

Diplomarbeit

„H.264 – Der neue Standard in der Videokompression“

Ausgeführt zum Zweck der Erlangung des akademischen Grades eines
Dipl.-Ing. (FH) Telekommunikation und Medien
am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von
Dipl.-Ing. (FH) Klemens Huber

Zweitbegutachtung von
Prof. Dr. Jakob Wassermann

ausgeführt von
Iris Etter
tm0210038027

St. Pölten, am 4. September 2006

Unterschrift:

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland, noch im Ausland einem Begutachter / einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter beurteilten Arbeit überein.

.....
Ort, Datum

.....
Unterschrift

Kurzfassung

Die vorliegende Arbeit setzt sich mit dem jüngsten Standard in der Videokompression, H.264 / MPEG-4 AVC auseinander. Zum besseren Verständnis werden die Grundlagen der Videokompression erklärt und ein Überblick über zuvor standardisierte Kompressionssysteme gegeben.

Neben einer detaillierten Darstellung der technischen Komponenten des H.264-Standards werden die Fragen der Lizenzierung sowie der Qualitätsbeurteilung bei Videokompressionsstandards behandelt. Eine Reihe von Beispielen praktischer Implementierungen des Standards aus den unterschiedlichen Bereichen gibt einen Eindruck über seinen bisherigen und zukünftig zu erwartenden Erfolg.

Eine praktische Testreihe zeigt abschließend die Leistungsfähigkeit zweier verschiedener softwarebasierter H.264-Implementationen im Vergleich zu einigen konkurrierenden Codecs.

Abstract

This diploma thesis deals with the most recent standardization effort in the field of digital video compression, H.264 / MPEG-4 AVC. For a better understanding of the subject matter, the basics of video compression are explained, and an overview of previously standardized video compression systems is given.

Apart from a detailed description of H.264's technical components, the questions of open standards licensing and video quality measurement are discussed. A number of examples of practical implementations give an impression of H.264's present and expected success. Finally, the standard's efficiency is demonstrated in a test series comparing two different H.264 implementations to some competing proprietary codecs.

Inhalt

Kurzfassung	3
1 Einleitung	7
2 Grundlagen der Videokompression	9
2.1 Wozu Kompression?	9
2.2 Codec-Entwicklung	9
2.3 Grundmodell eines Codecs	10
2.3.1 Prädiktion und Bewegungskompensation	12
2.3.2 I-, P- und B-Bilder	13
2.3.3 DCT	13
2.3.4 Quantisierung	14
2.3.5 Zick-Zack-Abtastung	14
2.3.6 Lauflängencodierung	15
2.3.7 Entropiecodierung	15
3 Standards zur Videokompression	16
3.1 Historischer Überblick	16
3.2 Kompressionsstandards vor H.264	17
3.2.1 H.120	17
3.2.2 H.261	17
3.2.3 MPEG-1 Teil 2	18
3.2.4 MPEG-2 Teil 2 (= H.262)	19
3.2.5 H.263	19
3.2.6 MPEG-4 Teil 2 (MPEG-4 Visual)	20
3.3 Übersicht Videokompressions-Standards	22
4 H.264	23
4.1 Technische Beschreibung	24
4.1.1 Ganz neu: Örtliche Prädiktion (Intra-Prädiktion) von Makroblocks	26
4.1.2 Kleinere Sub-Blocks für bessere Bewegungskompensation	27
4.1.3 Bewegungskompensation mit Genauigkeit von 1/4-Bildpunkt	28
4.1.4 Bewegungsvektoren über die Bildgrenze hinaus	30
4.1.5 Prädiktion der Bewegungsvektoren	30
4.1.6 Mehrere Referenzbilder für die Bewegungsvorhersage	30
4.1.7 P_Skip-Blocks	30
4.1.8 B-Prädiktion	31
4.1.9 Behandlung von Halbbildern	31
4.1.10 Transformation	33
4.1.11 Quantisierung	34
4.1.12 Entropiecodierung	35
4.1.13 Deblocking-Filter in der Prädiktionsschleife	36
4.1.14 FMO – Flexible Anordnung von Makroblocks (flexible macroblock ordering)	38
4.1.15 ASO – Beliebige Anordnung der Slices (arbitrary slice order)	39
4.1.16 SP- und SI-Slices	39
4.1.17 Data Partitioned Slices (DP) und Redundant Slices (RS)	39
4.2 Network Abstraction Layer (NAL)	40
4.2.1 NAL-Units	40

4.2.2 NAL-Units im Bytestrom-Format.....	40
4.2.3 NAL-Units in paketorientierten Transportsystemen.....	41
4.2.4 Parameter-Sets	41
4.2.5 Access-Units und redundant codierte Bilder	41
4.2.6 Codierte Videosequenzen	42
4.3 Profile und Levels.....	42
4.4 Erweiterung 2004: FRExt (Fidelity Range Extensions)	44
4.4.1 8x8 Intra-Prädiktion, 8x8-Transformation.....	44
4.4.2 Weitere Verbesserungen in FRExt sind u.a.	44
4.5 Scalable Video Coding (SVC).....	45
4.5.1 Layered Coding	46
4.5.2 Zeitliche Skalierbarkeit.....	46
4.5.3 Örtliche Skalierbarkeit.....	47
4.5.4 SNR-Skalierbarkeit	48
4.6 Was kommt nach H.264?.....	49
5 Lizenzierung von Offenen Standards.....	51
5.1 Sonderfall: SMPTE 421M / VC-1.....	53
6 Qualität eines Videocodecs	54
6.1 Objektive Bewertung	54
6.1.1 Video Quality Experts Group.....	55
6.2 Subjektive Bewertung.....	56
7 H.264 in der Praxis	58
7.1 Vorgesehene Einsatzgebiete	59
7.2 Systemstandardisierungen zur Einbettung von H.264	59
7.3 Digitales Fernsehen.....	60
7.3.1 IPTV	60
7.3.2 Satellitenfernsehen	61
7.3.3 DVB-T	61
7.3.4 Mobiles Fernsehen	61
7.4 Speichermedien	62
7.5 Aufzeichnung	63
7.6 Kommunikation.....	63
7.7 Mobile Anwendungen	63
7.8 Computerbasierte Anwendungen	63
7.9 Andere Anwendungen	65
8 H.264-Codecs im Test	66
8.1 Testsequenzen.....	69
8.2 Getestete Codecs.....	70
8.2.1 Apple QuickTime H.264	70
8.2.2 x264	71
8.2.3 DivX 6.2.5.....	73
8.2.4 Windows Media Video 9 (WMV9).....	73
8.2.5 Sorenson Video 3 Pro (SVQ3)	73
8.3 Vorgangsweise bei den Tests.....	74

8.4 Testergebnisse im Einzelnen: PSNR, SSIM, VQM	75
8.4.1 Sequenz „City“	75
8.4.2 Sequenz „Rugby“	76
8.4.3 Sequenz „Susi“	77
8.4.4 Sequenz „Wind“	78
8.4.5 Sequenz „Zug“	79
8.5 Testergebnisse im Einzelnen: Blocking	80
8.6 Zusammenfassung der Testergebnisse	84
10 Abschließende Zusammenfassung und Reflexion	87
Anhang	89
Quellenverzeichnis	89
Abbildungsverzeichnis	96

1 Einleitung

Das digitale Videosignal stellt in der Welt der Datenübertragung die höchsten Anforderungen an Bandbreiten, egal auf welchem Übertragungsweg. Auch wenn die Entwicklung hin zu einem weltumspannenden Hochleistungs-Datennetz mit heute vielleicht noch kaum vorstellbaren Übertragungskapazitäten bereits begonnen hat, wird auch langfristig die Minimierung der Datenraten und damit die effizienteste Nutzung vorhandener Bandbreiten auf den verschiedensten Übertragungswegen eine zentrale Frage sein.

Das Ziel aller Ansätze der Videokompression ist daher eine Minimierung der zu übertragenden Daten und damit der benötigten Bandbreite bei gleichzeitiger Maximierung der Bildqualität und Stabilität der Übertragung.

H.264, auch bekannt als MPEG-4/AVC (Advanced Video Coding) oder MPEG-4/Teil¹⁰, ist der jüngste Standard in der digitalen Videokompression, der gemeinsam von der Video Coding Experts Group (VCEG) der ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) und der Moving Pictures Expert Group (MPEG) der ISO/IEC (International Organisation for Standardization/International Electrotechnical Commission) entwickelt wurde.

H.264 zeichnet sich gegenüber allen zuvor entwickelten Standards durch eine deutliche Effizienzsteigerung, aber auch durch seine vielfältigen Einsatzmöglichkeiten aus, die sich daraus ergeben, dass seine Leistungsfähigkeit von sehr niedrigen Bitraten und kleinen Bildauflösungen, die zum Beispiel für die Übertragung auf Mobiltelefonen geeignet sind, bis hin zu professionellen Anwendungen im High-Definition-Bereich konstant ist.

Seine hohe Effizienz, vielfältige Einsetzbarkeit und nicht zuletzt die Tatsache, dass es sich bei H.264 um einen offenen Standard handelt, ließen bereits bei seiner Veröffentlichung 2003 Experten vermuten, dass er „der Video-Standard der Zukunft“ werden könnte.

Hat H.264 nun, drei Jahre nach seiner Veröffentlichung, bereits den ihm prophezeiten Siegeszug angetreten? Kann er die in ihn gesetzten Erwartungen erfüllen?

Diese Arbeit möchte zunächst zeigen, in welchem Ausmaß und wodurch sich H.264 gegenüber seinen Vorgängern auszeichnet. Weiters wird untersucht, wie weit die

Implementierung des neuen Standards in den verschiedenen vorgesehenen Bereichen bis Mitte 2006 vorangeschritten ist. Eine Versuchsreihe soll außerdem die Leistungsfähigkeit von verschiedenen H.264-Implementationen im Vergleich zueinander und zu anderen Codecs deutlich machen.

Aus diesen Informationen soll sich abschließend beantworten lassen, welche Bedeutung der H.264-Standard heute gegenüber anderen Videokompressionssystemen einnimmt, und welche Rolle er in näherer Zukunft spielen wird.

2 Grundlagen der Videokompression

2.1 Wozu Kompression?

Ein unkomprimiertes digitales Videosignal in Standard-Fernsehauflösung (SDTV, Standard Definition Television) hat eine Bitrate von 270 Mbit/s (nach der Norm ITU-R BT.601 bei 4:2:2 Farb-Unterabtastung und 10 Bit Tiefe).

Erst durch eine effektive Kompression dieses Signals, welche die Datenrate und damit die benötigte Bandbreite um ein Vielfaches reduziert, macht es überhaupt Sinn, digitales Video auf Medien wie CD und DVD zu speichern, über Antenne, Satellit oder Kabel zu senden oder in weiträumigen Datennetzen zu übertragen.

2.2 Codec-Entwicklung

Jedes Video-Kompressionssystem oder Codec besteht aus einem Encoder, der die Daten codiert/komprimiert, und einem Decoder, der die Daten am anderen Ende wieder zum ursprünglichen Signal, oder der bestmöglichen Annäherung daran, zusammensetzt.

Offene Kompressionsstandards spezifizieren jeweils nur die Syntax des codierten Datenstroms, und die Anforderungen an den Decoder. Die Implementierung des Encoders ist aber nicht vorgeschrieben, und lässt den Entwicklern somit alle Möglichkeiten zur Optimierung offen.

Um einem Standard zu entsprechen, muss ein Encoder fähig sein, einen Bitstrom gemäß den Vorgaben zu erzeugen. Ein Decoder wiederum muss jeden standardkonformen Bitstrom decodieren können. Dieses Prinzip ermöglicht die Konkurrenz zwischen verschiedenen Anbietern hinsichtlich der Entwicklung von möglichst effizienten Codierungsverfahren und schafft gleichzeitig ein hohes Maß an Kompatibilität zwischen verschiedenen Anwendungen und Herstellern.

