geben Sie das Datenaufkommen D_u (in kByte) für ein Bild des Ursprungsmaterial an! **3 Pkt**

b) Geben Sie das durchschnittliche Datenaufkommen D (in kByte) der Bildtypen P und B für dieses Beispiel an! Runden Sie auf den nächstliegenden ganzahligen Wert! **6 Pkt**

12 Pkt c) Geben Sie die durchschnittlich erzielbaren Kompressionsraten für die o. a. 6er-GOP und die 13er-GOP an. Runden Sie auf den nächstliegenden ganzahligen Wert!

Übungsaufgaben zu Kapitel "Standards der digitalen Videocodierung"

6 Pkt **Aufgabe 18**: Geben Sie die Kompressionsrate *R* für den reinen aktiven 50i-Videodatenstrom gemäß MPEG-2 MP@HL an, wenn für Audio und Protokoll Overhead 10 % der verfügbaren Datenrate angesetzt wird! Runden Sie auf den nächstliegenden ganzahligen Wert!

Hinweis: 1024 kbit = 1 Mbit, 1024 Bit = 1 kbit

65 Pkt **Aufgabe 19**: In einer Zeilensprungbildsequenz bewege sich eine vertikale Kante horizontal von links nach rechts. Diese Sequenz soll mit dem MPEG-2 Verfahren codiert werden. Für einen Beispielmakroblock (I-Bild) ist ein einzelner quantisierter DCT-Block zum Vergleich einmal als *Field* und einmal als *Frame* berechnet worden. An diesem Beispiel soll ermittelt werden, ob eine von den beiden Codiermethoden bessere Kompressionsraten liefert.

226	48	–6	–9	4	2	–2	–1	267	29	–8	–1	0	3	–2	0
0	0	0	0	0	0	0	0	–4	3	0	–1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	–3	3	1	–1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	–4	3	0	–2	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	–11	10	0	–4	2	0	0	0

Quantisierter DCT-Block: *Field* (links), *Frame* (rechts)

5 Pkt a) Betrachten Sie die beiden DCT-Blöcke und überschlagen das Ergebnis. Welche der nachfolgenden Aussagen sind richtig:

a. Der mit der Frame-Codierung ermittelte DCT-Block benötigt ein geringeres Datenaufkommen als der mit der Field-Codierung.

b. Der mit der Field-Codierung ermittelte DCT-Block benötigt ein geringeres Datenaufkommen als der mit der Frame-Codierung.

c. Beide Codiermethoden benötigen das gleiche Datenaufkommen.

d. Beide Codiermethoden benötigen unterschiedliche Datenaufkommen.

e. Die beiden DCT-Blöcke repräsentieren den gleichen Bildinhalt.

b) Berechnen Sie das Datenaufkommen (in Bit) des mit der Field-Codierung ermittelten DCT-Blocks! Verwenden Sie dafür das für diese Codierung übliche Verfahren bei der Zusammenfassung! **10 Pkt**

c) Berechnen Sie das Datenaufkommen (in Bit) des mit der Field-Codierung ermittelten DCT-Blocks! Verwenden Sie dafür das für die Frame-Codierung übliche Verfahren bei der Zusammenfassung! **10 Pkt**

d) Berechnen Sie das Datenaufkommen (in Bit) des mit der Frame-Codierung ermittelten DCT-Blocks! Verwenden Sie dafür das für diese Codierung übliche Verfahren bei der Zusammenfassung! **20 Pkt**

e) Berechnen Sie das Datenaufkommen (in Bit) des mit der Frame-Codierung ermittelten DCT-Blocks! Verwenden Sie dafür das für die Field-Codierung übliche Verfahren bei der Zusammenfassung! **20 Pkt**

Aufgabe 20: Berechnen Sie die mittlere Kompressionsrate, die ein Videoencoder (625/50-Zeilen System) selbst erreichen muss, um einen *S-VCD* Videodatenstrom mit 2520 kbit/s zu erzeugen! Runden Sie auf den nächstliegenden ganzahligen Wert! **13 Pkt**

Hinweis: Berechnen Sie zunächst die Eingangsdatenrate für das *S-VCD* Videoformat! 1024 Bit = 1 kbit; 1024 kbit = 1 Mbit

Aufgabe 21: Berechnen Sie die maximal mögliche Wiedergabezeit für ein *S-VCD* konformes Video (maximale Datenrate, ohne Audiodaten), das auf ein 4.7 GByte großes *DVD-Medium* geschrieben wird! Runden Sie auf den nächstliegenden ganzahligen Wert! **13 Pkt**

Übungsaufgaben zu Kapitel "Videostandards in Anwendungen"

Aufgabe 22: Gesucht sind die ersten 10 Bytes (hexadezimal) des Signals, welches das rückgekoppelte Zufallszahlengenerator der Energieverwischung alleine nach einer Initialisierung erzeugt! Beachten Sie, dass das *MSB (most significant Bit)* zuerst gebildet wird und das erste Symbol direkt nach der Initialisierung mitgerechnet werden soll! **25 Pkt**

Ergänzen Sie die fehlenden Symbole! Geben Sie das Ergebnis hexadezimal an!

03_H, $F6_H$, **a**, **b**, **c**, $B8_H$, $A3_H$, **d**, $C9_H$, **e**

a. =

b. =

c. =

d. =

e. =

30 Pkt **Aufgabe 23**: Gegeben sei ein Faltungsinterleaver nach Forney mit der Basisverzögerung M = 2 und einer Interleavingtiefe I = 4.

5 Pkt a) Wie groß ist die maximale Symbolverschiebung V_S am Ausgang des Interleavers?

a.

$$V_S = I^2 \cdot (M-1) \quad 64$$

b.

$$V_S = I \cdot (M-1) \cdot M \quad 65$$

c.

$$V_S = I \cdot (I-1) \cdot M \quad 66$$

d.

$$V_S = I \cdot M \quad 67$$

e.

$$V_S = I \cdot M^2 - 1 \quad 68$$

25 Pkt b) An den o. g. Faltungsinterleaver nach Forney werde als Eingangssignal die einfach ansteigende Wertefolge *I(n)*

I(n) = {1, 2, 3, 4, 5, 6 ... }

angelegt. Gesucht ist die Wertefolge *O(n)*, welche der Faltungsinterleaver am Ausgang erzeugt für einen Wertebereich von

n = 1 ... 40.

Ergänzen Sie die fehlenden Werte!

O(n) = {1, 0, 0, 0, 5, 0, 0, 0, 9, 2, 0, 0, ***a***, 6, 0, 0, 17, 10, 3, 0, 21, 14, 7, 0, 25, 18, 11, **b**, **c**, 22, 15, 8, 33, 26, 19, 12, 37, 30, **d**, **e**}

a =

b =
c =
d =
e =

Aufgabe 24: Gegeben sei der Faltungscoder nach dem DVB-Standard. Am Eingang werde folgender Datenstrom I angelegt (hexadezimale Schreibweise, MSB zuerst): **40 Pkt**

$I = \{F2_H, D6_H, 2C_H\}$.

Das Schieberegister des Faltungscoders sei mit Nullen initialisiert.

a) Errechnen Sie die Ausgangsdatenströme x und y! **20 Pkt**

Hinweise: Rechnen Sie *I* zunächst in Bit-Schreibweise um. Beachten Sie, dass das MSB des ersten Wertes zuerst in den Faltungscoder gelangt. Fassen Sie jeweils 8 Bit am Ausgang *x* und *y* zu einem Symbol zusammen, dass Sie in Dezimalschreibweise angeben! Ergänzen Sie die fehlenden Zahlen!

$x = \{a, 153, b\}, y = \{246, c, d\}$

a =
b =
c =
d =

b) Gegeben sei der in **b)** berechnete Ausschnitt von 2 mal 5 Bytes der Datenfolge am Ausgang des o. a. Faltungscodes. Errechnen Sie das Ergebnis x' und y', welches sich nach einer 3/4 Punktierung aus diesen Daten ergibt. Geben Sie die fehlenden Werte als Dezimalzahlen an! **20 Pkt**

Hinweise: Beachten Sie auch hier, dass die MSBs von x und y zuerst der Punktierung zugeführt werden! Berechnen Sie *x*' und *y*' zunächst in hexadezimaler Schreibweise!

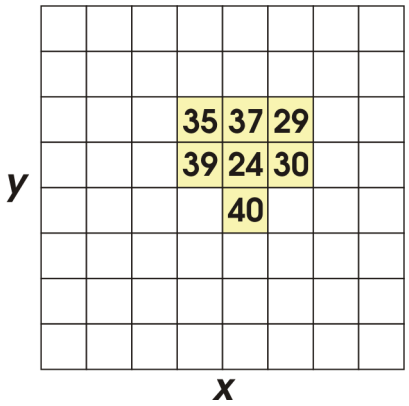
$x' = \{a, b\}, y' = \{c, d\}$

a =
b =
c =
d =

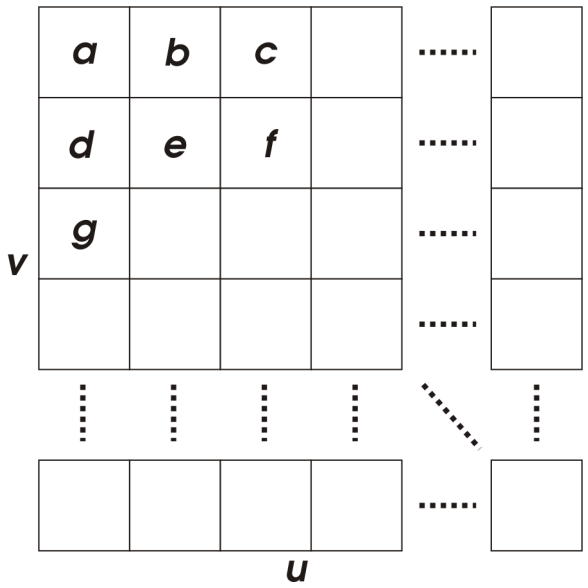
Aufgabe 25: Welche Bandbreitenausnutzung B (in [Bit/sek * Hz], eine Nachkommastelle genau) ergibt sich bei einer DVB-C Übertragungskette, wenn eine *256-QAM* verwendet wird? Berücksichtigen Sie auch den Einfluss der bei DVB-C eingesetzten Kanalcodierung! **5 Pkt**

Übungsaufgaben zu Kapitel "Verbesserte Codiersysteme für die Bildkommunikation"

35 Pkt **Aufgabe 26**: Gegeben sei der nachfolgende 8 x 8 Luminanzblock eines Videobildes, welches in MPEG-4 Video Intraframe-codiert werden soll. In den Block eingetragen ist eine kleine Objektkontur mit den zugehörigen Grauwerten (*Y*-Werte) für die entsprechenden Bildpunkte.



Führen Sie eine SA-DCT für diesen Block durch und geben Sie die Koeffizienten des Ergebnisblockes an! Geben Sie die fehlenden Werte auf eine Nachkommastelle genau an!



a =

b =

c =

d =
e =
f =
g =

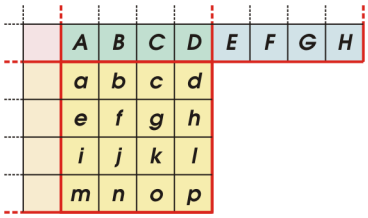
Aufgabe 27: Ein 4 x 4 Bildpunkte großer Luminanzblock (Bildpunkte *a* ... *p*) soll mit Hilfe von benachbarten Blöcken und den dort bekannten Werten mit dem 4 x 4 Intra-Prädiktions-*Mode* 3 von H.264/AVC prädiziert werden (vgl. Bild). **16 Pkt**

Folgende Werte seien gegeben, bzw. bereits errechnet:

A = 24, *B* = 26, *C* = 30, *D* = 34, *E* = 34, *F* = 32, *G* = 30, *H* = 28

a = 27, *b* = 30.5, *d* = 34, *f* = 33.5, *g* = 34, *i* = 33.5, *k* = 32.5, *o* = 30.5

Geben Sie die fehlenden 5 Werte des gesuchten Bildblockes mit einer Nachkommastelle Genauigkeit an!



c =
e =
h =
j =
l =
m =
n =
p =

Aufgabe 28: Für den Bildblock $\underline{X}$, mit **35 Pkt**

$$\underline{X} = \begin{pmatrix} 5 & 11 & 8 & 10 \\ 9 & 8 & 4 & 12 \\ 1 & 10 & 11 & 4 \\ 19 & 6 & 15 & 7 \end{pmatrix}$$

97

soll eine Transformation mit der klassischen DCT ($\underline{Y}_{DCT}$) und vergleichend mit der *Integer-Transformation* ($\underline{Y}_{INT}$) berechnet werden!