In Konkurrenz zu den offenen Standards stehen die so genannten „proprietären“ Codecs, die von einzelnen Unternehmen meist für spezifische Anwendungsbereiche entwickelt werden (z.B. Real Video oder Windows Media). Auf die Unterschiede zwischen standardkonformen und proprietären Codecs, besonders in rechtlicher Hinsicht, wird in einem eigenen Kapitel genauer eingegangen.

Die einzelnen Implementierungen der Standards sind in den meisten Fällen asymmetrisch gebaut: Der Decoder ist möglichst simpel gehalten, der Encoder um ein Vielfaches komplexer – er muss u.a. Bewegungen erkennen, optimale Bewegungsvektoren bestimmen, und je nach Bildinhalt die günstigste Codierungsart auswählen können. Zum Encoder gehört auch immer ein Puffer, der laufend den Kompressionsfaktor anpasst, um eine konstante Bitrate in der Ausgabe zu gewährleisten.

Auch nach der Veröffentlichung eines Standards können sich dessen Kompressionsraten durch verbesserte Hardware oder durch Optimierung des Codierungs-Prozesses noch deutlich steigern. MPEG-2 brauchte beispielsweise bei seiner Veröffentlichung 1995 für ein Fernsehsignal in guter Qualität noch 6 Mbit/s, inzwischen reichen für die gleiche Qualität 2 Mbit/s, wobei am Decoder nichts verändert werden muss.

Eine möglichst effiziente Datenreduktion ist allerdings nicht das einzige Kriterium, an dem man sich bei der Entwicklung von Kompressionssystemen orientiert. Bei bidirektionalen Anwendungen wie z.B. Videokonferenz sind kurze Verzögerungszeiten (Delays), extrem wichtig, eine detailgetreue Bildwiedergabe eher weniger. Im Bereich der Archivierung, im Broadcastbereich und bei der Videoaufzeichnung sind Delays in bestimmten Ausmaßen durchaus akzeptabel.

Die Robustheit gegenüber Übertragungsfehlern ist ein weiterer wichtiger Faktor, der z.B. bei der Datenübertragung über Funknetze oder über das Internet eine große Rolle spielt. Dann ist da natürlich noch die Kostenfrage: Je geringer die Komplexität des Codecs, desto billiger die Chip-Herstellung bei hardwarenahen Umsetzungen bzw. desto geringer der Speicher- und Rechenleistungsbedarf bei Software-Lösungen.

Diese Faktoren müssen je nach Einsatzgebiet gegeneinander abgewogen werden. Ideal ist natürlich ein Codec, der sich an die verschiedenen Anwendungsbereiche optimal anpassen lässt, und durchwegs konstante Leistungen bringt.

2.3 Grundmodell eines Codecs

Alle Standards zur Videokompression seit ITU-T H.261 folgen einem „block-basierten hybriden Modell“ ([18]). Hybrid bezieht sich dabei auf die Kombination von

Bildvorhersage zur Nutzung von zeitlichen statistischen Abhängigkeiten, und Transformationscodierung zur Nutzung von örtlichen statistischen Abhängigkeiten ([12]).

Weil das menschliche Auge für Farbinformationen weniger empfindlich ist als für Helligkeitswerte, gehen die meisten Standards grundsätzlich von einer 4:2:0 Farb-Unterabtastung aus, d.h. für die Chrominanz (Cr = Abweichung von Grau in Richtung Rot; Cb = Abweichung von Grau in Richtung Blau) werden in horizontaler und vertikaler Richtung jeweils nur halb so viele Werte verwendet wie für die Luminanz (Helligkeit, Y) – auf 4 Y-Werte kommen also jeweils nur ein Cr- und Cb-Wert.

Die einzelnen Verarbeitungsschritte im Encoder sehen folgendermaßen aus:

- Jedes Bild wird zur Verarbeitung in sog. Makroblocks à 16x16 Luminanzwerte bzw. 8x8 Chrominanzwerte unterteilt.
- Das Bild wird mit einem Referenzbild (einem zuvor codierten Bild) verglichen, und für jeden Makroblock wird der dazupassende Makroblock aus dem Referenzbild gefunden. Aus dem räumlichen Abstand zwischen den beiden wird ein Bewegungsvektor gebildet.
- Auf Basis dieses Bewegungsvektors entsteht eine bewegungskompensierte Bildvorhersage (Prädiktion)
- Die Prädiktion wird vom aktuellen Makroblock subtrahiert, es entsteht ein Differenzmakroblock.
- Die Differenz wird einer Transformation unterzogen. In den meisten Fällen einer Diskreten Kosinustransformation (DCT). Die Transformation wird üblicherweise auf 8x8-Blocks angewandt.
- Die daraus resultierenden Transformationskoeffizienten werden quantisiert.
- Danach werden die DCT-Koeffizienten neu angeordnet, meist durch Zick-Zack-Abtastung, und lauflängencodiert.
- Am Schluss werden die Koeffizienten, die Bewegungsvektoren und die jeweilige Header-Information für jeden Makroblock entropiecodiert.

Jeder dieser Schritte wird im Decoder dann in umgekehrter Reihenfolge wieder rückgängig gemacht, um das ursprüngliche Bild zu rekonstruieren. Bei der Quantisierung der Transformationskoeffizienten gehen jedoch Daten verloren, sodass das Endergebnis eine Verzerrung gegenüber dem Original aufweist, die umso größer ist, je größer die Quantisierungsstufen.

2.3.1 Prädiktion und Bewegungskompensation

Eine Teilaufgabe des Kompressionsalgorithmus ist es, zeitliche statistische Abhängigkeiten im Videosignal zu reduzieren, um die Daten für die weiteren Codierungsschritte optimal aufzubereiten. Dabei geht man von dem Gedanken aus, dass sich in einer Bildsequenz von einem Bild zum nächsten nur relativ wenig ändert.

Es ist daher nicht notwendig, den vollen Inhalt jedes einzelnen Bildes für sich zu codieren, sondern es genügt, die Differenz des Bildes, das gerade verarbeitet wird, zu einem bereits verarbeiteten Referenzbild (z.B. dem direkt vorhergehenden), zu übertragen. Die Rekonstruktion des Bildes erfolgt dann durch die Addition des Differenzwerts zum Referenzbild.



Original

Differenzbild ohne Bewegungs-
Kompensation

Differenzbild mit Bewegungs-
Kompensation (Prädiktionsfehler)

Abb. 2: Differenzbilder

Je kleiner die Differenz zum Referenzbild, desto weniger Daten braucht man, um das Bild zu repräsentieren, und somit desto effektiver die Kompression. Bei schnellen Bewegungen ergeben sich große Differenzen und damit höhere Datenraten. Um das Prinzip der Differenzbildung optimal zu nutzen, wird daher eine so genannte Bewegungskompensation eingesetzt: Aus den bereits codierten Bildern einer Sequenz werden nach bestimmten mathematischen Algorithmen Bewegungsvektoren berechnet, mit deren Hilfe eine Voraussage (Prädiktion) über das aktuelle Bild getroffen wird. Codiert wird dann nur der Prädiktionsfehler (engl. „Residual“, Rest).

2.3.2 I-, P- und B-Bilder

Nachteile bei einer ausschließlichen Differenzbild-Codierung sind die Fortpflanzung von einmal aufgetretenen Fehlern, und die erschwerte Möglichkeit eines Zugriffs auf einzelne Bilder. Aus diesem Grund werden so genannte GOPs (groups of pictures) gebildet, die sich aus unterschiedlich codierten Bildern zusammensetzen.

So genannte I-Bilder werden zur Gänze unabhängig von anderen Bildern codiert (intra-codiert). Sie sind auf jeden Fall am Anfang jeder neuen Bildfolge notwendig, und danach in regelmäßigen Abständen, um eine Fortpflanzung von Fehlern zu vermeiden und den Zugriff auf Einzelbilder zu ermöglichen.

Bilder, die als Prädiktionsfehler und damit in Abhängigkeit zu bereits verarbeiteten Bildern codiert werden, werden P-Bilder genannt (Predictive). Weiters gibt es die Möglichkeit einer bidirektionalen Prädiktion. Dabei wird zusätzlich zu einem vorhergehenden auch ein nachfolgendes Bild als Referenz für die Bewegungsvorhersage verwendet. Auf diese Weise codierte Bilder heißen B-Bilder.

Eine GOP setzt sich typischerweise aus einem I-Bild und mehreren P- und B-Bildern zusammen.

2.3.3 DCT

Als nächster Schritt wird auf I-, P- und B-Bilder eine Transformation angewandt, um örtliche statistische Abhängigkeiten zu nutzen. Die bei der Videokompression am häufigsten verwendete Art ist die Diskrete Kosinustransformation (DCT).

Die Verarbeitung erfolgt dabei in 8x8-Blocks. Der Bildinhalt jedes Blocks wird durch folgende Berechnungsvorschrift in seine Frequenzanteile zerlegt:

$$X[k] = C_0 \cdot \sqrt{\frac{2}{N}} \cdot \sum_{n=0}^{N-1} x[n] \cdot \cos \left[(2 \cdot n + 1) \cdot \frac{k \cdot \pi}{2 \cdot N} \right], \quad \text{mit } C_0 = \begin{cases} \frac{1}{\sqrt{2}} & \text{für } k = 0 \\ 1 & \text{für } k \neq 0 \end{cases}$$

Zur Rekonstruktion wird im Decoder die inverse Transformation (IDCT) durchgeführt, was im Prinzip exakt möglich ist; Durch unterschiedliche Implementierungen an Encoder und Decoder kann es dabei allerdings zu Rundungsfehlern kommen.

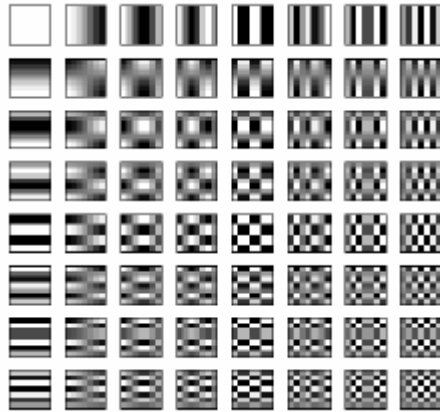


Abb. 3: Durch DCT wird der Bildinhalt eines 8x8 Luminanzblocks in diskrete Frequenzen umgewandelt. Niedrigste Frequenz im linken oberen Eck. Die horizontale Frequenz nimmt nach rechts zu, die vertikale nach unten.

2.3.4 Quantisierung

Als nächstes werden die Transformationskoeffizienten je nach ihrer Bedeutung für die menschliche Wahrnehmung gewichtet. Die Anteile an hohen Frequenzen, die ohnehin typischerweise sehr niedrig sind, sind dabei kaum relevant, und können oft auf Nullwerte gerundet werden.

Die Quantisierung ist der einzige Schritt in der Codierung, durch den der Bildinhalt verändert und so ein Informationsverlust und damit eine Verzerrung im rekonstruierten Bild eingeführt wird.

Durch Veränderung der Quantisierungswerte lässt sich der Kompressionsfaktor (und der damit einhergehende Qualitätsverlust) steuern.

2.3.5 Zick-Zack-Abtastung

Um die Koeffizienten nach absteigenden Werten zu reihen und damit möglichst viele hintereinander folgende Nullwerte zu erreichen (als Vorbereitung für die folgende

Lauf­längen­codierung), werden sie in einem ganz bestimmten Muster ausgelesen – typischerweise in einem Zick-Zack-Muster.

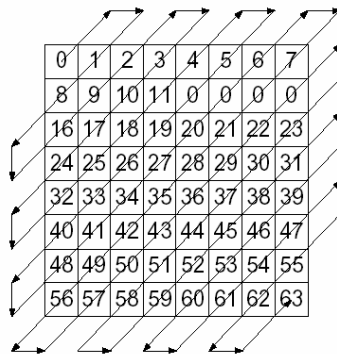


Abb. 4: Zick-Zack-Abtastung

2.3.6 Lauf­längen­codierung

Lauf­längen­codierung (RLC, run length coding) bedeutet, dass eine Anzahl von gleichen Symbolen (Codewörtern), die hintereinander auftreten, in einem einzigen Symbol codiert werden. Im Fall der gewichteten DCT-Koeffizienten werden die hintereinander folgenden Nullwerte gemeinsam mit dem nächsten Nicht-Null-Wert zu einem Symbol zusammengefasst.

2.3.7 Entropiecodierung

Im abschließenden Schritt werden die Daten durch Entropiecodierung mit variabler Wortlänge (VLC, variable length coding) weiter komprimiert. Dabei wird die statistische Verteilung von Symbolen innerhalb der Datenstruktur ausgenutzt, indem häufiger vorkommende Symbole mit kürzeren Wortlängen codiert werden.

3 Standards zur Videokompression

3.1 Historischer Überblick

Ende der 1980er Jahre begannen zwei verschiedene Expertengruppen internationale Standards zur Codierung bzw. Kompression von digitalen Videosignalen zu entwickeln. Das war einerseits die Video Coding Experts Group (VCEG), ein Gremium innerhalb der ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) und andererseits die Moving Pictures Experts Group (MPEG), ein Komitee der ISO und der IEC (International Organization for Standardization, International Electrotechnical Commission).

Die Ziele beider Gremien waren hinsichtlich der Anwendungsbereiche vorerst unterschiedlich – die VCEG konzentrierte sich auf Bildtelefonie und Videokonferenzsysteme, und ging daher von sehr niedrigen Bitraten und Auflösungen aus. Das erste Standardisierungspaket der MPEG (MPEG-1, 1993) war dagegen für die Speicherung von audiovisuellen Inhalten auf CD-ROMs bzw. Video-CDs ausgelegt.

In weiterer Folge bauten die beiden Gruppen jeweils auf den Entwicklungen der anderen auf und arbeiteten teilweise sehr eng zusammen. 1994 wurde erstmals ein Standard von beiden Gremien im selben Wortlaut verabschiedet: Unter dem Namen H.262 von der ITU-T, und als MPEG-2 Teil 2 (Video) von ISO/IEC.

Zur Entwicklung des derzeit jüngsten internationalen Standards, H.264/MPEG-4 Teil 10, schloss sich die MPEG in der letzten Entwicklungsphase (2001-2003) direkt der VCEG an, und es wurde das gemeinsame Joint Video Team (JVT) gegründet.

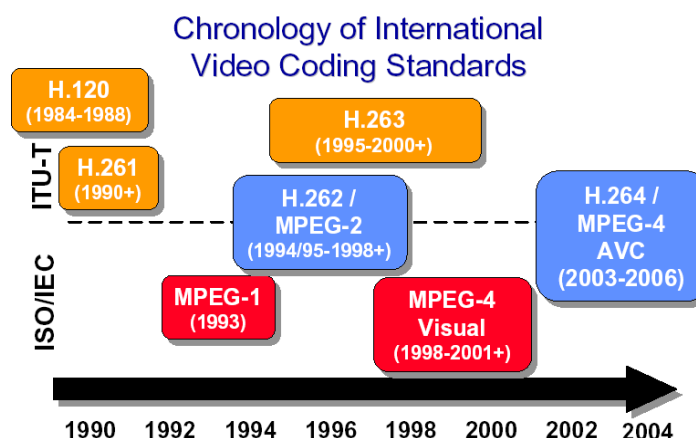


Abb. 1: Chronologie der internationalen Videokompressionsstandards

3.2 Kompressionsstandards vor H.264

3.2.1 H.120

Der erste Standardisierungsversuch in der digitalen Videocodierung war ITU-T H.120 im Jahr 1984. Obwohl dieser Standard kaum angewandt wurde und heute keine praktische Bedeutung mehr hat, enthält er schon einige Elemente, die in späteren Videokompressionssystemen vorkommen, wie z.B. skalare Quantisierung, Entropiecodierung, und in der zweiten Version (1988) auch eine Art der Bewegungskompensation. H.120 war ausschließlich für die Bitraten 2048 kbit/s (für PAL-Auflösung, 625 Zeilen, 50 Halbbilder/s) und 1544 kbit/s (NTSC, 525 Zeilen, 60 Halbbilder/s) ausgelegt.

3.2.2 H.261

ITU-T H.261 wurde 1990 verabschiedet und war der erste Kompressionsstandard, der sich international durchsetzte. Er gilt als Basis für alle seither entwickelten Standards, inklusive der MPEG-Familie. Seine einzelnen Elemente wurden in allen nachfolgenden Entwicklungen jeweils weitgehend übernommen, leicht verändert bzw. verfeinert und mit neuen Elementen ergänzt.

- 16x16 Makroblock Bewegungskompensation
- I- und P- Bilder
- 8x8 DCT
- Skalare Quantisierung
- Zick-Zack Scan
- Lauflängencodierung
- Entropiecodierung mit VLC

Auch der Prozess der Entwicklung bis zur Standardisierung von H.261, der durch die Zusammenarbeit vieler Experten in einer Gruppe gekennzeichnet war, wurde für alle weiteren Standardisierungsbemühungen im Videobereich beibehalten.

Ein Element von H.261, das nicht in die unmittelbar nachfolgenden Standards übernommen wurde, aber bei H.264 auftaucht, ist ein Deblocking-Filter, der Block-

Artefakte vermindert, die bei starker Kompression häufig auftreten und die subjektive Qualitätswahrnehmung besonders stark beeinträchtigen.

Der Anwendungsbereich, für den H.261 spezifiziert wurde, ist Bildtelefonie und Videokonferenz über ISDN-Leitungen, die Datenraten von 64 kbit/s bzw. ein Vielfaches davon ermöglichen. Der Kompressionsalgorithmus ist optimiert für Bitraten zwischen 40 kbit/s und 2 Mbit/s und die Bildauflösungen CIF (Common Intermediate Format, 352×288 Pixel) und QCIF (Quarter CIF, 176×144 Pixel) bei einer 4:2:0-Abtastung.

Inzwischen wurde H.261 weitgehend durch H.263 ersetzt, und ist in modernen Videokonferenz-Systemen nur noch aus Gründen der Rückwärts-Kompatibilität implementiert.

3.2.3 MPEG-1 Teil 2

Das MPEG-1-Paket war 1993 der erste Standardisierungserfolg der Moving Pictures Expert Group von ISO und IEC. Alle MPEG-Standards gliedern sich in mehrere Teile, wobei Teil 2 jeweils die Video-Spezifikationen enthält und deshalb oft auch mit dem Zusatz „Visual“ bezeichnet wird. (Andere Teile beziehen sich auf Audiocodierung, Synchronisation und Einbettung von Audio- und Videodaten in Systemströme, Referenzsoftware, Testprozeduren etc.)

Das Ergebnis von MPEG-1 Teil 2 war eine bessere Bildqualität als H.261 bei Bitraten über 1 Mbit/s, und eine mit VHS vergleichbare Bildqualität im SIF-Format (252x288 Pixel bei PAL) bei Datenraten zwischen 1 und 2 Mbit/s, wodurch eine Speicherung von Videodaten auf und Abspielung von CD-ROMs bzw. Video-CDs möglich wurde. Eine große Schwäche des Standards ist, dass er keine Halbbild-Verarbeitung (interlace) unterstützt.

Inzwischen wurde MPEG-1 weitgehend durch MPEG-2 abgelöst, ist aber immer noch in Verwendung, einerseits weil Video-CDs im asiatischen Raum noch relativ weit verbreitet sind, andererseits weil alle MPEG-2-Decoder verpflichtend auch MPEG-1 decodieren können müssen (dieses Prinzip der Rückwärts-Kompatibilität in der MPEG-Familie wurde auch bei MPEG-4 Teil 2 fortgesetzt; H.264 ist davon allerdings nicht betroffen).

Aufbauend auf den Funktionen von H.261 wurden u.a. folgende Verbesserungen eingeführt:

- Bidirektionale Bewegungsvorhersage (auch B-Bilder zusätzlich zu I- und P-Bildern)
- halb-pixel-genaue Bewegungsvorhersage (bisher nur pixelgenau)
- Bilder werden zur Verarbeitung zusätzlich in „Slices“ geteilt. Ein Slice ist eine Gruppe von Makroblocks innerhalb eines Bildes, die unabhängig von den anderen Teilen des Bildes verarbeitet werden kann. Die Entscheidung über I-, P-, oder B- Codierung erfolgt sliceweise.

3.2.4 MPEG-2 Teil 2 (= H.262)

MPEG-2 ist heute der am weitesten verbreitete und auch wirtschaftlich erfolgreichste Videokompressions-Standard. Er wurde für die Speicherung und Übertragung von Video in Broadcast-Qualität entwickelt und ist in diesen Bereichen – DVD-Video und DVB (Digital Video Broadcast, digitales Fernsehen) über Satellit, Kabel und terrestrisch - DER Standard schlechthin.

Seit der Standardisierung 1994 konnte die Leistungsfähigkeit von MPEG-2-Encodern wesentlich verbessert werden. Ursprünglich ging man von einer CCIR 601 Auflösung – 720 x 576 Pixel (PAL) bzw. 720 x 480 Pixel (NTSC) mit Datenraten zwischen 3 und 10 Mbit/s aus. Inzwischen kann MPEG-2 SDTV (Standard Definition Television) in guter Qualität bei ca. 2 Mbit/s bieten.

Die MPEG sah ursprünglich für HDTV (High Definition Television)-Anwendungen einen neuer Standard mit höherer Kompressionsleistung vor (MPEG-3). MPEG-2 erwies sich aber auch für solche hohen Datenraten als leistungsfähig genug – im digitalen Fernsehen werden heute MPEG-2-codierte HDTV-Signale typischerweise mit Datenraten zwischen 10 und 20 Mbit/s übertragen.

3.2.5 H.263

Der unmittelbare Vorgänger von H.264 aus der ITU-T-Serie ist heute der am weitesten verbreitete Kompressionsstandard für Videokonferenz-Anwendungen. Bei seiner Einführung 1995 überbot er alle anderen Standards hinsichtlich der Bildqualität bei allen

Bitraten (ausgenommen interlaced Video, vgl. [34]). Bei sehr niedrigen Datenraten bietet er eine doppelt so effiziente Kompression wie seine Vorgänger.

Die Verbesserungen gegenüber MPEG-2 Visual wurden u.a. durch folgende Neuerungen erreicht:

- längere Bewegungsvektoren, die auch über den Bildrand hinaus gehen können
- Prädiktion von Bewegungsvektoren mittels Median-Filterung der Nachbarvektoren
- verbesserte Codierung der Transformationskoeffizienten
- optimierte Code-Tabellen
- optionale Unterteilung der Makroblocks in 8x8-Blöcke für die Bewegungskompensation

H.263+

Die erste Erweiterung von H.263 aus dem Jahr 1998 sieht einen Deblocking-Filter in der Prädiktionsschleife vor, der die subjektive Bildqualität und damit auch die Effizienz enorm verbessert und als ein wichtiges Element von H.264 übernommen wurde. Außerdem wird bei intra-codierten Blöcken die Vorhersage jener AC-Koeffizienten eingeführt, die nur horizontale oder vertikale Frequenzinformationen enthalten. Dieses Prinzip wurde bei H.264 erweitert zur Intra-Prädiktion von ganzen Makroblocks. Andere Verbesserungen betreffen die Robustheit gegenüber Übertragungsfehlern.

H.263++

Eine zweite Erweiterung führte 2000 erstmals die so genannte Langzeit-Prädiktion ein. Ein Referenzbild muss dabei kein unmittelbarer zeitlicher Vorgänger mehr sein, sondern kann in der Bildfolge auch schon weiter zurückliegen. Dieses Prinzip bildete die Basis für das H.26L-Projekt, aus dem sich später H.264 entwickelte.

3.2.6 MPEG-4 Teil 2 (MPEG-4 Visual)

Nach MPEG-2 war bei der Entwicklung eines neuen Kompressionsstandards die Effizienzsteigerung nur eines von mehreren Zielen. Mit MPEG-4 wollte man vor allem einen Standard mit einem Maximum an Flexibilität für alle Anwendungsbereiche von Telefonie über Internet bis zu Broadcast und Speicherung, inklusive Interaktivität und Einbeziehung von synthetischen Inhalten, erreichen. Die MPEG-4 Visual Spezifikationen sind daher um ein Vielfaches umfangreicher und komplexer als die aller Vorgänger.