18 Pkt a) Errechnen Sie die klassischen DCT (YDCT)! Runden Sie im Ergebnis alle Zahlen auf 2 Nachkommastellen genau! Geben Sie die fehlenden Koeffizienten an!

Hinweis: Errechnen Sie zunächst für den Bildblock $\underline{X}$ die *core-Transformationsmatrizen* $\underline{W}$ für die DCT-Transformation aus!

$$\underline{Y}_{DCT} = \begin{pmatrix} a & -0.08 & -1.50 & 1.12 \\ -3.3 & b & c & -9.01 \\ 5.50 & 3.03 & 2 & 4.7 \\ -4.05 & -3.01 & -9.38 & -1.23 \end{pmatrix}$$

a =
b =
c =

17 Pkt b) Errechnen Sie nun die Integer-Transformation ($\underline{Y}_{Int}$)! Runden Sie auch hier im Ergebnis alle Zahlen auf 2 Nachkommastellen genau! Geben Sie die fehlenden Koeffizienten an!

Hinweis: Errechnen Sie zunächst für den Bildblock $\underline{X}$ die *core-*Transformationsmatrizen $\underline{W}$' für die Integer-Transformation aus!

$$\underline{Y}_{INT} = \begin{pmatrix} a & -0.16 & -1.5 & 1.11 \\ -3 & b & c & -9.2 \\ 5.5 & 2.69 & 2 & 4.90 \\ -4.27 & -3.2 & -9.33 & -2.1 \end{pmatrix}$$

a =
b =
c =

Lösungen zu Übungsaufgaben

Lösung zu Aufgabe 1:

a) 72 Hz

b) 13.89 ms

c) Die Geschwindigkeit des Objekts ist geben bezogen auf eine Sekunde. Mit Hilfe eines Dreisatzes wird die Geschwindigkeit des Objektes auf die Zeitspanne *T* umgerechnet:

$$\frac{\frac{1}{2}}{1\ s} = \frac{x}{\frac{1}{72}s} \qquad 1$$

Umgeformt: $x = \frac{1}{2s} \cdot \frac{1}{72}s = \frac{1}{144} \hat{=} 0.69444\%$

d) 1.39

Die Darstellung weicht bei einer 3-fach wiederholten Darstellung bei der zweifachen Zeit von *T* am meisten ab. Vergleichen Sie auch **Abb. 1.2-9** im Skript.

e) Die horizontale Bildbreite beträgt 1920 Bildpunkte. Die Anzahl der Bildpunkte für die maximale Darstellungsdifferenz errechnet sich aus Aufgabenteil **c)** und **d)** wie folgt:

$$x = 2 \cdot \frac{1}{144} \cdot 1920\ pel = 26.66667 \qquad 2$$

Aufgerundet ergibt sich 27.

Lösung zu Aufgabe 2:

Lösung: b
Vergleichen Sie die Hinweise im **Abschnitt 1.3** sowie das **Abb. 1.3-4**.

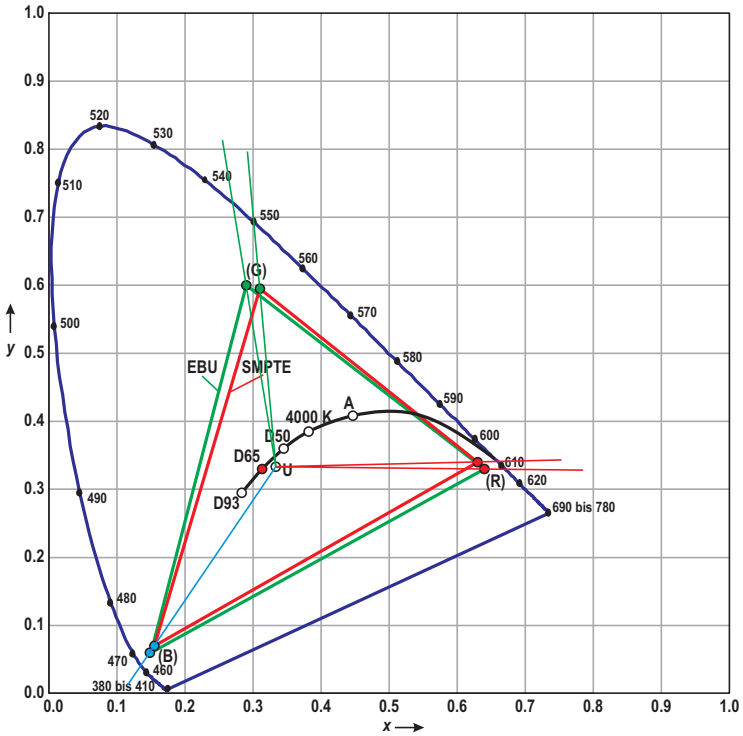
Lösung zu Aufgabe 3:

a) Lösung: EBU-Phosphore
R_{min}: 610
R_{max}: 620
G_{min}: 540
G_{max}: 550
B_{min}: 460
B_{max}: 470

Lösung: SMPTE-Phosphore:

R_{min}: 600
R_{max}: 610
G_{min}: 550
G_{max}: 560
B_{min}: 460
B_{max}: 470

Es wird der jeweilige Schnittpunkt der Gerade mit dem Spektralfarbenzug gesucht, die durch den theoretischen Unbuntpunkt U(x = y = 1/3) und die Eckpunkte der EBU- und SMPTE-Farbdreiecke gebildet wird (vgl. nachfolgende Abbildung).



b) Lösung:
R: 611 nm (611.0 / 611.0)
G: 545 nm (545.0 / 545.0)
B: 465 nm (465.0 / 465.0)

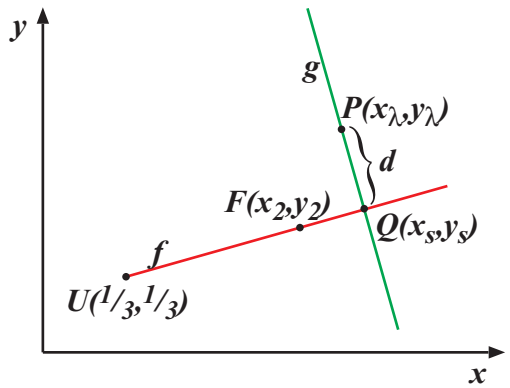
Zunächst wird eine Gerade f(x) definiert, welche durch den Unbuntpunkt $U(x_1, y_1)$ mit x_1 = y_1 = 1/3 und dem jeweiligen Eckpunkt des Farbdreiecks $F(x_1, y_2)$ festgelegt ist:

$$f(x) = a + bx \qquad 3$$

Die Konstanten a und b lassen sich durch Einsetzen der beiden Punkte U und F in die Geradengleichung bestimmen:

$a = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}$, $b = \frac{y_2 - y_1}{x_2 - x_1}$

Gesucht ist nun der Punkt P(x ,y) aus der o. a. Tabelle, der möglichst nahe an der Gerade f liegt. Dazu ist die minimale Distanz des Punktes P zur Geraden f zu berechnen. Praktisch geschieht das, in dem man zunächst eine zu f rechtwinklige Gerade g mit $g(x) = c + dx$ konstruiert. Der Schnittpunkt der beiden Geraden sei Q(xs,ys). Die Distanz d zwischen P und Q errechnet sich entsprechend der euklidischen Metrik zu $d = \sqrt{(y_s - y_\lambda)^2 + (x_s - x_\lambda)^2}$ und stellt die Zielfunktion dar, die es zu minimieren gilt. Das nachfolgende Bild veranschaulicht den Zusammenhang.



Als Ansatz für den Punkt P nimmt man einige Werte des Intervalls, die in **a)** berechnet wurden. Die Werte x_2 und y_2 lassen sich für R, G und B aus der **Gl.1.3-3** entnehmen.

1. R:x_2 = 0.640, y2 = 0.330. Berechnung einiger Werte d für R im Intervall zwischen 610 und 620 nm:

d610 = 0.00422904
d611 = 0.00123233
d612 = 0.00170654

d613 = 0.00452717

Das Minimum liegt bei **611 nm**.

2. G:x_2 = 0.290, y_2 = 0.600. Berechnung einiger Werte d für R im Intervall zwischen 540 und 550 nm:

d540 = 0.03484479
d541 = 0.02860484
d542 = 0.02239886
d543 = 0.01622525
d544 = 0.01008291
d545 = 0.00397021
d546 = 0.02239886
d547 = 0.02114964
d548 = 0.01421978

Das Minimum liegt bei **545 nm**.

3. B:x_2 = 0.150, y_2 = 0.060. Berechnung einiger Werte d für R im Intervall zwischen 460 und 470 nm:

d460 = 0.01186069
d461 = 0.00962795
d462 = 0.00727711
d463 = 0.00477900
d464 = 0.00209302
d465 = 0.00083183
d466 = 0.00405436
d467 = 0.00760390
d468 = 0.01149219

Das Minimum liegt bei **465 nm**.

Lösung zu Aufgabe 4:

Mit **Gl. 2.1-2** lässt sich der Signal-Störabstand für N = 8 und N = 10 leicht berechnen:

$$S_{Q,N=8} \approx (6 \cdot 8 + 10.8)\ dB = 58.8\ dB \quad (4)$$

$$S_Q \approx (6 \cdot N + 10.8)\ dB \quad (5)$$

$$S_{Q,N=10} \approx (6 \cdot 10 + 10.8)\ dB = 70.8\ dB \quad (6)$$

Lösung zu Aufgabe 5:

a) Lösung: b
Beim ITU-R BT.601-Standard sind horizontal 720 aktive Pixel definiert. Vergleichen Sie dazu **Tabelle 2.2-1**.

b) Bei 576 aktiven Zeilen und einem Bildseitenverhältnis von 4:3 ergeben sich daraus nicht quadratische Pixel. Um dieses dennoch zu erreichen müssten

$$n_{square} = 576Pixel \cdot {}^{4}/_{3} = 768Pixel$$

angesetzt werden (**Gl. 2.2-8**).

Lösung zu Aufgabe 6:

Der Pixeltakt errechnet sich aus der Anzahl der Abtastwerte je Vollbild, multipliziert mit der Vollbildfrequenz:

$f_{\perp} = \frac{50Hz}{2} \cdot 2640 \cdot 1125 = 74.25\ MHz.$

Dieses Ergebnis ist auch in **Tabelle 2.3-1** dargestellt.

Lösung zu Aufgabe 7:

Zunächst wird die Ausgangsdatenrate des SMPTE 296M Videosignals nach **Abb. 2.3-4** berechnet:

$$R = 1920\frac{Pixel}{Zeile} \cdot 1080\frac{Lines}{Frame} \cdot 10\frac{Bit}{Pixel} \cdot 25\frac{Frames}{s} \cdot 2 = 1036800000\mathrm{Bit/s} \quad 8$$

Anmerkung: Die Multiplikation mit 2 ergibt sich aus dem Abtastformat 4:2:2. Mit den Umrechnungen (siehe Hinweis) ergibt sich:

$$R = 988.76953Mbit/s \quad 9$$

Anschließend wird R durch die Kanalkapazität von 10 Mbit/s dividiert:

$$\frac{R}{10} = 98.876953 \quad 10$$

Aufgerundet ergibt sich ein Wert von 99.

Lösung zu Aufgabe 8:

a) Lösung: d
Die Transformierte einer Matrix erhält man, in dem man Zeilen und Spalten vertauscht. Dies entspricht einer Spiegelung der Koeffizienten an der Diagonalen von oben links nach unten rechts.

b) Die Inverse einer Matrix errechnet sich über die Matrixgleichung:

$$\underline{A} \cdot \underline{A}^{-1} = \underline{I} \tag{19}$$

wobei I die Einheitsmatrix darstellt. Da einige Zahlen vorgegeben sind muss nicht das komplette Gleichungssystem gelöst werden.