Neben den Kompressionselementen für klassische rechteckige Videosequenzen gibt es u.a. die Möglichkeit, beliebig geformte Video-Objekte zu codieren. Generell wird ein objektorientierter Ansatz ermöglicht, bei dem eine Videosequenz aus verschiedenen unabhängig voneinander codierten Objektebenen (Video Objekt Planes, VOP) zusammengesetzt wird. Dadurch kann für jede VOP, die auch synthetisch erzeugte Bilder enthalten kann, die effektivste Kompressionsmethode eingesetzt werden. Gleichzeitig können die einzelnen Objekte unabhängig voneinander bearbeitet und wieder verwendet oder mit Interaktionsmöglichkeiten in Verbindung gebracht werden.

Der Teil von MPEG-4, der die klassische Videokompression betrifft, setzt unmittelbar auf den H.263-Standard auf und unterscheidet sich davon nur in wenigen Punkten. Die Bewegungskompensation ist auf $\frac{1}{4}$ -Bildpunkt-Genauigkeit verbessert. Außerdem gibt es den neuen Modus der „globalen Bewegungskompensation“, bei dem erstmals auch nicht-translatorische Bewegungen wie Zooms oder Rotationen berücksichtigt werden können (die Entscheidung zwischen lokaler und globaler Bewegungskompensation erfolgt auf Makroblockebene).

Für B-Bilder gibt es vier Prädiktionsmodi: vorwärts, rückwärts, kombiniert oder Direkt-Modus. Ein Deblocking-Filter ist anders als bei H.263 nicht innerhalb der Prädiktionsschleife, sondern erst nach der Decodierung und Rekonstruktion, also quasi nur als Nachbearbeitungsschritt, vorgesehen, und kann auf diese Weise nicht zu einer Verbesserung der Prädiktion beitragen.

MPEG-4 sieht auch eine Möglichkeit der Codierung von Hintergrundobjekten als skalierbare Standbilder vor, wobei zur Transformation keine DCT, sondern ein waveletbasierter Algorithmus eingesetzt wird.

Ein Zusammenspiel mehrerer Ursachen hat verhindert, dass MPEG-4 Visual an den Erfolg seines Vorgängers MPEG-2 anschließen konnte. Ein Grund dafür war die Weiterentwicklung von MPEG-2 selbst. Seine verbesserte Kompressionsleistung machte die Effizienzsteigerung, die mit MPEG-4 erreicht werden konnte, zu klein, um eine Umrüstung von ganzen Broadcast-Systemen zu rechtfertigen.

Außerdem schafften der große Umfang und die vielen Möglichkeiten des Standards bei den meisten Entwicklern eher Verwirrung als Orientierung.

Ein nicht zu unterschätzendes Problem bei der Verbreitung von MPEG-4 Visual war die Lizenzierungspolitik. Wie auch bei MPEG-2 werden die Lizenzen durch einen Pool,

MPEG-LA, verwaltet. Die Gestaltung der Lizenzierungsbedingungen für MPEG-4 erschienen für viele potenzielle Lizenznehmer so ungünstig, dass sie in eine öffentliche Auseinandersetzung mit der MPEG LA-Gruppe traten. Diese Problematik wird in einem eigenen Kapitel detailliert behandelt.

Nicht zuletzt ist die rasche Nachfolge des H.264-Standards, der im Gegensatz zu MPEG-4 Visual möglichst simpel gehalten und sehr stark auf Kompressionseffizienz und Fehlerresistenz zielgerichtet ist, ein Faktor, der gegen einen breiten Erfolg von MPEG-4 als „universeller Videocodec“ spricht.

3.3 Übersicht Videokompressions-Standards

Name	Erscheinungsjahr	Bemerkung
H.261	1990	Erster erfolgreicher Kompressionsstandard für Bildtelefonie und Videokonferenz über ISDN-Leitungen (niedrige Bitraten)
MPEG-1 Teil 2	1993	Kompression von Vollbild-Videos im SIF-Format (252x288), u.a. für Video-CDs. (Bitraten 1-2 Mbit/s) Zum Audio-Teil von MPEG-1 gehört MP3 (MPEG-1 Layer 3).
MPEG-2 Teil 2 (Visual)	1994/95	Video in SD-Fernsehqualität, auch Halbbildmodus. Wird für DVD-Videos und DVB verwendet. (Bitraten bis 15 Mbit/s)
H.262		= MPEG-2 Teil 2
H.263	1995	Verbesserung der Kompression gegenüber MPEG-2 / H.262 um Faktor 2
H.263+	1998	Neu: Deblocking-Filter
H.263++	2000	Neu: Langzeit-Prädiktion
MPEG-3	(nie erschienen)	Sollte der Standard für HDTV werden. Es genügte aber eine Erweiterung von MPEG-2. MPEG-3 kam daher nie zustande.
MPEG-4 Teil 2 (Visual)	2001	Objektbasierte Codierung, Integration von 3D-Objekten, Kombination von synthetischen und natürlichen Bildinhalten, nicht-rechteckige Video-Objekte, verbesserte Kompression.
H.264	2003	= MPEG-4 Teil 10 bzw. ISO/IEC 14496-10 AVC. Weitere Bezeichnungen: H.26L, JVT Codec, Advanced Video Coding (AVC), JM2.x, JM3.x und JM4.x
H.264 FRExt	2004	Erweiterung um zusätzliche Profile für professionelle Anwendungen
(H.264 SVC)	(2007)	(Scalable Video Coding – noch in Arbeit)

Tab. 1: Übersicht Videokompressions-Standards

4 H.264

H.264 ist der jüngste internationale Standard in der Videokompression, dessen erste Version im Mai 2003 gemeinsam von zwei Standardisierungsgremien verabschiedet wurde: Von der ITU-T unter dem Namen H.264; von der ISO/IEC unter dem Namen MPEG-4 Teil 10 oder auch MPEG-4 AVC (Advanced Video Coding) bzw. ISO/IEC 14496-10.

Aufbauend auf seine Vorgänger aus der H.26x-Serie, schloss die Video Coding Experts Group (VCEG) der ITU-T im August 1999 die Entwicklung ihres H.263-Standards ab. Parallel dazu wurde aber bereits seit 1998 am Entwurf für einen neuen Standard, ursprünglich unter dem Namen H.26L gearbeitet, mit dem Ziel, die Kompressionsrate gegenüber allen bisherigen Standards mindestens zu verdoppeln (L = long-term prediction – zeitlich weiter zurückliegende Referenzbilder).

In den anfänglichen Entwürfen unterschied sich H.26L bezüglich der Kompressionsleistung kaum von seinen Vorgängern H.263 und MPEG-4 Teil 2 bzw. war ihnen in dieser Hinsicht sogar unterlegen. Von Anfang an zeigte der neue Standard aber Vorzüge in der objektiven Bildwahrnehmung, und es war klar, dass man durch weitere Entwicklung auch die Effizienz steigern würde.

Als 2001 bereits eine erhebliche Verbesserung in der Kompressionsleistung gegenüber den Vorgängern erreicht worden war, schloss sich die Moving Pictures Expert Group (MPEG), eine Arbeitsgruppe der ISO/IEC, der Entwicklung des neuen Standards an, und es wurde von da an gemeinsam im „Joint Video Team“ (JVT) weiter gearbeitet.

In den folgenden Jahren wurden schrittweise (über insgesamt 15 vorläufige Versionen) die einzelnen Komponenten des Standards so optimiert, dass schließlich 2003 das vorgegebene Ziel, d.h. die Halbierung der Datenrate bei gleicher Bildqualität gegenüber allen anderen Standards, erreicht werden konnte.

Zu den Entwicklungszielen abgesehen von der effizienteren Kompression zählten eine verbesserte Netzwerktauglichkeit – einerseits hinsichtlich der Möglichkeit zur Einbindung in bestehende Übertragungsprotokolle, andererseits betreffend die Robustheit gegenüber Übertragungsfehlern bei mobilen Netzwerken und im Internet. Der neue Standard sollte außerdem möglichst simpel gehalten sein und eine zu große Vielfalt von verschiedenen Optionen und Profilen vermeiden.

Es war von Anfang an klar, dass H.264 wegen seiner komplexeren Codierungselemente die Anforderungen aller vorherigen Standards betreffend Rechenkapazität um ein mehrfaches überschreiten würde – verglichen mit MPEG-2 etwa zwei- bis dreimal soviel auf der Encoderseite, und drei- bis viermal so viel beim Decodieren. Dieser Nachteil wurde zukunftsweisend bewusst in Kauf genommen, weil sich Prozessorleistungen und Arbeitsspeicherkapazitäten ohnehin ständig weiter entwickeln.

Für Kommunikationsanwendungen, bei denen geringe Verzögerungszeiten wichtig sind und nicht genügend Rechenleistung zur Verfügung steht, sollte der Standard aber auch Lösungen mit geringem Berechnungsaufwand bieten.

Anders als bei den bisherigen MPEG-Standards war bei H.264 keine Rückwärts-Kompatibilität zu den Vorgängern vorgesehen.

4.1 Technische Beschreibung

H.264 ist kein „Wundercodec“ in dem Sinn, dass ein neuer Algorithmus nie zuvor da gewesene Ergebnisse erzielt. Die beträchtliche Effizienzsteigerung gegenüber anderen Standards kommt größtenteils aus der Verbesserung und Verfeinerung bereits zuvor verwendeter Elemente zustande.

Folgende Merkmale hat H.264 mit MPEG-2 bzw. seinen anderen Vorgängern gemeinsam:

- blockbasierte Verarbeitung
- Einsatz einer Transformation
- Quantisierung und Zick-Zack-Abtastung der Transformationskoeffizienten
- Zeitliche Prädiktion von Blöcken mit Hilfe von Bewegungsvektoren
- I-, P- und B-Bilder
- Entropiecodierung mit VLC
- Unterteilung der Bilder in Slices und Makroblocks
- Makroblocks umfassen 16x16 Bildpunkte in der Luminanz
- 4:2:0 Farbunterabtastung, Makroblocks umfassen 8x8 Bildpunkte in den Chrominanz (vgl. MPEG1)
- Spezielle Behandlung von Halbbildern (vgl. MPEG2)

Hier zunächst ein kurzer Überblick über die Verbesserungen und zusätzlichen neuen Elemente, die anschließend im Detail beschrieben werden:

Verbesserte Prädiktion durch:

- Örtliche Prädiktion von Blöcken / Intra-Prädiktion (ganz neu)
- Flexible Blockgröße bei der Bewegungskompensation, bis hinunter zu 4x4
- Bewegungsvektoren mit Genauigkeit von $\frac{1}{4}$ -Bildpunkt (war schon Option bei MPEG-4 Teil 2)
- Bewegungsvektoren über Bildgrenzen hinaus (gab es schon bei H.263)
- Multiple Referenzbilder
- Loslösen der Referenzierung von der Reihenfolge der Bilder
- Auch B-Bilder können als Referenzbilder verwendet werden
- Gewichtete Mittelung von Referenzblöcken bei bidirektionaler Prädiktion
- Umschalten zwischen Vollbild- und Halbbild-Modus auch auf Makroblock-Ebene möglich
- Deblocking-Filter in der zeitlichen Prädiktionsschleife

Effizienzsteigerung in der Codierung durch:

- Transformation basiert auf 4x4 Pixel-Blöcken (statt 8x8)
- Bei gleichmäßigen Bildbereichen wird ein 16x16-Modus mit hierarchischer Transformation eingesetzt
- Integer-Transformation mit 16-Bit-Arithmetik (statt 32-Bit)
- Exakte inverse Transformation
- Logarithmische Abstufung der globalen Quantisierung
- Deutlich verbesserte Entropiecodierung (CAVCL und CABAC)

Verbesserte Robustheit gegenüber Übertragungsfehlern und Flexibilität in der Anpassung an verschiedene Netzwerkkumgebungen durch:

- Syntax-Struktur mit NAL- (network abstraction layer) Units
- Flexible Slice-Größe (Slice = Gruppe von Makroblocks)
- Flexible Anordnung der Makroblocks (FMO, flexible macroblock ordering)
- Beliebige Anordnung der Slices (ASO, arbitrary slice order)
- Redundante Bilder

4.1.1 Ganz neu: Örtliche Prädiktion (Intra-Prädiktion) von Makroblocks

Ein Prinzip, das in H.264 ganz neu eingeführt wird, ist die Intra-Prädiktion von Makroblocks basierend auf den Werten bereits codierter unmittelbar angrenzender Blocks. (Eine Art von Intra-Prädiktion wurde bereits in H.263+ und MPEG-4 Teil 2 verwendet, dort allerdings bezogen auf die Transformationskoeffizienten).