Für die Berechnung von a wird z. B. die zweite Zeile von $\underline{A}$ mit der ersten Spalte von $\underline{A}^{-1}$ multipliziert. Das Ergebnis ist in der Einheitsmatrix 0:

$$4 \cdot 7 + 1 \cdot a + 0 \cdot 3 = 28 + a = 0, \mathrm{a} = -28 \tag{20}$$

Für die Berechnung von b wird z. B. die zweiten Zeile von$\underline{A}$ mit der dritten Spalte von $\underline{A}^{-1}$ multipliziert. Das Ergebnis ist in der Einheitsmatrix 0:

$$4 \cdot 2 + 1 \cdot b + 0 \cdot 1 = 8 + b = 0, \mathrm{b} = -8 \tag{21}$$

Für die Berechnung von c wird z. B. die dritte Zeile von $\underline{A}$ mit der zweiten Spalte von $\underline{A}^{-1}$ multipliziert. Das Ergebnis ist in der Einheitsmatrix 0:

$$1 \cdot (-2) + 1 \cdot 9 + 7 \cdot c = 7 + 7c = 0, \mathrm{c} = -1 \tag{22}$$

Ergebnis: Die Matrix $\underline{A}$ ist nicht orthogonal.

Lösung zu Aufgabe 9:

Für allgemeine M und N ist die allgemeine DCT-Transformation in der **Gl.** dargestellt. Mit M = N = 3 ergibt sich nach Umformung:

$$F_{DCT}(u,v) = \frac{2}{3} C(u) \cdot C(v) \sum_{x=0}^{2} \sum_{y=0}^{2} B(x,y) \cdot \cos\left[\frac{(2x+1)\pi \cdot u}{6}\right] \cos\left[\frac{(2y+1)\pi \cdot v}{6}\right]$$

$$C(u) = \begin{cases} \sqrt{1/2} & \text{fr u} = 0 \\ 1 & \text{fr u} \neq 0 \end{cases}$$

$$\mathrm{C(v)} = \begin{cases} \sqrt{1/2} & \text{fr v} = 0 \\ 1 & \text{fr v} \neq 0 \end{cases}$$

Für den DC-Koeffizienten gilt u = v = 0. Damit lässt sich die Formel der DCT für diesen Koeffizienten leicht vereinfachen:

$$F_{DCT}(0,0) = \frac{1}{3} \sum_{x=0}^{2} \sum_{y=0}^{2} B(x,y)$$

Damit ergibt sich der DC-Koeffizient aus der Summe aller Werte im Block geteilt durch 3:

$$F_{DCT}(0,0) = F_{DC} = \frac{612}{3} = 204$$

Lösung zu Aufgabe 10:

Der Gleichanteil entspricht a und errechnet sich zu:

$$a = \frac{1}{2\sqrt{2}} \cdot \sum_{x=0}^{7} f(x-128) = \frac{1}{2\sqrt{2}} \cdot (-224) = -\frac{112}{\sqrt{2}} \approx -79.196, \qquad 29$$

gerundet: –79.

b ist der quantisierte und gerundete Gleichanteil:

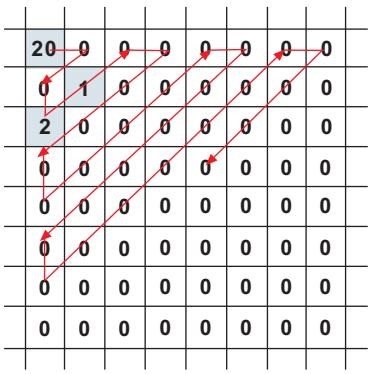
$$b = Round\left(\frac{a}{16}\right) = Round\left(\frac{-79}{16}\right) = -5 \qquad 30$$

Da im Signal nur ein Gleichanteil vorhanden ist, müssen alle höherwertigen Koeffizienten gleich 0 sein. So ist auch c = 0.

Die Rekonstruktion des Ausgangssignal gelingt exakt. Damit entsprechen d und e wieder den Ausgangswerten: d = 100, e = 100.

Lösung zu Aufgabe 11:

a) Lösung: c
Die Koeffizienten werden aus dem DCT-Block mit dem Zick-Zack-Scan ausgelesen (vgl. auch Bild **Abb. 3.3-1**).



Damit ergibt sich die Zahlenfolge: 20, 0, 0, 2, 1, 0, 0 ... Die Vorgehensweise zum Zusammenfassen mit der RLC ist in Kapitel **Abschnitt 3.3.2** dargestellt. Der Gleichanteil wird direkt angegeben: (20). Die nachfolgenden AC-Koeffizienten sind

solange in Zahlenpaaren angegeben, solange es noch Koeffizienten ungleich 0 gibt. In diesem Beispiel gibt es zwei Koeffizienten. Daher werden zwei Zahlenpaare für die Codierung mit der RLC benötigt. Die erste Zahl gibt jeweils die Anzahl von Nullen vor der eigentlichen Ziffer an. Damit ergeben sich die Zahlenpaare: (2,–2) und (0,1). Der Wert EOB gibt an, dass im Block nur noch Nullen folgen (End of Block).

b) Lösung: a
Zunächst wird die VLC-Tabelle erstellt wie in **Tabelle 3.3-4** erstellt.

RLC	Kategorie / Code	Huffman-Code	Ergebnis
DC: 20	5 / 10100	110	110 10100
(2,–2)	02. Jan	11111001	11111001 01
–0,1	0 / 1	0	00 1
EOB	1010		

Das Ergebnis ist 3 Byte und 1 Bit lang: 11010100 11111001 01001101 0.

Lösung zu Aufgabe 12:

Das Mittelwertsignal ergibt sich, wenn man den Mittelwert von aufeinander folgenden Werten errechnet. In diesem Fall ergibt sich für das Beispiel: 70 87 96 98 28 15 22 25.

Die entsprechende Differenzfolge lautet: –6 –3 –2 0 16 –3 2 1.

Lösung zu Aufgabe 13:

Allgemein sei der Bildausschnitt durch die folgende Matrix gegeben:

$$B(x,y) = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \tag{33}$$

Im ersten Schritten werden spaltenweise die Tiefenanteile T (Haar-Scaling Funktion) und Höhenanteile H (Haar-Wavelet) errechnet:

Haar-Scaling Funktion ist in Kapitel **Abschnitt 3.4.4** als Mittelwertsignal beschrieben, das Haar-Wavelet entspricht dem Differenzsignal. Damit errechnen sich T und H zu

$$T : \frac{1}{2} \begin{pmatrix} a+b & c+d \\ e+f & g+h \\ i+j & k+l \\ m+n & o+p \end{pmatrix} \tag{34}$$

$$H : \frac{1}{2}\begin{pmatrix} a-b & c-d \\ e-f & g-h \\ i-j & k-l \\ m-n & o-p \end{pmatrix} \qquad 35$$

Anschließend wird zeilenweise auf T und H jeweils noch einmal das Höhen- und das Tiefensignal gebildet. Es ergibt sich dann:

$$TT : \frac{1}{4}\begin{pmatrix} a+b+e+f & c+d+g+h \\ i+j+m+n & k+l+o+p \end{pmatrix} \qquad 36$$

$$TH : \frac{1}{4}\begin{pmatrix} a+b-e-f & c+d-g-h \\ i+j-m-n & k+l-o-p \end{pmatrix}$$

$$HT : \frac{1}{4}\begin{pmatrix} a-b+e-f & c-d+g-h \\ i-j+m-n & k-l+o-p \end{pmatrix}$$

$$HH : \frac{1}{4}\begin{pmatrix} a-b-e+f & c-d-g+h \\ i-j-m+n & k-l-o+p \end{pmatrix}$$

Mit der Anordnung der vier 2 x 2 Matrizen gemäß der Skizze in Bild 3.22 ergibt sich für BHaar(u,v):

$$B_{Haar}(u,v) = \frac{1}{4}\begin{pmatrix} a+b+e+f & c+d+g+h & a-b+e-f & c-d+g-h \\ i+j+m+n & k+l+o+p & i-j+m-n & k-l+o-p \\ a+b-e-f & c+d-g-h & a-b-e+f & c-d-g+h \\ i+j-m-n & k+l-o-p & i-j-m+n & k-l-o+p \end{pmatrix}$$

Setzt man nun die Zahlenwerte für a bis p der Aufgabenstellung in die Gleichung ein ergibt sich das Ergebnis zu:

$$B_{Haar}(u,v) = \frac{1}{q}\begin{pmatrix} a' & b' & c' & d' \\ e' & f' & g' & h' \\ i' & j' & k' & l' \\ m' & n' & o' & p' \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 7 & 10 & 3 & -4 \\ 12 & 16 & -2 & 0 \\ 7 & 2 & 3 & 0 \\ 2 & 6 & 0 & -2 \end{pmatrix}$$

Lösung zu Aufgabe 14:

a) Angewendet werden die Formeln zur Codiereffizienz aus Kapitel 3.5.4. Es gilt zu beachten, dass sich im Beispielblock nur wenige Bildpunkte zwischen Origi-

nal und dem codierten Ergebnis unterscheiden. Das vereinfacht die Rechnung sehr. Zunächst erstellt man die quadratischen Differenzen:

$$(s(x,y)-s'(x,y))^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 1 & 4 & 0 \\ 0 & 4 & 1 & 4 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Der MSE berechnet sich aus der Summe der Koeffizienten dividiert durch die Anzahl der Koeffizienten, die in diesem Block 64 beträgt:

$$MSE(s,s') = \frac{4+1+4+4+1+4+1+1+1+4+1+4+1+1}{64} = \frac{32}{64} = 0.5$$

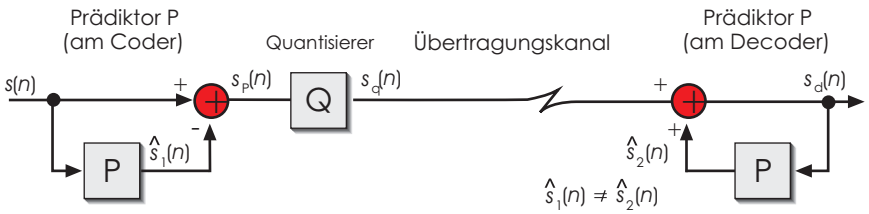
b) Der PSNR berechnet sich nach **Gl.3.5-2** aus **Abschnitt 3.5.4**, wobei der MSE aus **a)** benutzt werden kann:

$$\text{PSNR}(s,s') = 10{\cdot}\log_{10}\left(\frac{255^2}{MSE(s,s')}\right) = 10{\cdot}\log_{10}\left(2\cdot 255^2\right) = 10{\cdot}\log_{10}\left(130050\right) \approx 51.114$$

46

Lösung zu Aufgabe 15:

a) Mit dem Prinzipschaltbild **Abb. 4.2-1** lässt sich eine Tabelle aufstellen, in der alle wichtigen Zustände für das o. a. Eingangssignal eingetragen sind.



Deutlich ist zu erkennen, dass die D*PCM am Ausgang des Decoders zur Oszillation neigt, d. h. das Stabilitätskriterium nicht in jedem Fall gewährleistet ist.

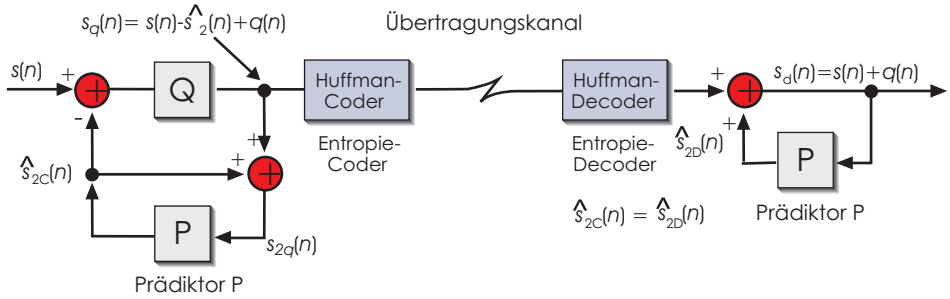
Tab. 1: *Zustände*

	Zustände n													
s(n)	0	7	9	10	8	8	9	7	6	5	9	0	0	...
$\hat{s}_1(n)$	0	0	7	9	10	8	8	9	7	6	5	9	0	...
$s_p(n)$	0	7	2	1	–2	0	1	–2	–1	–1	4	–9	0	...
$s_q(n)$	0	6	2	0	–2	0	0	–2	0	0	4	–8	0	...
$s_d(n)$	0	6	2	6	0	6	0	4	0	4	4	–4	4	...
$\hat{s}_2(n)$	0	0	6	2	6	0	6	0	4	0	4	4	–4	...