Jeder Makroblock, dessen Werte nicht zeitlich vorhergesagt werden, kann in einem von drei verschiedenen Intra-Prädiktions-Modi codiert werden: Der Intra-4x4-Modus wird auf Blocks mit großen Details angewandt, der Intra-16x16-Modus für gleichförmige Bereiche. Im I_PCM-Modus werden die Werte direkt codiert, er ist somit die einzige Möglichkeit, die aktuellen Werte präzise wiederzugeben. Das ist einerseits notwendig, um untypische Bildinhalte mit möglichst geringem Datenaufwand zu codieren, und setzt andererseits eine Obergrenze für die Menge von Bits, die der Decoder pro Makroblock verarbeiten können muss.

Bei der Intra-4x4-Prädiktion gibt es neun verschiedene Richtungs-Modi, bei denen die Werte des Blocks aus den unmittelbar oben und/oder links angrenzenden Pixelwerten vorhergesagt werden.

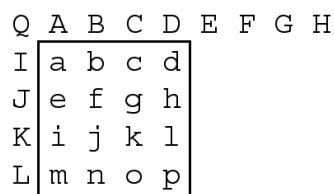


Abb. 5: Intra-4x4-Prädiktion: Werte der Pixel a-p werden aus bereits codierten Werten A-L vorhergesagt.

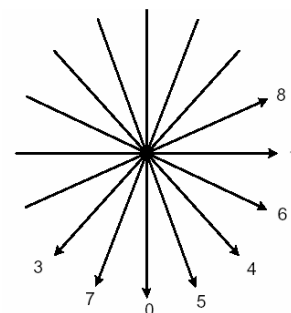


Abb. 6: Die acht möglichen Richtungen der Vorhersage. Modus 2 ist die DC-Prädiktion.

Modus 0 ist beispielsweise die rein vertikale Prädiktion, bei der die Werte der 4 oben angrenzenden Pixel vertikal nach unten kopiert werden, Modus 1 rein horizontal. In Modus 2, DC-Prädiktion, wird ein gemittelter Wert der oben und links angrenzenden Pixel auf den gesamten Block kopiert. Weitere Möglichkeiten sind: diagonal nach unten / links bzw. rechts, vertikal-rechts, horizontal-nach unten, vertikal-links, und horizontal-nach oben.

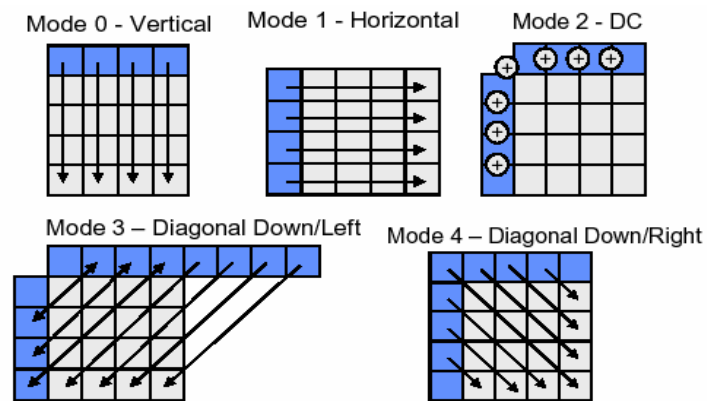


Abb. 7: Fünf der neun Intra-4x4-Prädiktions-Modi

Wenn die Werte für Pixel E-H nicht vorhanden sind, weil sie noch nicht codiert wurden, oder zu einem anderen Slice gehören, werden sie durch den Wert von D ersetzt.

In früheren Entwürfen des Standards wurden auch die vier Werte vertikal unter Pixel L für einige Intra-Prädiktionsmodi herangezogen, es zeigte sich jedoch, dass diese Möglichkeit nicht genug zur Effizienz beitragen konnte, um den zusätzlichen Rechenaufwand zu rechtfertigen.

In der Intra-16x16-Prädiktion, die bei gleichmäßigen Bildbereichen eingesetzt wird, wird der gesamte Luminanzanteil eines Makroblocks vorhergesagt. Hier gibt es vier verschiedene Modi – 0, 1 und 2 stehen wie bei der Intra-4x4-Prädiktion für vertikale, horizontale und DC-Prädiktion, es werden dementsprechend jeweils die 16 Werte der oben bzw. links angrenzenden Pixel verwendet.

Modus 3 ist eine Ebenen-Prädiktion, in der die vorhergesagten Werte durch eine lineare Funktion zwischen den oben und links benachbarten Pixeln errechnet werden, und eignet sich besonders für Bereiche, in denen die Werte sich nur schwach ändern.

Die Chrominanzwerte werden ausschließlich makroblockweise, d.h. in 8x8-Blocks, mit 4 verschiedenen Modi analog zur 16x16-Intra-Prädiktion behandelt.

4.1.2 Kleinere Sub-Blocks für bessere Bewegungskompensation

Eine deutliche Verbesserung der Bewegungskompensation wird bei H.264 durch eine je nach Bildinhalt adaptive Unterteilung der Makroblocks in kleinere Einheiten bis 4x4-Blocks erreicht. Bei MPEG-4 Teil 2 und H.263 ist bereits eine flexible Blockgröße in der Bewegungskompensation möglich, allerdings nur 16x16- und 8x8-Blocks.

Die Unterteilung erfolgt in zwei Schritten: Zuerst wird der Makroblock bei Bedarf in kleinere Teile zerlegt (16x8, 8x16 oder 8x8 Bildpunkte); In einem weiteren Schritt wird bei Bedarf jeder entstandene 8x8-Block weiter unterteilt (in 8x4, 4x8 oder 4x4 Bildpunkte).

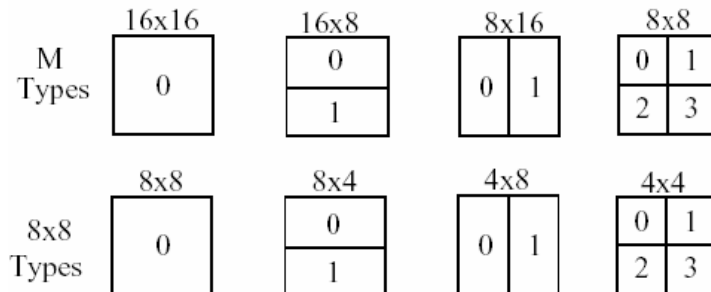


Abb. 8: Oben: Unterteilung eines Makroblocks für die Bewegungskompensation
Unten: weitere Unterteilung eines 8x8-Blocks

Pro Unterteilung wird ein Bewegungsvektor zusammen mit einem Bildreferenz-Index übermittelt. Insgesamt können so pro P-Makroblock bis zu 16 Bewegungsvektoren übertragen werden.

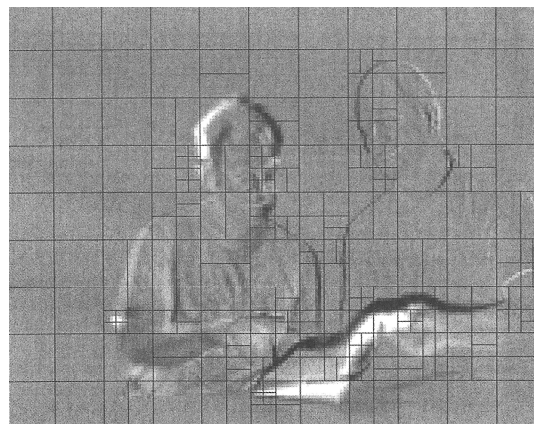


Abb. 9: Differenzbild (ohne Bewegungskompensation)
mit Unterteilung der Blockgrößen

4.1.3 Bewegungskompensation mit Genauigkeit von 1/4-Bildpunkt

Ein weiteres Element, das die Bewegungskompensation deutlich gegenüber anderen Codecs verbessert und damit zur Effizienzsteigerung beiträgt, ist eine Vorhersage-Genauigkeit von einem Viertel-Bildpunkt. (Seit MPEG-1 ist die Genauigkeit bei 1/2-Pixel-Position; Bei MPEG-4 ist 1/4-Pixel-Genauigkeit im Advanced Simple Profile vorgesehen.)

Wenn der Bewegungsvektor nicht auf eine ganzzahlige Bildpunkt-Position zeigt, werden die Prädiktionswerte interpoliert. Für die 1/2-Pixel-Positionen wird dabei ein Interpolationsfilter verwendet.

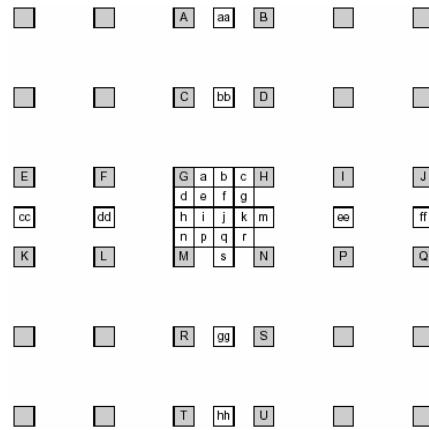


Abb. 10: Interpolation bei der Bewegungsvorhersage

Eine rein horizontale oder vertikale Interpolation wird folgendermaßen berechnet:

$$b_1 = (E - 5 F + 20 G + 20 H - 5 I + J)$$

$$h_1 = (A - 5 C + 20 G + 20 M - 5 R + T)$$

$$b = (b_1 + 16) \gg 5$$

$\gg \dots$ Bitshift – Operation nach rechts

$$h = (h_1 + 16) \gg 5$$

(entspricht Division durch 2 mit Abrunden)

Werte, bei denen horizontal und vertikal interpoliert wird berechnen sich wie folgt:

$$j_1 = cc - 5 dd + 20 h_1 + 20 m_1 - 5 ee + ff$$

$$j = (j_1 + 512) \gg 10$$

Für die Werte auf 1/4-Pixel-Positionen wird entweder zwischen den beiden angrenzenden Ganz- und 1/2-Pixel-Positionen (für Positionen a, c, d, n, f, i, k, q laut Skizze) oder den beiden diagonal angrenzenden 1/2-Pixel-Positionen (Position e, g, p und r laut Skizze) gemittelt.

$$a = (G + b + 1) \gg 1$$

$$e = (b + h + 1) \gg 1$$

Alle errechneten Werte werden in einem letzten Schritt auf den Bereich zwischen 0 und 255 eingeschränkt.

Die Prädiktionswerte für die Chrominanz werden immer mittels bilinearer Interpolation berechnet. Die Genauigkeit liegt entsprechend der 4:2:0-Unterabtastung bei $\frac{1}{8}$ -Pixel.

4.1.4 Bewegungsvektoren über die Bildgrenze hinaus

Zur Optimierung der Bewegungsvorhersage ist es möglich, dass Bewegungsvektoren auf Bereiche außerhalb des aktuellen Bildes zeigen. Dieses Prinzip war schon Teil von H.263 und MPEG-4 Teil 2. Dafür wird das Referenzbild über die Bildgrenze hinaus extrapoliert, indem die Randwerte wiederholt werden.