Tab. 2: *Fortsetzung Zustände*

	Zustände n													
s(n)	...	0	9	8	7	6	5	4	7	8	9	9	8	7
$\hat{s}_1(n)$	...	0	0	9	8	7	6	5	4	7	8	9	9	8
$s_p(n)$	...	0	9	–1	–1	–1	–1	–1	3	1	1	0	–1	–1
$s_q(n)$	...	0	8	0	0	0	0	0	2	0	0	0	0	0
$s_d(n)$	...	–4	12	–4	12	–4	12	–4	14	–4	14	–4	14	–4
$\hat{s}_2(n)$	...	4	–4	12	–4	12	–4	12	–4	14	–4	14	–4	14

b) Mit dem Prinzipschaltbild **Abb. 4.2-3** lässt sich eine Tabelle aufstellen, in der alle wichtigen Zustände für das o. a. Eingangssignal eingetragen sind.



Tab. 3: *Zustände*

	Zustände n													
s(n)	0	7	9	10	8	8	9	7	6	5	9	0	0	...
$s_q(n)$	0	6	2	2	–2	0	0	0	–2	0	2	–8	0	...
$\hat{s}_{2C}(n)$	0	0	6	8	10	8	8	8	8	6	6	8	0	...
$S_{2p}(n)$	0	6	8	10	8	8	8	8	6	6	8	0	0	...
$s_d(n)$	0	6	8	10	8	8	8	8	6	6	8	0	0	...
$\hat{s}_{2D}(n)$	0	0	6	8	10	8	8	8	8	6	6	8	0	...

Tab. 4: *Fortsetzung Zustände*

	Zustände n													
s(n)	...	0	9	8	7	6	5	4	7	8	9	9	8	7
$s_q(n)$	...	0	8	0	0	–2	0	–2	2	2	0	0	0	0
$\hat{s}_{2C}(n)$	...	0	0	8	8	8	6	6	4	6	8	8	8	8
$S_{2p}(n)$	...	0	8	8	8	6	6	4	6	8	8	8	8	8
$s_d(n)$	...	0	8	8	8	6	6	4	6	8	8	8	8	8
$\hat{s}_{2D}(n)$	...	0	0	8	8	8	6	6	4	6	8	8	8	8

Erkennbar wird **Gl. 4.2-1** eingehalten: Das Ausgangssignal $s_d(n)$ entspricht exakt dem Eingangssignal $s(n)$ bis auf den Quantisierungsfehler $q(n)$. Zudem erzeugen ersichtlich der Prädiktor am Coder das gleiche Signal, wie der Prädiktor am Decoder:

$$\hat{s}_{2C}(n) = \hat{s}_{2D}(n) \qquad 50$$

Die Huffman-Codierung ist überdies eine Redundanzreduktion und hat somit keinen Einfluss auf das Codierergebnis.

Lösung zu Aufgabe 16:

a) In **Gl.4.3-8** ist die maximale Suchweite angegeben. Für das 3-schrittige TDL-Verfahren ist d_{max} = 8. Damit ergibt sich N_{TDL} zu 17. Dies wurde auch schon mal in der Selbstlernaufgabe 4.9 errechnet.

b) In **Abb. 4.3-7** Mitte sind die Suchpositionen für 3 Suchschritte angegeben. In der größten Suchweite (4 Bildpunkte) wird einschließlich dem Nullvektor fünfmal die MSE berechnet.

c) dx_1 = 0

dy_1 = –4

Siehe **d)**.

d) Für den ersten Suchschritt ergeben sich folgende Werte:

MSE_1*(0,0) = 2658,* MSE_1*(0,–4) = 2046,* MSE_1*(0,4) = 2644,* MSE_1*(–4,0) = 2405,* MSE_1*(4,0) = 3012*. Der kleinste Wert liegt bei dx = 0, dy = –4.

e) In **Abb. 4.3-7** Mitte sind die Suchpositionen für 3 Suchschritte angegeben. In der zweiten Suchweite (2 Bildpunkte) wird nur viermal die MSE berechnet.

f) dx_2 = 0, dy_2 = –6

Siehe **g)**!

g) Für den zweiten Suchschritt ergeben sich folgende Werte:

MSE_2(0,–6) = 7, MSE_2(0,–2) = 3511, MSE_2(2,–4) = 2980, MSE_2(–2,–4) = 2455.

Der kleinste Wert liegt bei *dx* = 0, *dy* = –6.

h) In **Abb. 4.3-7** Mitte sind die Suchpositionen für 3 Suchschritte angegeben. In der dritten Suchweite (1 Bildpunkt) wird achtmal die MSE berechnet.

i) dx_3 = 1, dy_3 = –6

Siehe **j)** !

j) Für den dritten Suchschritt ergeben sich folgende Werte:

MSE_3(0,–7) = 1201, MSE_3(0,–5) = 1854, MSE_3(–1,–7) = 1116, MSE_3(–1,–6) = 15, MSE_3(–1,–5) = 1722, MSE_3(1,–7) = 1291, MSE_3(1,–6) = 0, MSE_3(1,–5) = 1652.

Der kleinste Wert liegt bei *dx* = 1, *dy* = –6.

Das Bild veranschaulicht noch einmal die Suchpositionen der einzelnen Suchschritte:
1. Schritt (violett, rot)
2. Schritt (cyan)
3. Schritt (gelb, grün)
An der Position dx = 1 und dy =–6 ist der Referenzblock fast gleich dem Suchblock B(x,y).

x →

y ↓	0	1	2	3	4	5	6	7	8	9	10
0	34	35	20	34	40	34	45	47	48	50	44
1	62	57	54	30	23	21	21	20	21	18	15
2	59	55	57	51	50	49	52	47	60	62	65
3	97	99	98	102	97	99	98	102	102	105	117
4	102	103	105	65	49	60	48	76	126	127	129
5	111	107	107	109	107	109	110	115	117	119	121
6	97	88	87	85	85	90	97	86	86	80	79
7	70	73	70	65	67	68	70	71	70	74	78
8	50	54	52	53	56	56	55	52	51	49	49
9	34	32	29	20	25	23	26	27	23	20	21
10	49	50	51	49	48	47	49	50	52	53	51
11	80	81	83	84	90	93	95	98	99	100	102
12	85	89	90	101	102	100	99	98	97	95	99
13	87	86	85	90	97	87	90	93	95	97	99

k) Lösung: d

Das TDL-Verfahren findet bereits im ersten Suchschritt eine abweichende Verschiebung, was dazu führt, dass das absolute Minimum mit diesem Verfahren nicht erreicht werden kann. Für den ersten Suchschritt ergeben sich folgende Werte:

MSE_1(0,0) = 2658, MSE_1(0,–4) = 3645, MSE_1(0,4) = 2644, MSE_1(–4,0) = 2405, MSE_1(4,0) = 3012. Der kleinste Wert liegt nicht mehr bei *dx* = 0, *dy* = –4 sondern bei *dx* = –4, *dy* = 0. Von diesem Startwert aus können auch die weiteren Suchschritte mit reduzierter Suchweite nicht mehr die Verschiebung von *dx* = 1, *dy* = –6 erreichen.

Das TSS-Verfahren (vgl. **Abschnitt 4.3.4**, **Abb. 4.3-7**) prüft im ersten Schritt zusätzlich 4 Suchpunkte, diese ebenfalls im 4 Punkte Raster. Für den ersten Suchschritt ergeben sich folgende Werte:

MSE_1(0,0) = 2658, MSE_1(0,–4) = 3645, MSE_1(0,4) = 2644, MSE_1(–4,0) = 2405, MSE_1(4,0) = 3012, MSE_1(–4,–4) = 3013, MSE_1(4,–4) = 4733, MSE_1(–4,4) = 1807, MSE_1(4,4) = 2899. Der kleinste Wert liegt nicht mehr bei *dx* = 0, *dy* = –4 sondern bei *dx* = –4, *dy* = 4. Von diesem Startwert aus können auch die weiteren Suchschritte mit reduzierter Suchweite nicht mehr die Verschiebung von *dx* = 1, *dy* = –6 erreichen.

Das CSA-Verfahren sucht eine Untermenge von Punkten des TSS-Verfahrens ab. Es kann also auch nicht das ursprüngliche Minimum finden.

Der Full-Search Algorithmus sucht jede Position des Suchbereiches ab. Es wird das absolute Minimum bei *dx* = 1, *dy* = –6 finden, da sich für diesen Bereich die Bildpunkte nicht geändert haben.

Lösung zu Aufgabe 17:

a)

$$D_u = 720 \cdot 576 \cdot 2 = 829440\ Byte = \ 810\ kByte$$

b) Mit Hilfe der Gl. 5.5 und einem einfachen Dreisatzes ergibt sich:

$$D_P = \frac{52}{140} \cdot 128\ kByte \approx 48\ kByte$$

$$D_B = \frac{36}{140} \cdot 128\ kByte \approx 33\ kByte$$

c) In der folgende Tabelle sind die Kompressionsraten für die Bildtypen I, P und B eingetragen, wobei die Verhältnisse aus TM5 Gl. 5.5 angenommen wurden.

Bildtyp	$D_{I,P,B}$ in kByte	Kompressionsrate ca.
I	128	6:1
P	47.54	17:1
B	32.91	25:1

Dieses Beispiel verdeutlicht auch, dass mit kurzen GOPs geringere Kompressionsraten als mit langen GOPs möglich ist. Für das Beispiel der 6er-GOP I B B P B B ergibt sich eine durchschnittliche Kompressionsrate von

$$D : D_{6er-GOP} = \frac{6 \cdot 810\ kByte}{128\ kByte + 48\ kByte + 4 \cdot 33\ kByte} \approx 16 : 1$$

Die 13er-GOP I B B B P B B B P B B B P erzielt eine durchschnittliche Kompressionsrate von

$$D : D_{13er-GOP} = \frac{13 \cdot 810\ kByte}{128\ kByte + 3 \cdot 48\ kByte + 9 \cdot 33\ kByte} \approx 19 : 1$$

Lösung zu Aufgabe 18:

In **Abschnitt 5.3.2** sind die unterschiedlichen Profiles und Levels von MPEG-2 beschrieben. In **Tabelle 5.3-4** ist das Farbabtastraster für den Main Profile mit 4:2:0 angeben. In **Tabelle 5.3-6** sind die Auflösung und die Datenrate für den MP@HL angegeben. Von der angegebenen Datenrate muss für die Rechnung 10 % abgezogen werden:

$$R = \frac{1920 \cdot 1088 \cdot 8Bit \cdot 3 \cdot 25Hz}{1024 \cdot 1024 \cdot 2 \cdot 0.9 \cdot 80Mbit/s} \approx 8.3 \qquad 59$$

Lösung zu Aufgabe 19:

a) Lösung: b,d

Das Zusammenfassen bei der Frame-Codierung geschieht mit dem Zick-Zack-Scan, der bereits in **Abb. 3.3-1** dargestellt wurde. Es existieren 23 von Null verschiedene Koeffizienten, die relativ weit gestreut sind. Ein relativ hohes Datenaufkommen kann vermutet werden. Das Zusammenfassen bei der Field-Codierung geschieht mit dem Alternate Scanning (**Abb. 5.3-5**). Es existieren nur 8 von Null verschiedene Koeffizienten. Das führt voraussichtlich zu einem deutlich geringeren Datenaufkommen.

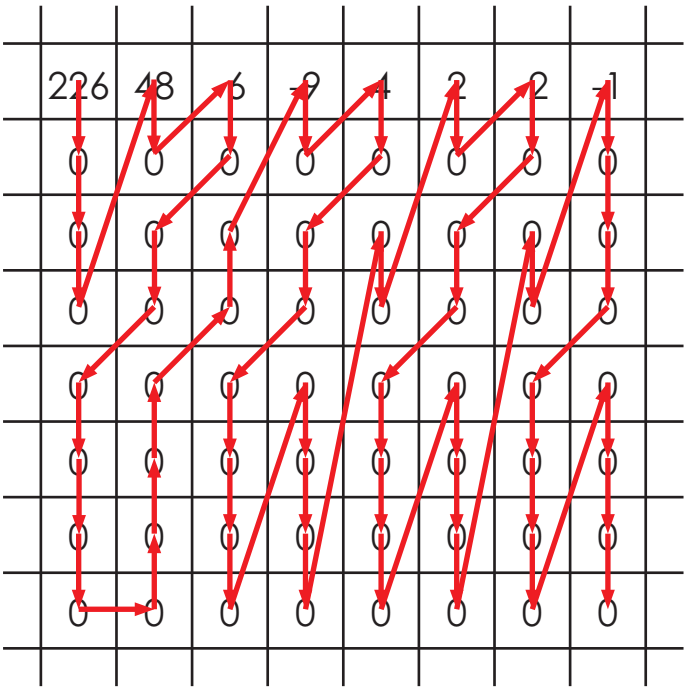
Die Field- und Frame-Methode sind in **Abschnitt 5.3.3** beschrieben. Die unterschiedliche Zusammenfassung der Halbbilder in den Blöcken führt dazu, dass sich auch unterschiedliche Bildinhalte in den Blöcken befinden müssen (vgl. **Abb. 5.3-3** und **Abb. 5.3-4**).

b) Der DCT-Block wird mit dem Alternate-Scanning zusammengefasst (vgl. Bild). Mit dem in **Abschnitt 3.3.2** beschriebenen Verfahren werden die RLC-Wertepaare gebildet:

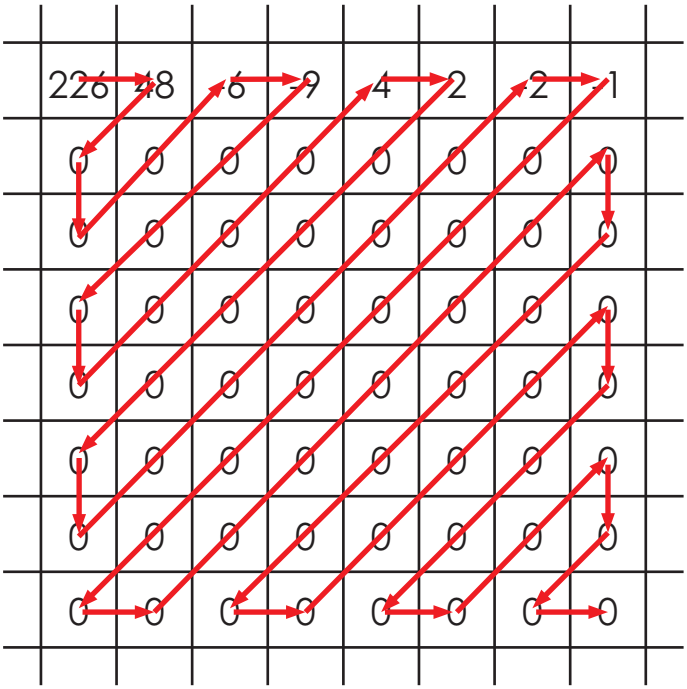
$RLC_{\text{Field,Alternate Scanning}}$ = (226) (3,48) (1,–6) (13,–9) (1,4) (13,2) (1,–2) (13,–1) *EOB*

Diese Wertepaare werden wie üblich mit den Tabellen **Tabelle 3.3-1** bis **Tabelle 3.3-4** VLC-codiert. In dieser Aufgabe genügt es, nur die Codewortlängen zu notieren und aufzuaddieren:

(226): 8+6 Bit, (3,48): 6+16 Bit, (1,–6): 3+7 Bit, (13,–9): 4+16 Bit, (1,4): 3+7 Bit, (13,2): 2+16 Bit, (1,–2): 2+5 Bit, (13,–1): 1+11 Bit, EOB: 4 Bit. In der Summe ergibt sich der Wert von 117 Bit.



c) Der DCT-Block wird mit dem Zick-Zack-Scan zusammengefasst (vgl. Bild).



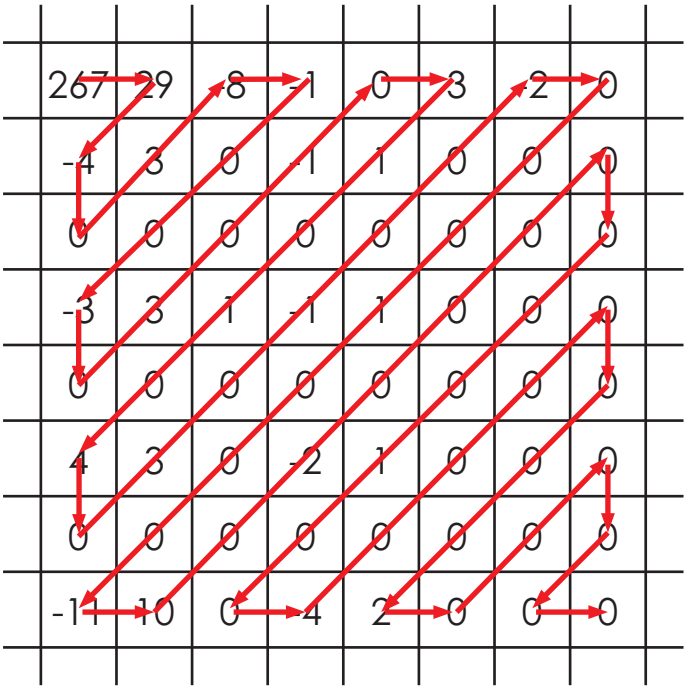
Mit dem in **Abschnitt 3.3.2** beschriebenen Verfahren werden die RLC-Wertepaare gebildet:

$RLC_{\text{Field,Zick-Zack-Scan}}$ = (226) (0,48) (3,–6) (0,–9) (7,4) (0,2) (11,–2) (0,–1) *EOB*

Diese Wertepaare werden wie üblich mit den Tabellen **Tabelle 3.3-1** bis **Tabelle 3.3-4** VLC-codiert. In dieser Aufgabe genügt es wiederum, nur die Codewortlängen zu notieren und aufzuaddieren:

(226): 8+6 Bit, (0,48): 6+7 Bit, (3,–6): 3+12 Bit, (0,–9): 4+4 Bit, (7,4): 3+16 Bit, (0,2): 2+2 Bit, (11,–2): 2+16 Bit, (0,–1): 1+2 Bit, EOB: 4 Bit. In der Summe ergibt sich der Wert von 98 Bit. In diesem Fall ist eine Zusammenfassung mit dem standardmäßigen Zick-Zack-Scan also günstiger als das Alternate Scannig.

d) Der DCT-Block wird mit dem Zick-Zack-Scan zusammengefasst (vgl. Bild).



Mit dem in **Abschnitt 3.3.2** beschriebenen Verfahren werden die RLC-Wertepaare gebildet:

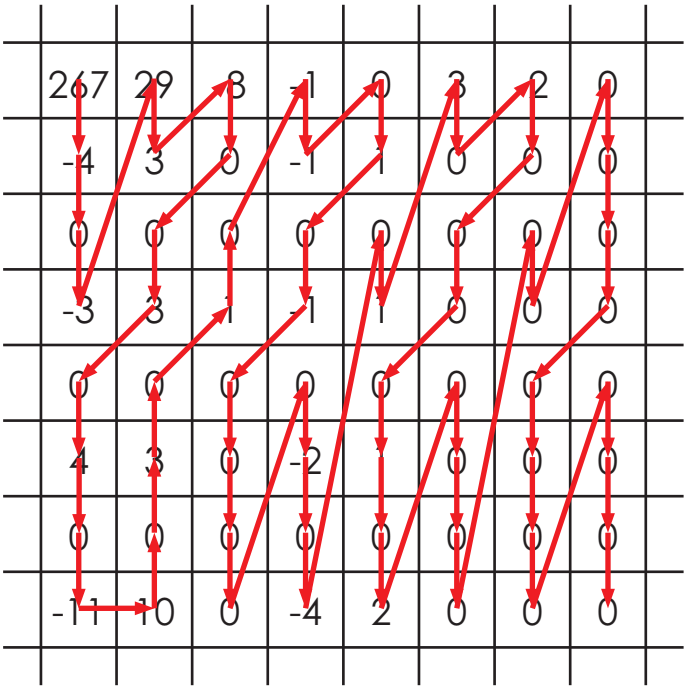
$RLC_{\text{Frame,Zick-Zack-Scan}}$ = (267) (0,29) (0,–4) (1,3) (0,–8) (0,–1) (2,–3) (1,3) (1,–1) (1,3) (0,1) (1,1) (1,–4) (1,3) (1,–1) (2,–2) (3,1) (3,–11) (0,10) (1,–2) (7,1) (2,–4) (7,2) *EOB*

Diese Wertepaare werden mit den Tabellen **Tabelle 3.3-1** bis **Tabelle 3.3-4** VLC-codiert. In dieser Aufgabe genügt es wiederum, nur die Codewortlängen zu notieren und aufzuaddieren:

(267): 9+7 Bit, (0,29): 5+5 Bit, (0,–4): 3+3 Bit, (1,3):3+7 Bit, (0,–8): 4+4 Bit, (0,–1): 1+2 Bit, (2,–3):2+8 Bit, (1,3):2+5 Bit, (1,–1): 1+4 Bit, (1,3):2+5 Bit, (0,1): 1+2 Bit, (1,1): 1+4 Bit, (1,–4): 3+7 Bit, (1,3): 2+5 Bit, (1,–1): 1+4 Bit, (2,–2): 2+8 Bit, (3,1): 1+6 Bit, (3,–11): 4+16 Bit, (0,10): 4+4 Bit, (1,–2): 2+5 Bit, (7,1): 3+16 Bit, (2,–4): 3+10 Bit, (7,2): 2+12 Bit, EOB: 4 Bit.

In der Summe ergibt sich der Wert von 201 Bit.

e) Der DCT-Block wird mit dem Alternate Scanning zusammengefasst (vgl. Bild).



Mit dem in **Abschnitt 3.3.2** beschriebenen Verfahren werden die RLC-Wertepaare gebildet:

$RLC_{\text{Frame,Alternate Sanning}}$ = (267) (0,–4) (1,–3) (0,29) (0,3) (0,–8) (2,3) (1,4) (1,11) (0,10) (1,3) (1,1) (1,–1) (0,–1) (1,1) (1,–1) (6,–2) (1,–4) (1,1) (0,3) (1,2) (4,1) (1,2) *EOB*

Diese Wertepaare werden mit den Tabellen **Tabelle 3.3-1** bis **Tabelle 3.3-4** VLC-codiert. In dieser Aufgabe genügt es wiederum, nur die Codewortlängen zu notieren und aufzuaddieren:

(267): 9+7 Bit, (0,–4): 3+3 Bit, (1,–3): 2+5 Bit, (0,29): 5+5 Bit, (0,3):2+2 Bit, (0,–8): 4+4 Bit, (2,3): 2+8 Bit, (1,4): 3+7 Bit, (1,11): 4+9 Bit, (0,10): 4+4 Bit, (1,3): 2+5 Bit, (1,1): 1+4 Bit, (1,–1): 1+4 Bit, (0,–1): 1+2 Bit, (1,1): 1+4 Bit, (1,–1): 1+4 Bit, (6,–2): 2+12 Bit, (1,–4): 3+7 Bit, (1,1): 1+4 Bit, (0,3): 2+2 Bit, (1,2): 2+5 Bit, (4,1): 1+6 Bit, (1,2): 2+5 Bit, EOB: 4 Bit.

In der Summe ergibt sich der Wert von 172 Bit. In diesem Block lässt sich mit dem Alternate Scanning eine geringeres Datenaufkommen erreichen.

Lösung zu Aufgabe 20:

Die Eingangsdaten für die *S-VCD* sind:
- Auflösung 480 x 576 Pixel pro Vollbild bzw. 480 x 288 pro Halbbild
- Abtastung: 4:2:0, 8 Bit pro Komponente
- 625-Zeilen System entspricht 50 Hz Zeilensprung.