4.1.5 Prädiktion der Bewegungsvektoren

Um die Codierung der Bewegungsvektoren möglichst effizient zu halten, wird auch bei ihnen eine Prädiktionmethode angewandt. Die Bewegungsvektoren werden mit Hilfe von drei Nachbarblocks vorausgesagt und anschließend nur mehr der Vorhersagefehler codiert.

Alle Formen der Prädiktion, sei es bei den Bewegungsvektoren oder dem Bildinhalt selbst, werden jeweils nur innerhalb von Slice-Grenzen angewandt, sodass ein Slice immer unabhängig von anderen Bildteilen verarbeitet werden kann.

4.1.6 Mehrere Referenzbilder für die Bewegungsvorhersage

Eine neue Möglichkeit, die es vor H.264 noch nicht gegeben hat, ist die Verwendung von mehr als einem bereits codierten/decodierten Bild als Referenz für die Bewegungskompensation. Die Anzahl der Referenzbilder ist hier nur durch den verfügbaren Speicherplatz für den Referenzbild-Puffer des Encoders begrenzt.

4.1.7 P_Skip-Blocks

Zusätzlich zu den beschriebenen Codierungsmethoden für P-Makroblocks gibt es die Möglichkeit der so genannten P_Skip-Codierung. Für einen solchen Block wird weder

der Prädiktionsfehler noch ein Bewegungsvektor oder Referenzparameter übertragen. Rekonstruiert wird der Block aus dem Referenzblock mit Index 0 (unmittelbar vorhergehendes Bild) und dem vorhergesagten Bewegungsvektor. Diese Art der Codierung ist sehr effektiv bei großen Flächen ohne Veränderung oder mit konstanter Bewegung (z.B. langsamer Schwenk).

4.1.8 B-Prädiktion

Das Konzept der bidirektionalen Bewegungsprädiktion, das es seit MPEG-1 gibt, wird in H.264 erweitert. Bisher war es so, dass die Bewegungsvorhersage von Makroblocks innerhalb von B-Bildern aufgrund der Werte eines vergangenen und eines zukünftigen Bildes erfolgen konnte. Diese Referenzbilder konnten entweder I- oder P-Bilder sein.

Bei H.264 kann ein Makroblock innerhalb eines B-Slices auf eine von vier verschiedenen Arten codiert werden. Es werden zwei verschiedene Listen von Referenzbildern verwendet, Liste 0 und Liste 1. Welche Bilder jede Liste genau enthält, wird innerhalb des Referenzbild-Puffers kontrolliert. Für jeden Block erfolgt die Bewegungsvorhersage entweder aufgrund eines Bildes aus der Liste 0, oder eines aus der Liste 1, oder eines gewichteten (je nach zeitlichem Abstand zum aktuellen Bild) Mittelwertes aus einem Liste-0- und einem Liste-1-Referenzbild.

Es ist dabei nicht relevant, ob die Referenzbilder zeitlich vor oder nach dem aktuellen Bild liegen. Die Bedeutung des „B“ ändert sich also sozusagen von „bidirektional“ auf „biprädiktiv“.

Die vierte Möglichkeit der Codierung innerhalb von B-Slices ist der sog. B_Skip-Modus analog zum P_Skip-Modus. Die Entscheidung über die Art der Codierung kann für jeden Makroblock bzw. jeden kleineren Unterteilungsblock einzeln erfolgen.

Im Gegensatz zu früheren Standards ermöglicht H.264 auch die Verwendung von Bildern mit B-Slices als Referenzbilder.

4.1.9 Behandlung von Halbbildern

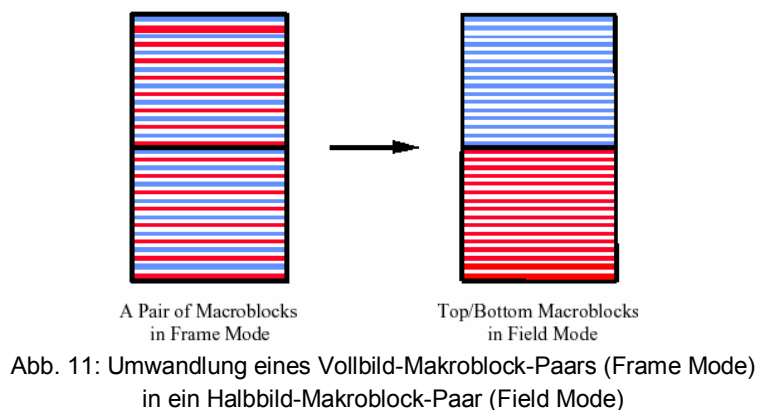
Wenn ein Videosignal im Halbbildmodus vorliegt („interlaced“), kann es bei bewegten Bildinhalten effizienter sein, jedes Halbbild separat zu komprimieren. H.264 sieht vor,

dass der Encoder die effizienteste von drei verschiedenen Varianten auswählt, mit Halbbildern umzugehen.

Im PAFF-Modus (picture adaptive frame/field coding) wird für jedes Bild zwischen Vollbild- oder Halbbildmodus entschieden und dementsprechend entweder zwei Halbbilder zusammengefasst als ganzes Bild codiert oder jedes Halbbild für sich codiert.

Der Unterschied bei der Codierung im Halbbildmodus liegt lediglich darin, dass statt Referenzbildern Referenz-Halbbilder verwendet werden, und dass die Transformationskoeffizienten in einem anderen Muster abgetastet werden.

Eine weitere Möglichkeit bietet der MBAFF-Modus (macroblock-adaptive frame/field coding), bei dem die Entscheidung über Vollbild- oder Halbbild-Codierung einzeln für jedes vertikal aneinander grenzende Makroblock-Paar getroffen wird. Diese Art der Codierung ist dann sinnvoll, wenn das Bild sowohl statische als auch Regionen mit Bewegung enthält.



Die Entscheidung darüber, welche der drei Möglichkeiten zur Anwendung kommt, kann für jedes Bild einzeln getroffen werden.

Der PAFF-Modus zeigte in Versuchen eine Effizienzsteigerung zwischen 16 und 20% gegenüber reinem Vollbildmodus. MBAFF wiederum erreichte bei bestimmten Bildinhalten Steigerungen zwischen 14 und 16% gegenüber PAFF (vgl. [18], S. 7/8). Die Bestimmung von benachbarten Makroblocks ist im MBAFF-Modus vergleichsweise kompliziert, und auch die Bewegungsvorhersage ist wegen der verschiedenen Arten der Codierung innerhalb eines Bildes nicht immer so effizient wie bei PAFF. Bei sehr schneller globaler Bewegung oder Szenenwechsel kann daher der PAFF-Modus der günstigere sein.

4.1.10 Transformation

Ganz neu bei H.264 ist, dass die Transformation der Prädiktionsfehlerwerte nicht auf 8x8- sondern auf 4x4-Blocks angewandt wird. Keine Diskrete Kosinustransformation, sondern eine (stark daran angelehnte) Integer-Transformation wird eingesetzt.

Diese Transformation hat zwar mehr oder weniger dieselben Eigenschaften wie die DCT, kann aber im Gegensatz dazu exakt invertiert werden und somit Rundungsfehler vermeiden. Sie kann durch reine Additions- und Shift-Operationen bei einer 16-Bit-Arithmetik (bei 8-Bit-Tiefe der Werte) implementiert werden. Das bedeutet einen wesentlich geringeren Berechnungsaufwand bei der Transformation im Vergleich zu früheren Standards, die für die DCT 32 Bits brauchten.

Die 4x4-Blockgröße kommt der möglichen feineren Unterteilung der Makroblocks bei der Bewegungskompensation entgegen. Außerdem vermindern sich dadurch Artefakte wie das Rauschen an Kanten („ringing“) und Blockstrukturen bei stärkerer Quantisierung.

Durch die 4x4-Transformation wird zwar gegenüber der 8x8-DCT keine bessere Kompressionsleistung erzielt, es wird dafür aber an Rechenleistung gespart, die bei anderen Elementen von H.264 stark in Anspruch genommen wird.

Die Transformationsmatrix für 4x4-Luminanz- und Chrominanzblocks sieht folgendermaßen aus:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Bei Makroblocks, die im Intra-16x16-Modus codiert werden (gleichmäßige Bildbereiche), wird in einem zweiten Schritt eine weitere Transformation (Walsh-Hadamard-Transformation) auf die 16 resultierenden DC-Koeffizienten der Luminanzkomponente angewandt. (Der DC-Koeffizient ist der erste (links oben) und wichtigste Transformationskoeffizient. Er repräsentiert den Gleichanteil jedes transformierten Blocks.) Durch diesen Schritt ergeben sich 4x4 Transformationskoeffizienten, die den gesamten Makroblock repräsentieren.

Transformationsmatrix für die 4x4 DC-Koeffizienten eines im Intra-16x16-Modus codierten Makroblocks:

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Für die Chrominanzkomponenten eines 16x16-intra-codierten Makroblocks ergeben sich nach der ersten Transformation 4 DC-Koeffizienten (wegen der Farb-Unterabtastung), die mit der folgenden 2x2-Matrix noch einmal transformiert werden:

$$H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

4.1.11 Quantisierung

Die Stärke der Quantisierung wird bei H.264 von einem Quantisierungsparameter bestimmt, der 52 verschiedene Werte annehmen kann. Diese Werte sind so gewählt, dass bei Erhöhung des Parameters um 1 die Quantisierungsschritte um 12% vergrößert werden. Wird der Quantisierungsparameter um 6 erhöht, entspricht das einer Vergrößerung der Quantisierungsschritte um den Faktor 2. Eine Vergrößerung der Quantisierungsschritte um einen gewissen Prozentsatz erwirkt gleichzeitig auch die Reduktion der Bitrate etwa im selben Ausmaß.

Im Vergleich zu früheren Standards können die Quantisierungsschritte viel kleiner gewählt werden, wodurch eine Kompression auch bei sehr hoher Bildqualität ermöglicht wird.

Die quantisierten Transformationskoeffizienten werden im Zick-Zack-Muster abgetastet, ausgenommen die 2x2 DC-Koeffizienten der Chrominanzkomponente; Hier erfolgt die Abtastung zeilenweise.

4.1.12 Entropiecodierung

H.264 bietet zwei verschiedene Arten der Entropiecodierung, die beide eine laufende Anpassung an die aktuelle Signalstatistik ermöglichen, und deshalb effizienter sind als früher verwendete Arten: CAVLC (context adaptive variable length coding) und CABAC (context adaptive binary arithmetic coding).

CAVLC wird nur auf die quantisierten Transformationskoeffizienten angewandt. Alle restlichen Syntaxelemente werden in dieser Variante der Entropiecodierung auf Basis einer einzigen VLC-Tabelle codiert, wobei die Zuordnung zu dieser Tabelle an die Datenstatistik angepasst wird.

Bei der Codierung der Transformationskoeffizienten wird die effizientere CAVLC-Methode eingesetzt, bei der in Abhängigkeit von den bereits übertragenen Daten zwischen verschiedenen VLC-Tabellen gewechselt wird.

Die Anzahl der Koeffizienten ungleich Null und derer mit dem Wert 1 werden zusammengefasst und nach einer von vier verschiedenen VLC-Tabellen codiert, die aufgrund der Werte der benachbarten Blocks ausgewählt wird. Die Werte dieser Koeffizienten werden dann separat (in umgekehrter Scan-Reihenfolge) nach einer von sechs verschiedenen Tabellen codiert, wobei die Tabelle je nach Wert des vorangegangenen Koeffizienten gewechselt werden kann. Auch für die Codierung der Anzahl und Position der Nullwerte vor dem letzten Wert ungleich Null wird zwischen verschiedenen Tabellen gewählt.

Mit CABAC ermöglicht H.264 eine weitere Methode der Entropiecodierung, die CAVLC um 5-15% überbieten kann. Besonders effizient ist CABAC bei der Kompression von Videosignalen im Halbbildformat. Die arithmetische Codierung bietet den Vorteil, dass einem Symbol nicht nur ganze sondern auch Bruchteile von Bits zugeordnet werden können und damit den Wahrscheinlichkeiten besser entsprochen werden kann. Außerdem werden die bereits codierten Elemente verwendet, um Wahrscheinlichkeiten zu bestimmen und zwischen verschiedenen Wahrscheinlichkeitsmodellen zu wählen.