Berechnung der Eingangsdatenrate:

$$R_{Input} = 480 \cdot 288 Bit \cdot 8\frac{3}{2} \cdot 50 Hz = 82944000 \text{ Bit/s } = 79 \text{ Mbit/s.} \quad 60$$

Berechnung der Kompressionsrate:

$$R_{Input} \,:\, R_{Output} \,=\, 79\, Mbit/s \,:\, 2.46\, Mbit/s \,=\, 32. \quad 61$$

Lösung zu Aufgabe 21:

Das DVD-Medium hat eine Speicherkapazität von 4700000000 Byte. Der S-VCD Datenstrom benötigt pro Minute:

$$D_{S-VCD,Minute} = \frac{2520\ kbit \cdot Byte \cdot 60s}{8\ Bit \cdot s} \quad = 18900 \text{ kByte} = 19353600 \text{ Byte}$$

Damit passen auf ein DVD-Medium

$$\text{Kapazität } = \frac{4700000000}{19353600} \approx 242.8489 \text{ Minuten} \quad 63$$

Das entspricht knapp 243 Minuten S-VCD Video.

Lösung zu Aufgabe 22:

Die Energieverwischung wird in **Abschnitt 6.4.1** vorgestellt. Ein Blockschaltbild zur Generierung ist in **Animation 6.4-1** dargestellt. Gesucht ist die Symbolfolge nach der Initialisierung am Punkt A. Für die EX-OR Verknüpfung gilt folgende logische Verknüpfungstabelle:

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	0

Das in der elektronischen Version des Lehrtextes enthaltende Flash-Applet kann helfen, das äquivalente Signal zu generieren. Dazu muss das Enable-Signal auf High

(1) stehen und am Eingang "uncodierte Daten" muss eine Null-Folge mit 80 Nullen eingegeben werden. Am Ausgang wird dann sequenziell die zugehörige Bitfolge erzeugt. Es ist zu beachten, dass das erste am Ausgang erscheinende Bit das MSB des ersten Symbols darstellt, also in umgekehrter Reihenfolge notiert und zusammengefasst werden muss.

Ergebnis der ersten 10 Bytes: 03_H, $F6_H$, 08_H, 34_H, 30_H, $B8_H$, $A3_H$, 93_H, $C9_H$, 68_H.

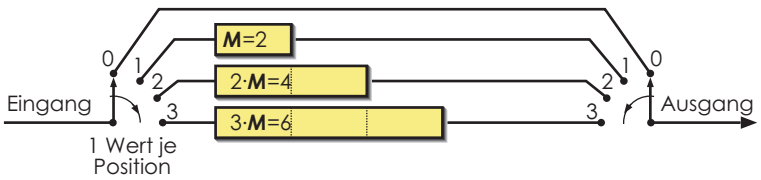
Lösung zu Aufgabe 23:

a) Lösung: c
Die maximale Symbolverschiebung am Ausgang des Interleavers entspricht gleichzeitig die resultierende Gesamtverzögerung am Ausgang des Deinterleavers. Diese ist in **Gl. 6.4-12** angegeben:

$$V_S = D_{Forney} = I \cdot (I-1) \cdot M \qquad 69$$

Damit ist Lösung c richtig.

b) Die Werte lassen sich leicht mit Hilfe eines Blockschaltbildes entsprechend **Abb. 6.4-4** ermitteln.



In diesem Beispiel werden 3 Schieberegister mit einer Verzögerung von 2, 4 und 6 Werten (Symbolen) eingesetzt. Der Schalter am Eingang und am Ausgang wird jeweils nach einem Wert eine Position weiter geschaltet. Gleichzeitig schieben sich die Symbole in den Schieberegistern eine Symbolbreite (in diesem Fall: 1 Wert) nach rechts. Nach der Schalterstellung 3 wird wieder zyklisch mit der Schalterstellung 0 begonnen.

Die Wertefolge O(n) lautet damit:

O(n) = {1, 0, 0, 0, 5, 0, 0, 0, 9, 2, 0, 0, 136 , 0, 0, 17, 10, 3, 0, 21, 14, 7, 0, 25, 18, 11, 4, 29, 22, 15, 8, 33, 26, 19, 12, 37, 30, 23, 16}

Lösung zu Aufgabe 24:

a) In Bild 6.15 ist der Faltungscoder für den DVB-Standard angegeben. Zunächst wird I in Bit-Schreibweise umgerechnet:

I = {11110010, 11010110, 00101100}

Bei der Zuführung zum Faltungscode muss beachtet werden, dass das MSB des ersten Symbols zuerst genommen werden muss. Die mit dem +–Zeichen angegebenen Verknüpfungen ergeben sich aus einer Modulo-2 Berechnung der beiden Eingangskomponenten (auf Bit-Ebene). Dieses entspricht einer logischen EX-OR Verknüpfung. Als Ergebnis für x und y ergibt sich:

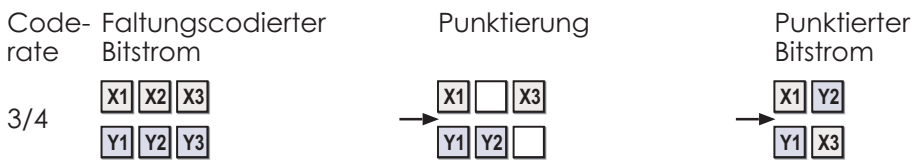
x = {10101010, 10011001, 00101100}

= $\{AA_H, 99_H, 2C_H\}$ = {170, 153, 44}

y= {11110110, 10001011, 11000101}

= $\{F6_H, 8B_H, C5_H\}$ = {246, 139,197}

b) Zur Veranschaulichung sei zunächst das Verfahren für die 3/4 Punktierung angegeben:



Die Datenströme *x* und *y* werden zunächst in Bit-Schreibweise umgewandelt, damit eine Zusammenfassung in 3-Bit langen Mustern möglich wird. Dabei muss beachtet werden, dass das MSB ganz links im Symbol liegt, also *X1* bzw. *Y1* entspricht:

$\underline{x}$ = {101, 010, 101, 001, 100, 100, 101, 100}

$\underline{y}$ = {111, 101, 101, 000, 101, 111, 000, 101}

Diese acht 3-Bit Muster werden nun punktiert und dabei in acht 2-Bit Muster umgewandelt:

x’ = {11, 00, 10, 00, 10, 11, 10, 10}

y’= {11, 10, 11, 01, 10, 10, 01, 10}

In hexadezimaler und dezimaler Schreibweise ergibt sich die Datenfolge für x’ und y’:

x’ = $\{C8_H, BA_H\}$ = {200, 186}

y’ = $\{ED_H, A6_H\}$ = {237, 166}

Lösung zu Aufgabe 25:

Die *Bandbreitenausnutzung B* liegt für eine 256-QAM im Idealfall bei 8 Bit pro Symbol. Wird das Roll-Off-Filter berücksichtigt mit *r* = 0.15 dann reduziert sich *B* zu

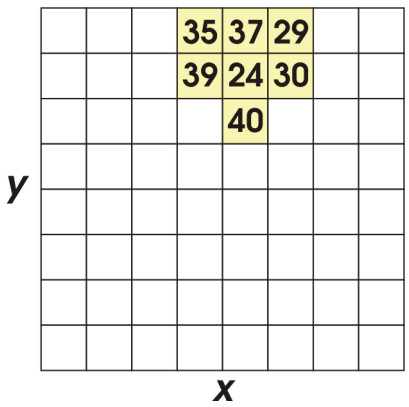
$$B_{256-QAM} = \frac{Bits\ pro\ Symbol}{\frac{B}{R_S}} = \frac{8Bit/Symbol}{1.15\frac{Hz}{Symbole/s}} = 6.96\frac{Bit}{s \cdot Hz}$$

Mit der Korrektur durch die Coderate des RS(204,188) ergibt sich ein Wert von

$$B_{256-QAM,RS(204,188)} = 6.96\frac{Bit}{s \cdot Hz} \cdot \frac{188}{204} = 6.41\frac{Bit}{s \cdot Hz}$$

Lösung zu Aufgabe 26:

Die Vorgehensweise für die SA-DCT bei MPEG-4 Video ist in **Abschnitt 7.2.3** beschrieben. Zunächst werden die Bildpunkte der Objektkontur vertikal an den oberen Rand des Referenzblockes geschoben (vgl. Bild). Anschließend ist für die Spalten eine eindimensionale DCT durchzuführen, in denen Bildpunkte der Objektkontur vorhanden sind.



Im vorliegenden Beispiel werden die Spalten 4 bis 6 einer eindimensionalen DCT-Transformation unterworfen.

Die Spalten 4 und 6 haben 2 Elemente, so dass gilt: *N* = 2. Die zugehörige Transformationsmatrix $\underline{A}$ vereinfacht sich aus **Gl. 7.2-2** zu

$$\underline{A} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Der Vorfaktor in **Gl. 7.2-1** wird zu 1. Die eindimensionale DCT-Transformation (Spaltentransformation für die 4. Spalte) ergibt sich nach **Gl. 7.2-1** dann:

$$\underline{s}_{k=4} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 35 \\ 39 \end{pmatrix}_{k=4} = \begin{pmatrix} \frac{74}{\sqrt{2}} \\ -2\sqrt{2} \end{pmatrix}$$

Die Spalte 5 hat 3 Elemente; also ist N = 3. Die Transformationsmatrix ergibt sich wieder aus **Gl. 7.2-2**:

$$\underline{A} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix}$$

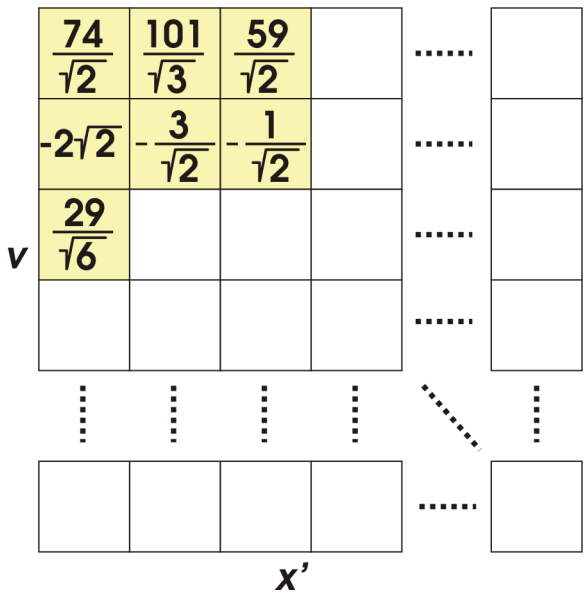
Unter der Berücksichtigung des Vorfaktors in **Gl. 7.2-1** ergibt sich die DCT-Transformation für die 5. Spalte:

$$\underline{s}_{k=5} = \sqrt{\frac{2}{3}} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 37 \\ 24 \\ 40 \end{pmatrix}_{k=5} = \begin{pmatrix} \frac{101}{\sqrt{3}} \\ -\frac{3}{\sqrt{2}} \\ \frac{29}{\sqrt{6}} \end{pmatrix}$$

Spalte 6 hat wieder 2 Elemente. Es kann daher auf die Matrix der Spalte 4 zurückgegriffen werden. Die eindimensionale DCT-Transformtion (Spaltentransformation für die 6. Spalte ergibt sich dann:

$$\underline{s}_{k=6} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 29 \\ 30 \end{pmatrix}_{k=6} = \begin{pmatrix} \frac{59}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Für die anschließende horizontale SA-DCT werden die transformierten Koeffzienten linksbündig an den linken Rand des Referenzblockes verschoben, wie im nachfolgenden Bild erkennbar ist.



Hier sind dann für die Zeile 1 bis 3 DCT-Transformationen durchzuführen:

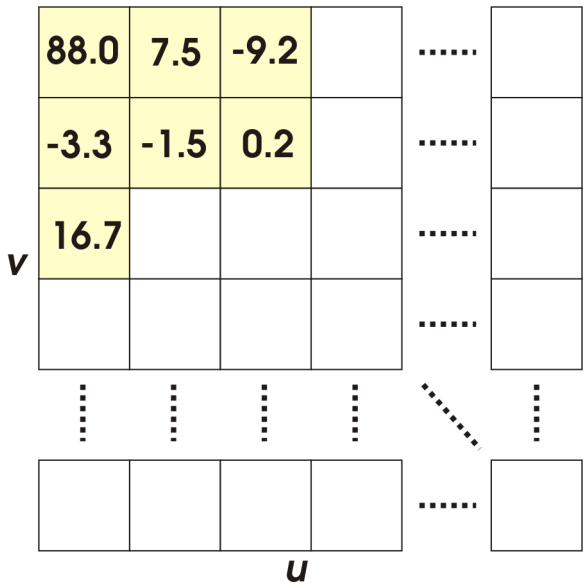
$$\underline{z}_{k=1} = \sqrt{\frac{2}{3}} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} \frac{74}{\sqrt{2}} \\ \frac{101}{\sqrt{3}} \\ \frac{59}{\sqrt{2}} \end{pmatrix}_{k=1} = \begin{pmatrix} \frac{101}{\sqrt{3}} + \frac{133}{\sqrt{6}} \\ \frac{15}{2} \\ \frac{133}{2\sqrt{3}} - \frac{101 \cdot \sqrt{2}}{3} \end{pmatrix} \approx \begin{pmatrix} 87.964 \\ 7.5 \\ -9.218 \end{pmatrix}$$

$$\underline{z}_{k=2} = \sqrt{\frac{2}{3}} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} -2\sqrt{2} \\ -\frac{3}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}_{k=2} = \begin{pmatrix} -\frac{8}{\sqrt{6}} \\ -\frac{3}{2} \\ \frac{1}{2 \cdot \sqrt{3}} \end{pmatrix} \approx \begin{pmatrix} -3.266 \\ -1.5 \\ 0.229 \end{pmatrix}$$

Für Zeile 3 (N = 1) ergibt sich der Koeffizient zu

$$\underline{z}_{k=3} = \sqrt{2} \cdot \frac{29}{\sqrt{6}} = \frac{29}{\sqrt{3}} \approx 16.743$$

Damit ergibt sich der SA-DCT transformierte Block:



Lösung zu Aufgabe 27:

Die Intra-Prädiktion für 4 x 4 Bildpunkte große Blöcke wird in **Abschnitt 7.3.5** vorgestellt. Die Prädiktion für den Mode 3 (diagonal nach links unten) errechnet sich nach **Gl. 7.3-10** wie folgt:

$$\text{Präd}[x,y] = \begin{cases} \frac{1}{4}\left(G+3H+2\right), & x=y=3 \\ \frac{1}{4}\left(p[x+y,-1]+2p[x+y+1,-1]+p[x+y+2,-1]+2\right), & sonst \end{cases} \quad 80$$

Zunächst wird eine Zuordnung zwischen *p*[*x*,*y*] aus dieser Gleichung und den Buchstaben der o. a. Skizze hergestellt:

A = *p*[0,–1], *B* = *p*[1,–1], *C* = *p*[2,–1], *D* = *p*[3,–1],

E = *p*[4,–1], *F* = *p*[5,–1], *G* = *p*[6,–1], *H* = *p*[7,–1].

Anschließend können die fehlenden Bildpunkte des aktuellen Blocks leicht berechnet werden:

$$c = p[0,2] = \frac{1}{4}\left(p[2,-1]+2p[3,-1]+p[4,-1]+2\right) = \frac{1}{4}\left(C+2D+E+2\right) = 33.5 \quad 81$$

$$e = p[1,0] = \frac{1}{4}\left(p[1,-1]+2p[2,-1]+p[3,-1]+2\right) = b = 30.5 \quad 82$$

$$h = p[1,3] = \frac{1}{4}\left(p[4,-1]+2p[5,-1]+p[6,-1]+2\right) = \frac{1}{4}\left(E+2F+G+2\right) = 32.5 \quad 83$$

$$j = p[2,1] = \frac{1}{4}\left(p[3,-1]+2p[4,-1]+p[5,-1]+2\right) = d = 34 \quad 84$$

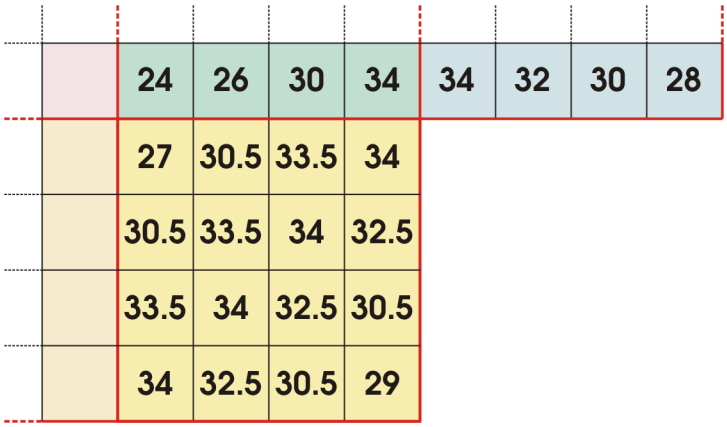
$$l = p[2,3] = \frac{1}{4}\left(p[5,-1]+2p[6,-1]+p[7,-1]+2\right) = \frac{1}{4}\left(F+2G+H+2\right) = 30.5 \quad 85$$

$$m = p[3,0] = \frac{1}{4}\left(p[3,-1]+2p[4,-1]+p[5,-1]+2\right) = d = 34 \quad 86$$

$$n = p[3,1] = \frac{1}{4}\left(p[4,-1]+2p[5,-1]+p[6,-1]+2\right) = h = 32.5 \quad 87$$

$$p = p[3,3] = \frac{1}{4}\left(G+3H+2\right) = 29 \quad 88$$

Das nachfolgendes Bild zeigt das Ergebnis für den ganzen Block. Deutlich zu erkennen ist, dass die Werte auf den Diagonalen innerhalb des Blockes identisch sind.



Mit den folgenden Gleichungen lassen sich die bereits gegebenen 8 Werte des Blockes überprüfen:

$$a = p[0,0] = \frac{1}{4}\left(p[0,-1] + 2p[1,-1] + p[2,-1] + 2\right) = \frac{1}{4}\left(A + 2B + C + 2\right) = 27$$

$$b = p[0,1] = \frac{1}{4}\left(p[1,-1] + 2p[2,-1] + p[3,-1] + 2\right) = \frac{1}{4}\left(B + 2C + D + 2\right) = 30.5$$

$$d = p[0,3] = \frac{1}{4}\left(p[3,-1] + 2p[4,-1] + p[5,-1] + 2\right) = \frac{1}{4}\left(D + 2E + F + 2\right) = 34$$

$$f = p[1,1] = \frac{1}{4}\left(p[2,-1] + 2p[3,-1] + p[4,-1] + 2\right) = c = 33.5$$

$$g = p[1,2] = \frac{1}{4}\left(p[3,-1] + 2p[4,-1] + p[5,-1] + 2\right) = d = 34$$

$$i = p[2,0] = \frac{1}{4}\left(p[2,-1] + 2p[3,-1] + p[4,-1] + 2\right) = c = 33.5$$

$$k = p[2,2] = \frac{1}{4}\left(p[4,-1] + 2p[5,-1] + p[6,-1] + 2\right) = h = 32.5$$

$$o = p[3,2] = \frac{1}{4}\left(p[5,-1] + 2p[6,-1] + p[7,-1] + 2\right) = l = 30.5$$

Lösung zu Aufgabe 28:

a) Die Integer-Transformation ist in **Abschnitt 7.3.6** beschrieben. Zunächst wird core-Transformation $\underline{W}$ gebildet. Allgemein ist $\underline{W}$ wie folgt zu ermitteln:

$$\underline{W} = (\underline{B} \cdot \underline{X})\underline{B}^T \qquad 99$$

Die Matrix $\underline{B}$ ergibt sich aus den Gleichungen 7.29, 7.30 und 7.31 für die DCT:

$$\underline{B} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{pmatrix}$$

$$d = \frac{\cos\left(\frac{3\pi}{8}\right)}{\cos\left(\frac{\pi}{8}\right)} \approx 0.41$$

Eingesetzt:

$$\underline{W} = \begin{pmatrix} 140 & -0.24 & -6 & 3.41 \\ -10.1 & -11.17 & 1.36 & -21.11 \\ 22 & 9.27 & 8 & 14.38 \\ -12.38 & -7.05 & -28.73 & -2.89 \end{pmatrix}$$

Die klassische DCT-Transformation $\underline{Y}_{DCT}$ lässt sich aus den Gleichungen **Gl. 7.3-22** und **Gl. 7.3-23** wie folgt errechnen:

$$\underline{Y}_{DCT} = \underline{W} \otimes \underline{C} = \left((\underline{B} \cdot \underline{X})\,\underline{B}^T\right) \otimes \underline{C}$$

wobei

$$\underline{C} = \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4} & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) \\ \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4}\cos^2\left(\frac{\pi}{8}\right) & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4}\cos^2\left(\frac{\pi}{8}\right) \\ \frac{1}{4} & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4} & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) \\ \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4}\cos^2\left(\frac{\pi}{8}\right) & \frac{1}{2\sqrt{2}}\cos\left(\frac{\pi}{8}\right) & \frac{1}{4}\cos^2\left(\frac{\pi}{8}\right) \end{pmatrix}$$

$$\underline{C} \approx \begin{pmatrix} 0.25 & 0.33 & 0.25 & 0.33 \\ 0.33 & 0.43 & 0.33 & 0.43 \\ 0.25 & 0.33 & 0.25 & 0.33 \\ 0.33 & 0.43 & 0.33 & 0.43 \end{pmatrix}$$

Das Symbol $\otimes$ gibt die skalare Multiplikation an. Damit ergibt $\underline{Y}_{DCT}$ zu:

$$\underline{Y}_{DCT} = \begin{pmatrix} 35 & -0.08 & -1.50 & 1.12 \\ -3.3 & -4.77 & 0.44 & -9.01 \\ 5.5 & 3.03 & 2 & 4.7 \\ -4.05 & -3.01 & -9.38 & -1.23 \end{pmatrix}$$

b) Die Bildung der core-Transformation $\underline{W}$ wurde bereits in **a)** gebildet. Die Matrix $\underline{B}'$ ergibt sich aus den Gleichungen **Gl.7.3-21**, **Gl. 7.3-22** und **Gl. 7.3-23** für die Integer-Transformation:

$$\underline{B}' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

Eingesetzt:

$$\underline{W}' = \begin{pmatrix} 140 & -1 & -6 & 7 \\ -19 & -39 & 7 & -92 \\ 22 & 17 & 8 & 31 \\ -27 & -32 & -59 & -21 \end{pmatrix}$$

Die Integer-Transformation $\underline{Y}_{Int}$ lässt sich mit Gleichung 7.35 wie folgt errechnen:

$$\underline{Y}_{Int} = \underline{W}' \otimes \underline{C}'$$

wobei $\underline{C}'$ sich aus Gleichung 7.36 wie folgt annähern lässt:

$$\underline{C}' \approx \begin{pmatrix} 0.25 & 0.16 & 0.25 & 0.16 \\ 0.16 & 0.1 & 0.16 & 0.1 \\ 0.25 & 0.16 & 0.25 & 0.16 \\ 0.16 & 0.1 & 0.16 & 0.1 \end{pmatrix}$$

Das Symbol $\otimes$ gibt wiederum die skalare Multiplikation an. Damit ergibt $\underline{Y}_{INT}$ zu:

$$\underline{Y}_{INT} = \begin{pmatrix} 35 & -0.16 & -1.5 & 1.11 \\ -3 & -3.9 & 1.11 & -9.2 \\ 5.5 & 2.69 & 2 & 4.90 \\ -4.27 & -3.2 & -9.33 & -2.1 \end{pmatrix}$$

Zur Kontrolle: Die *core*-Transformationsmatrix $\underline{W}'$ enthält tatsächlich nur ganze Zahlenwerte.

Im Ergebnis ist erkennbar, dass sich viele Koeffizienten sehr ähnlich sind, andere relativ verschieden sind. Der DC-Koeffizient ist in beiden Fällen gleich.