In der H.264-Spezifikation wird die CABAC-Methode durch einfache Shift-Operationen und Tabellensuchen festgelegt. Trotzdem ist dafür ein bedeutender Mehraufwand an Rechenleistung gegenüber CAVCL notwendig.

4.1.13 Deblocking-Filter in der Prädiktionsschleife

Neben der effizienten Kompression hat H.264 eine zweite große Stärke, die jedem Betrachter sozusagen sofort ins Auge springt, nämlich die eindeutig bessere subjektive Bildqualität im Vergleich zu anderen Standards. Sie wird durch einen Filter erreicht, der innerhalb der Prädiktionsschleife liegt, und dadurch gleichzeitig die Präzision der Bewegungsvorhersage deutlich steigert.

Der Filter verringert die so genannten Block-Artefakte, die die subjektive Betrachtungsqualität mehr als alle anderen Arten der Bildverzerrung beeinträchtigen können. Blockstrukturen im decodierten Bild entstehen einerseits bei starker Quantisierung der Transformationskoeffizienten und andererseits durch die blockbasierte Bewegungskompensation. Wird ein Bild mit solchen Artefakten als Referenzbild verwendet, setzt sich die Blockstruktur im nächsten Bild fort. Dabei wird auch die Bewegungsvorhersage ungenauer, damit der Prädiktionsfehler größer und die Effizienz geringer.

Die kleinere Blockgröße bei der Transformation und Bewegungskompensation bei H.264 kann solche Probleme bereits deutlich vermindern. Durch den Deblocking-Filter wird aber die Kompressionsrate immerhin noch um 5-10% gegenüber dem ungefilterten Video (bei gleicher objektiver Qualität) gesteigert (vgl. [18], S. 14).

Die Stärke des Deblocking-Filters wird adaptiv je nach Quantisierungsstärke mit Hilfe mehrerer Parameter angepasst.

Die Grundidee dahinter ist folgende: Wenn Werte von Bildpunkten an aneinander grenzenden Blockkanten eine relativ große Differenz aufweisen, ist es ziemlich wahrscheinlich, dass hier ein Blockartefakt vorliegt. In diesem Fall wird der Filter eingesetzt und die Blockkanten geglättet. Wenn aber die Differenz der Werte an den Blockrändern so groß ist, dass sie nicht durch die Quantisierung verursacht sein kann, ist es wahrscheinlicher, dass es sich um die korrekte Repräsentation der aktuellen Bildwerte (also eine scharfe vertikale oder horizontale Kante) handelt. Die Werte werden daher nicht verändert.

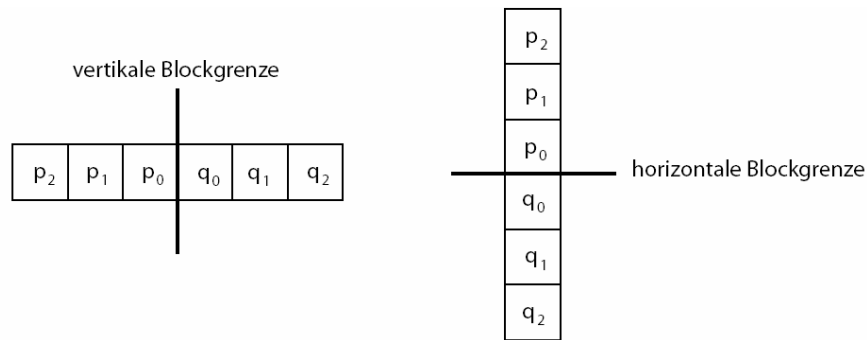


Abb. 12: Pixelwerte an den Blockgrenzen

Die Entscheidung darüber, ob p_0 , p_1 , q_0 oder q_1 gefiltert werden, wird mit Hilfe von zwei vom Quantisierungsparameter QP abhängigen Grenzwerten $\alpha(QP)$ und $\beta(QP)$ geregelt, wobei $\beta(QP)$ wesentlich kleiner ist als $\alpha(QP)$. Eine Filterung von p_0 und q_0 erfolgt dann, wenn alle der drei folgenden Bedingungen erfüllt sind:

$$|p_0 - q_0| < \alpha(QP)$$

$$|p_1 - p_0| < \beta(QP)$$

$$|q_1 - q_0| < \beta(QP)$$

Eine Filterung von p_1 oder q_1 erfolgt bei Zutreffen der Bedingungen

$$|p_2 - p_0| < \beta(QP) \text{ oder } |q_2 - q_0| < \beta(QP).$$



Abb. 13: Leistung des Deblocking-Filters: linkes Bild ohne Filter, rechtes mit Filter

4.1.14 FMO – Flexible Anordnung von Makroblocks (flexible macroblock ordering)

Die im MPEG-1-Standard eingeführte Unterteilung des Bildes in ein oder mehrere Slices wird in H.264 um zusätzliche Möglichkeiten erweitert. Wie bereits erwähnt, besteht ein Slice aus einer Anzahl von aufeinander folgenden Makroblocks, und wird unabhängig von anderen Bildteilen codiert. Im herkömmlichen Slice-Muster (s. Abb. 14) werden die Makroblocks zeilenweise verarbeitet.

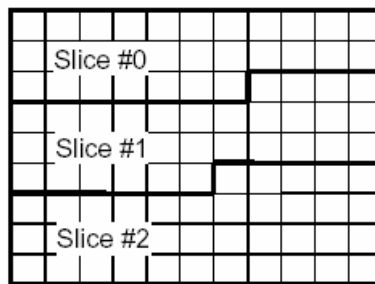


Abb. 14: Unterteilung eines Bildes in Slices (ohne FMO)

FMO (flexible macroblock ordering) sieht eine Unterteilung des Bildes in Slice-Gruppen (bestehend aus einem oder mehreren Slices) vor, die in verschiedenen Anordnungen erfolgen kann. Es ist zum Beispiel möglich, Vordergrund und Hintergrund verschiedenen Slice-Gruppen zuzuordnen, oder die Slice-Gruppen in einer Art Schachbrettmuster anzuordnen.

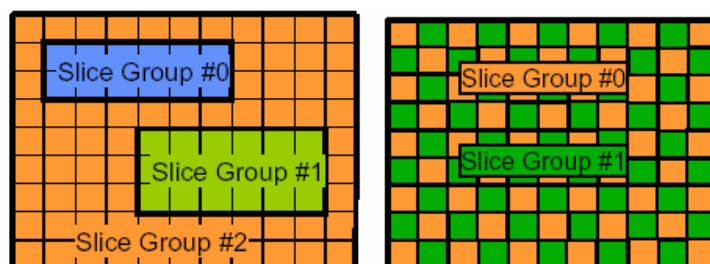


Abb. 15: Unterteilung eines Bildes in Slices mit FMO

Innerhalb jedes Slices einer Slice-Gruppe erfolgt die Verarbeitung der Makroblocks wiederum zeilenweise.

FMO kann z.B. so eingesetzt werden, dass bei einem Verlust von einem oder mehreren Datenpaketen die resultierenden Fehler gleichmäßig über das Bild verstreut sind, statt in

einzelnen Bildregionen aufzutreten. Durch die vorhandenen angrenzenden Bildinhalte können die fehlenden Daten so leichter kaschiert werden.

4.1.15 ASO – Beliebige Anordnung der Slices (arbitrary slice order)

H.264 ermöglicht die Decodierung der Slices eines Bildes in jeder beliebigen Reihenfolge. Das ist speziell bei paketerorientierten Systemen notwendig, bei denen die Pakete nicht notwendigerweise in der ursprünglichen Reihenfolge übertragen werden.

4.1.16 SP- und SI-Slices

SP- und SI-Slices (switching P/I slices) sind speziell codierte Slices, die u.a. ein effizientes Wechseln zwischen Video-Streams mit unterschiedlichen Bitraten und einen schnellen Zugriff des Decoders auf einzelne Bilder ermöglichen.

SP-Slices unterstützen ein Wechseln zwischen verschiedenen codierten Sequenzen mit dem gleichen Inhalt. Bei einem Video, das mit unterschiedlichen Bitraten codiert wurde, ist dadurch ein Umschalten zwischen niedriger und höherer Bitrate je nach momentanem Datendurchsatz in einem Kanal möglich, ohne das Datenaufkommen durch die Verwendung von I-Slices unverhältnismäßig zu erhöhen.

SI-Slices bieten u.a. die Möglichkeit, zwischen verschiedenen Sequenzen mit unterschiedlichen Inhalten zu wechseln, ohne dass sich dabei Probleme wegen der fehlenden Referenzbilder ergeben.

4.1.17 Data Partitioned Slices (DP) und Redundant Slices (RS)

Zwei weitere Maßnahmen, um die Fehlerrobustheit der Daten auf unzuverlässigen Übertragungswegen zu erhöhen, sind die Möglichkeit, die Daten eines Slices je nach Priorität in mehrere unabhängig decodierbare Teile zu gliedern und in unterschiedliche Datenpakete zu packen (data partitioned slices, DP), oder ganze Slices mehrfach zu übertragen (Redundant Slices).

4.2 Network Abstraction Layer (NAL)

H.264 unterscheidet zwischen den komprimierten Videodaten (VCL – video coding layer) und einer übergeordneten Schicht, die diese Daten optimal für jede erwünschte Verwendung in Einheiten „verpackt“ – der sog. „network abstraction layer“ (NAL).

Die NAL ist so konzipiert, dass sie sowohl in paketerorientierte als auch in Transportsysteme mit kontinuierlichem Datenstrom problemlos eingebunden werden kann.

4.2.1 NAL-Units

Jede NAL-Einheit („unit“) besteht aus einer ganzzahligen Anzahl von Bytes. Das erste davon ist ein Header-Byte, das anzeigt, welche Datenart diese NAL-Unit enthält. Außer den Videodaten können auch Bytes eingefügt sein, die verhindern, dass ein bestimmtes Datenmuster, der „start code prefix“ zufällig auftritt. Dieses aus drei Bytes bestehende Zeichen zeigt bei der Einbindung in einen Bytestrom jeweils den Beginn einer neuen NAL-Unit an.

Neben den NAL-Units mit Videodaten (VCL-NAL-Units) gibt es auch welche, die zusätzliche Informationen enthalten, wie etwa Parameter-Sets, die sich auf eine große Anzahl von NAL-Units beziehen, oder andere Zusatzdaten, die das decodierte Videosignal verbessern können, aber nicht selbst für die Decodierung notwendig sind.

4.2.2 NAL-Units im Bytestrom-Format

Für die Einbindung von H.264 in Transportströme wie z.B. MPEG-2 Systems (ITU-T H.222.0) oder H.320 (ITU-T-Standard zum Transport von Multimediatdaten über ISDN-Netzwerke) wurde ein Bytestrom-Format festgelegt. Um den Anfang und das Ende jeder NAL-Unit eindeutig zu identifizieren, wird jeder Einheit ein drei Byte langes „start code prefix“ vorangestellt. Zusätzlich wird ein Byte pro Videobild eingefügt, um Systemen, die nicht von sich aus auf die Bytegrenzen eingerichtet sind, den Abgleich zu erleichtern.

4.2.3 NAL-Units in paketorientierten Transportsystemen

In paketorientierten Systemen wie IP/RTP (internet protocol / real-time transport protocol) werden die NAL-Einheiten ohne weitere Zusatzinformation direkt in die protokolleigenen Pakete gepackt.

4.2.4 Parameter-Sets

H.264 definiert zwei verschiedene Arten von Parameter-Sets, die in NAL-Units vorkommen können:

- Sequenz-Parameter-Sets beziehen sich auf eine Serie aufeinander folgender codierter Videobilder, eine so genannte „codierte Videosequenz“ („coded video sequence“).
- Bild-Parameter-Sets beziehen sich auf ein oder mehrere einzelne Bilder innerhalb einer codierten Videosequenz.