Literaturverzeichnis

[1.1] Poynton, Ch.: Digital Video and HDTV - Algorithms and Interfaces, Morgan Kaufmann Publishers, 2003

[1.2] Hauske, G.: Systemtheorie der visuellen Wahrnehmung, Teubner 1994

[1.3] Hentschel, H.-J.: Licht und Beleuchtung, Hüthig 1987

[1.4] Kubitz, P. P.: Der Traum vom Sehen, Verlag der Kunst 1997

[1.5] Lang, H.: Farbmetrik und Farbfernsehen, Oldenbourg 1978

[1.6] Smith, W. J.: Modern Optical Engineering, McGraw Hill 2000

[1.7] Mäusl, R.: Fernsehtechnik - Vom Studiosignal zum DVB-Sendesignal, Hüthig-Verlag 2002

[1.8] Schmidt, U.: Professionelle Videotechnik, Springer-Verlag 2000

[1.9] Bonse, Th.: Zur Konzeption einer visuell angepassten Beschreibung und Darstellung von Bewegtbildern, Shaker Verlag 1996

[1.10] Kelly, D. H.: Motion and Vision. II. Stabilized Spatio-Temporal Threshold Surface, Journal of Optical Society of America., Vol 69, No. 10, pp. 1340-1349, 1979

[1.11] Watanabe, A.; Sakata, H.; Isona, H.: Chromatic Spatial Sine-Wave Responses of the Human Visual System, NHK Laboratories Notes Ser. No. 198, 1976

[1.12] EBU Document Tech. 3213: EBU Standard for Chromaticity Tolerances for Studio Monitors, 1975 reissued 1981

[1.13] Society of Motion Picture and Television Engineers (SMPTE): RP 145 SMPTE C Color Monitor Colorimetry updated 1999

[2.1] Mäusl, R.: siehe [1.7]

[2.2] Schmidt, U.: siehe [1.8]

[2.3] Reimers, U.: Digital Video Broadcasting - The International Standard for Digital Television, Springer-Verlag 2001

[2.4] Poynton, Ch.: siehe [1.1]

[2.5] ITU-R BT.601-4: Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios, 1994

[2.6] ITU-R BT.656-3: Interfaces for Digital Component Video Signals in 525-Line and 625-Line Television Systems Operating at the 4:2:2 Level of Recommendation ITU-R BT.601 (Part A), 1995

[2.7] Watkinson, J.: The Art of Digital Audio, Focal Press Oxford, 1994

[2.8] ITU-R BT.1120: Digital Interfaces for 1125/60/2:1 and 1250/50/2:1 HDTV Studio Signals

[2.9] Digital Display Working Group: Digital Visual Interface - DVI, 1999

[2.10] Ottenbreit, M.: SDI & Embedded Audio, FKT Jg. 57, Heft 10/03, S. 466ff, Hüthig-Verlag 2003

[3.1] Mäusl, R.: siehe [1.7]

[3.2] Schmidt, U.: siehe [1.8]

[3.3] Reimers, U.: siehe [2.3]

[3.4] Ohm, J.-R.: Digitale Bildcodierung - Repräsentation, Kompression und Übertragung von Bildsignalen, Springer-Verlag 1995

[3.5] Symes, P.: Video Compression Demystified - How Compression Really Works; DCT and wavelets; MPEG & DV, MCGraw-Hill 2001

[3.6] ITU-T T.81: Information Technology – Coded Representation of Picture and Audio Information – Progressive Bi-level Image Compression, 1992

[3.7] ISO/IEC 10918-1: *identisch mit* [3.6] 1993

[3.8] Lohscheller, H.: Subjectively Adapted Image Communication System, IEEE Transactions on Communication, COM-32(12), S. 1316-22, 1984

[3.9] Hilton, M. L.; Jawerth, B. D.; Sengupat, A.: Compression of Still and Moving Images with Wavelets, Multimedia Systems, Vol. 2, No. 3, 1994

[3.10] Edwards, T.: Discrete Wavelet Transforms: Theory and Implementation Stanford Unisversity, September 1991

[3.11] Daubechies, I.: Orthonormal Bases of Compactly Supported Wavelets, Comm. Pure Appl. Math., 41, S. 909–996, 1988

[3.12] Said, A; Pearlman, W. A.: A New Fast and Efficient Image Codec Based on Set Partitioning in Hierar-chical Trees, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, 1996

[3.13] Shapiro, J. M.: Embedded Image Coding Using Zerotrees of Wavelet Coefficients, IEEE Trans. Signal Proc. 41(12), S. 3445–3462, 1993

[3.14] ISO/IEC 15444-1: Information technology - JPEG 2000 image coding system Part 1: Core coding system, 2000

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

[3.16] Fisher, Y. (ed.): Fractal Image Compression - Theory and Application, Springer-Verlag 1994

[4.1] Mäusl, R.: siehe [1.7]

[4.2] Ohm, J.-R.: siehe [3.4]

[4.3] Poynton, Ch.: siehe [1.1]

[4.4] Reimers, U.: siehe [2.3]

[4.5] Schmidt, U.: siehe [1.8]

[4.6] Schönfelder, H.: Fernsehtechnik im Wandel, Springer-Verlag 1996

[4.7] Symes, P.: siehe [3.5]

[4.8] Wendland, B.; Schröder, H.: Fernsehtechnik, Band II: Systeme und Komponenten zur Farbbildübertragung, Hüthig-Verlag 1991

[4.9] Ziemer, A.: Digitales Fernsehen, Hüthig-Verlag 2002

[4.10] Hartwig, S.; Endemann, W.: Digitale Bildcodierung, Sonderdruck aus Fernseh- und Kinotechnik 01/92 bis 01/93, Hüthig-Verlag 1993

[4.11] Fell-Bosenbeck, F.: DV-Kompression - Grundlagen und Anwendungen, Fernseh- und Kinotechnik, 53. Jahrgang, Nr. 6, 1999, S. 336-345

[4.12] SMPTE 314M: Television - Data Structure for DV-Based Audio, Data and Compressed Video - 25 and 50 Mb/s, 1999

[4.13] IEC 61834: Helical-Scan Digital Video Cassette Recording System Using 6,35 mm Magnetic Tape for Consumer User, 1999

[4.14] Wiswell, D.: Panasonic DVCPRO - from DV to HD http://www.panasonic.com/broadcast

[4.15] Cafforio, C.; Rocca, F.: Methods for Measuring Small Displacements of Television Images, IEEE Transactions on Information Theory, No.5, 1976

[4.16] Sikora, T.: MPEG Digital Video-Coding Standards, IEEE Signal Processing Magazine, September 1997, pp. 82-100 1997

[4.17] Hollmann, Th.: Zur Gewinnung von Bewegungsinformation in einer Kamera mit hoher Bildaufnahmefrequenz, Shaker Verlag 1994

[5.1] ITU-T Recommendation H.120: Codec for Videoconferencing Using Primary Digital Group Transmission, 1. Version: 1984, 2. Version: 1988

[5.2] ITU-T Recommendation H.261: Video Codec for Audiovisual Service at p x 64 kbits, 1. Vorversionen: 1988, Standardisiert: 1991, Letzte Erweiterungen: 1993

[5.3] ISO/IEC 11172: Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1,5 Mbit/s (*MPEG-1*) 1992

[5.4] Heyna, A.; Briede, M.; Schmidt, U. (Hrsg.): Datenformate im Medienbereich, Fachbuchverlag Carl Hanser Verlag, Leipzig, 2003

[5.5] Symes, P.: siehe [3.5]

[5.6] Ohm, J.-R.: siehe [3.4]

[5.7] Poynton, Ch.: siehe [1.1]

[5.8] ISO/IEC 13818: Information Technology - Generic Coding of Moving Pictures and Associated Audio Information (*MPEG-2*); Part 1: Systems, Part 2: Video 1994

[5.9] Reimers, U.: siehe [2.3]

[5.10] Mäusl, R.: siehe [1.7]

[6.1] International Organization for Standardization ISO 9660: 1988 Information processing – Volume and file structure of CD-ROM for information interchange, 1988

[6.2] International Organization for Standardization ISO/IEC 10149: 1995 (Yellow-Book) Information technology – Data interchange on read-only 120 mm optical data disks (CD-ROM), 1995

[6.3] Taylor, J.: DVD Demystified, Second Edition, McGraw-Hill, 2001

[6.4] LaBarge, R.: DVD Authoring & Production, CMP books, 2001

[6.5] European Telecommunications Standards Institute (ETSI) Norm ETS 300 468 (V1.3.1): Digital Video Broadcasting (DVB) Specification for Service Information (SI) in DVB systems (DVB-SI), 1998

[6.6] Mäusl, R.: siehe [1.7]

[6.7] Reimers, U.: siehe [2.3]

[6.8] Symes, P.: siehe [3.5]

[6.9] Ziemer, A.: siehe [4.9]

[6.10] European Telecommunications Standards Institute (ETSI) Norm EN 300 421 (V1.1.2): Digital Video Broadcasting (DVB) Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services, 1997

[6.11] Sweeney, P.: Codierung zur Fehlererkennung und Fehlerkorrektur, Hanser Verlag 1992

[6.12] Forney, G. D.: Burst-Correcting Codes for the Classic Bursty Channel, IEEE Transactions on Communication Technology – COM-10 No. 5, 10/1971, S. 772ff

[6.13] European Telecommunications Standards Institute (ETSI) Norm EN 300 429 (V1.2.1): Digital Video Broadcasting (DVB) Framing Structure, Channel Coding and Modulation for Cable Systems (DVB-C), 1998

[6.14] European Telecommunications Standards Institute (ETSI) Norm EN 300 744 (V1.1.2): Digital Video Broadcasting (DVB) Framing Structure,

Channel Coding and Modulation for Digital Terrestrial Television (DVB-T), 1997

[7.1] Jack, Keith: Video Demystified – A Handbook for the Digital Enineer LLH Technology Publishing – 3rd Edition, 2001

[7.2] International Telecommunication Union Line Transmission of Non-Telephone Signals (ITU-T): Video Coding for Low Bit Rate Communication Recommendation H.263, Version 1, 12/1995

[7.3] International Telecommunication Union Line Transmission of Non-Telephone Signals (ITU-T): Video Coding for Low Bit Rate Communication Recommendation H.263, Version 2 (H.263+), 01/1998

[7.4] International Telecommunication Union Line Transmission of Non-Telephone Signals (ITU-T): Video Coding for Low Bit Rate Communication Recommendation H.263, Version 3 (H.263++), 11/2000

[7.5] International Telecommunication Union Series H: Audiovisual And Multimedia Systems Infrastructure of Audiovisual Services – Coding of Moving Video Video Coding for Low Bit Rate Communication Recommendation H.263, Annex X, 04/2001

[7.6] International Organization for Standardization: ISO/IEC 14496 Information Technology – Coding of Audio-Visual Objects, 2000, Part 1 - 6

[7.7] Sikora, T.; Makai, B.: Shape-Adaptive DCT for Generic Coding of Video" IEEE Transcations on Circuits and Systems for Video Technology Bd. 5, Nr. 1, S. 59-62 02/1995

[7.8] International Organization for Standardization: ISO/IEC 14496-2 Information Technology – Coding of Audio-Visual Objects Part 2: Visual, Amendment 1: Studio Profile 2002

[7.9] International Organization for Standardization: ISO/IEC 14496-2 Information Technology – Coding of Audio-Visual Objects Part 2: Visual, Amendment 2: Streaming Video Profile, 2002

[7.10] Waggoner, B.: Compression for Great Digital Video, CMP Books 2002

[7.11] Pereira, F.; Ebrahimi, T.: The MPEG-4 Book, IMSC Press Multimedia Series 2002

[7.12] Symes, P.: siehe [3.5]

[7.13] International Telecommunication Union Series H: Audiovisual And Multimedia Systems Infrastructure of Audiovisual Services – Coding of Moving Video Advanced Video Coding for Generic Audiovisual Services Recommendation H.264, 05/2003 (Prepublish)

[7.14] Marpe, D.: H.264 / MPEG-4 AVC: The New Video Coding Standard Beitrag zum Wiesbadener *Medien Symposium* "Codierung audiovisueller Daten", 29.10. – 30.10.2003

[7.15] Wiegand, Th.; Sullivan, G. J.; Bjoentegaard, G.; Luthra, A.; u .a.: Overview of the H.264/AVC Video Coding Standard IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7 pp. 560-576, July 2003

[7.16] Schäfer, R.; Wiegand, Th.; Schwarz, H.: The Emerging H.264/AVC Standard EBU Technical Review, Januar 2003

[7.17] Schäfer, R.: Videocodierung für Fernsehen, Bildkommunikation und Multimedia Beitrag zum Wiesbadener *Medien Symposium* “Codierung audiovisueller Daten”, 29.10. – 30.10.2003

[7.18] Heyna, A.; Briede, M.; Schmidt, U. (Hrsg.): siehe [5.4]

Index