Die Parameter-Sets erlauben eine Übertragung von Informationen, die über längere Zeit gleich bleiben, getrennt von den Daten, die die codierten Videobilder repräsentieren, und reduzieren so die zu übertragende Datenmenge.

Parameter-Sets können lange vor den VCL-NAL-Units übertragen werden, auf die sie sich beziehen, und können auch wiederholt werden um die Sicherheit vor Datenverlust zu erhöhen.

4.2.5 Access-Units und redundant codierte Bilder

Die Menge von NAL-Units, die ein ganzes decodiertes Bild ergibt, wird als Access-Unit (access = Zugriff) bezeichnet. Zusätzlich zu den Videodaten können auch redundante VCL-NAL-Units aus dem selben Bild in einer Access-Unit vorhanden sein, die vom Decoder bei Datenverlusten oder Übertragungsfehlern verwendet werden kann, um das Bild zu rekonstruieren. Solche Bilder werden als redundant codiert bezeichnet (redundant coded pictures).

4.2.6 Codierte Videosequenzen

Eine codierte Videosequenz besteht aus einer Abfolge von NAL-Units, die ein gemeinsames Sequenz-Parameter-Set benutzen, und kann unabhängig von jeder anderen codierten Videosequenz decodiert werden. Am Beginn jeder codierten Videosequenz steht eine IDR-Access-Unit (instantaneous decoding refresh), die ein intra-codiertes Bild enthält, welches unabhängig von allen vorhergehenden Bildern im Datenstrom decodiert werden kann.

4.3 Profile und Levels

Weil es für unterschiedliche Anwendungsarten nicht notwendig ist, alle in H.264 spezifizierten Elemente in jedem Encoder und Decoder zu implementieren, werden in der ersten Version des Standards drei verschiedene Profile definiert, die eine bestimmte Auswahl an Codierungs-Elementen enthalten: Baseline (BP), Extended (XP) und Main Profile (MP).

H.264 Profile			
Elemente	Baseline	Main	Extended
I- und P-Slices	X	X	X
CAVLC	X	X	X
CABAC		X	
B-Slices		X	X
Halbbild-Codierung (PAFF, MBAFF)		X	X
Übertragungsfehler-Robustheit (FMO, ASO, RS)	X		X
Erweiterte Übertragungsfehler-Robustheit (DP)			X
SP- und SI-Slices			X

Tab. 2: Profile bei H.264

Das Baseline Profil ist für Anwendungen mit geringer Verzögerungszeit und niedrigen Rechenkapazitäten optimiert und wird daher in erster Linie für Videokonferenzsysteme und mobile Applikationen eingesetzt.

Das Extended Profil unterstützt alle Elemente des Baseline Profils, zusätzliche Elemente verbessern die Kompressionsrate und den Fehlerschutz. Dieses Profil ist für Streaming-

Anwendungen vorgesehen und bietet mit SP- und SI-Slices auch die Möglichkeit zum Wechseln zwischen verschiedenen Serverströmen.

Als Nachfolger von MPEG-2 für Broadcast- und Speicheranwendungen war ursprünglich das Main Profil gedacht. Inzwischen wird diese Rolle aber größtenteils vom High Profil aus der erweiterten Version von H.264 (FRExt, 2004) übernommen.

Zusätzlich zu den verschiedenen Profilen definiert der Standard bestimmte Levels, die durch verschiedene Parameter wie Bitrate, Bildfrequenz, Wertebereich der Bewegungsvektoren oder die Größe des Referenzbildpuffers abgegrenzt werden.

Levels bei H.264

Level	Typ. Bildgröße	Typ. Bildfrequenz	Max. Bitrate (für VCL) bei Nicht-FRExt-Profilen*	Max. Anzahl an Referenzbildern für typ. Bildgröße
1	QCIF	15	64 kbit/s	4
1b	QCIF	15	128 kbit/s	4
1.1	CIF oder QCIF	7,5 (CIF) / 30 (QCIF)	192 kbit/s	2 (CIF) / 9 (QCIF)
1.2	CIF	15	384 kbit/s	6
1.3	CIF	30	768 kbit/s	6
2	CIF	30	2 Mbit/s	6
2.1	HHR (480i od. 576i)	30 / 25	4 Mbit/s	6
2.2	SD	15	4 Mbit/s	5
3	SD	30 / 25	10 Mbit/s	5
3.1	1280x720p	30	14 Mbit/s	5
3.2	1280x720p	60	20 Mbit/s	4
4	HD Formate (720p od. 1080i)	60p / 30i	20 Mbit/s	4
4.1	HD Formate (720p od. 1080i)	60p / 30i	50 Mbit/s	4
4.2	1920x1080p	60p	50 Mbit/s	4
5	2Kx1K	72	135 Mbit/s	5
5.1	2Kx1K od. 4Kx2K	120 / 30	240 Mbit/s	5

*Weil FRExt Profile speziell für Anwendungen mit sehr hohen Qualitätsansprüchen gedacht sind, gelten dort höhere maximale Bitraten: Die angegebenen Werte werden multipliziert mit 1,25 (High Profile), 3 (High 10) oder 4 (High 2:2:2, High 4:4:4).

Tab. 3: Levels bei H.264

4.4 Erweiterung 2004: FExt (Fidelity Range Extensions)

Nach Veröffentlichung der ersten Version von H.264 im Mai 2003 wurde an einer Erweiterung des Standards gearbeitet, um ihn auch für Anwendungen mit extrem hohen Qualitätsansprüchen wie die professionelle Filmproduktion, Postproduktion, High-Definition-Fernsehen und HD DVD zu optimieren. Diese Erweiterung wurde im September 2004 veröffentlicht und wird als FExt (Fidelity Range Extensions) bezeichnet.

4.4.1 8x8 Intra-Prädiktion, 8x8-Transformation

Die wichtigsten Änderungen in der neuen Version sind dabei die Möglichkeit einer 8x8-Intra-Prädiktion, und einer 8x8-Transformation, die für die Rekonstruktion von feinen Details vorteilhafter ist. In beiden Fällen wird adaptiv auf Makroblock-Ebene die beste Variante vom Encoder gewählt – Intra-Prädiktion entweder im 4x4-, 8x8- oder 16x16-Modus, Transformation entweder mit 4x4 oder 8x8 Koeffizienten.

Die 8x8-Transformation ist wie die 4x4-Transformation ganzzahlig, also ohne Rundungsfehler umkehrbar, und in der Umsetzung vom Rechenaufwand her kaum komplexer als die 4x4-Version.

4.4.2 Weitere Verbesserungen in FExt sind u.a.

- Sampletiefen bis 12 Bit
- Farbaufösungen bis 4:4:4
- besser an die Wahrnehmung angepasste, frequenzabhängige Quantisierung mit Skalierungsmatrizen → bessere subjektive Bildqualität
- effiziente verlustfreie prädiktive Kompression mit Hilfe einer relativ simplen Umgehung von Transformation und Quantisierung
- neue Arten von Zusatzdaten (SEI, supplemental enhancement information), die die Qualität des decodierten Videos verbessern können

In der FExt-Erweiterung sind 4 Profile definiert: High (HP), High 10 (H10P), High 4:2:2 (H422P) und High 4:4:4 Profile (H444P). In Tests zeigte sich das High Profile im Durchschnitt gegenüber Main Profile mit CABAC um 9,9%, gegenüber Main Profile mit CAVCL um 17,9% und gegenüber MPEG-2 MP@HL um 58,8% effizienter (vgl. [9], S. 3).

Das wichtigste Profil für Digitales Fernsehen und DVD wird nicht, wie von den Entwicklern des Standards ursprünglich vorgesehen, das Main Profil, sondern das FRExt High Profil sein. Das lässt sich daraus schließen, dass das High Profil als verpflichtend für die neuen Standards für DVB (Digital Video Broadcasting), die HD DVD (High Definition DVD) und die Blu-Ray Disc festgelegt wurde.

H.264 FRExt Profile				
Elemente	High	High 10	High 4:2:2	High 4:4:4
Main Profile Elemente	X	X	X	X
4:2:0 Farbunterabtastung	X	X	X	X
8 Bit Sampletiefe	X	X	X	X
Transformation adaptiv zwischen 8x8- und 4x4-Blocks	X	X	X	X
Quantisierung mit Skalierungs-Matrizen	X	X	X	X
Separate Cb und Cr Quantisierungsparameter	X	X	X	X
Monochromes Videoformat	X	X	X	X
9 und 10 Bit Sampletiefe		X	X	X
4:2:2 Farbunterabtastung			X	X
11 und 12 Bit Sampletiefe				X
4:4:4 Farbabtastung				X
Rest-Farb-Transformation				X
Verlustfreie prädiktive Kompression				X

Tab. 4: FRExt Profile

4.5 Scalable Video Coding (SVC)

Parallel zur Arbeit an der FRExt-Erweiterung des H.264-Standards wurden von ITU-T und MPEG ab Oktober 2003 Vorschläge für das Projekt „Scalable Video Coding“ (SVC), also skalierbare Videocodierung, entgegengenommen. Nach der Evaluierung von 14 Vorschlägen entschied man sich, das SVC-Konzept als Erweiterung des H.264-Standards zu verwirklichen (anstatt der Alternative, als Basis eine Wavelet-Transformation zu verwenden).

Skalierbarkeit bedeutet in diesem Zusammenhang, dass man bei einem einmal codierten Videosignal durch Weglassen von Teilen des Bitstroms die zeitliche Auflösung (Bildfrequenz), örtliche Auflösung (Bildgröße) und SNR (Signal-to-Noise-Ratio,

Signalrauschabstand, also objektive Qualität) reduzieren kann, bei entsprechender Steigerung der Effizienz bzw. Reduktion der Bitrate. Damit soll es möglich sein, mit ein und demselben Bitstrom Anwendungsbereiche mit unterschiedlichen Anforderungen zu versorgen. Die Kompressionsleistung sollte dabei mit einer herkömmlichen H.264-Codierung in der jeweiligen Auflösung vergleichbar sein.

Die Entwicklungsarbeit an dieser skalierbaren Erweiterung von H.264 ist so weit fortgeschritten, dass der Standardisierungsprozess planmäßig Anfang 2007 abgeschlossen sein sollte. Gearbeitet wird noch an einer SVC-Lösung für Videos im Halbbildformat, und an der Speicherung von SVC-Bitströmen im MP4-Dateiformat.

4.5.1 Layered Coding

Grundsätzlich funktioniert SVC so, dass die Codierung in mehreren Schichten (Layer) erfolgt. Der Basis-Layer repräsentiert die niedrigsten Auflösungsstufen und entspricht dem ursprünglichen H.264-Standard. „Enhancement Layers“ erweitern stufenweise die zeitliche, örtliche und SNR-Auflösung. Um Redundanzen zwischen den verschiedenen Layern zu nutzen, werden zusätzliche Inter-Layer-Prädiktionsmethoden angewandt.

4.5.2 Zeitliche Skalierbarkeit

Um mehrere Layer mit unterschiedlicher zeitlicher Auflösung zu erhalten, muss am ursprünglichen H.264-Standard nichts verändert werden. Dadurch, dass die Verwendung von Referenzbildern zur Bewegungsvorhersage sehr flexibel gestaltet werden kann, lässt sich eine hierarchische Prädiktionsstruktur als Grundlage für die zeitliche Skalierbarkeit einführen.

Der Basis-Layer mit der niedrigsten zeitlichen Auflösung wird dabei aus so genannten Schlüsselbildern (key pictures) gebildet, die ausschließlich vorangegangene Schlüsselbilder als Referenz benutzen. Zwischen den Key Pictures werden dann bidirektional vorhergesagte B-Bilder platziert. Die Gruppe von Bildern, die aus einem Schlüsselbild plus sämtlichen im zeitlichen Ablauf davor liegenden B-Bildern besteht, wird als GOP (Group of Pictures) bezeichnet. Diese Struktur lässt sich hierarchisch über mehrere Ebenen fortsetzen, wobei jeder Layer die Bildfrequenz verdoppelt.

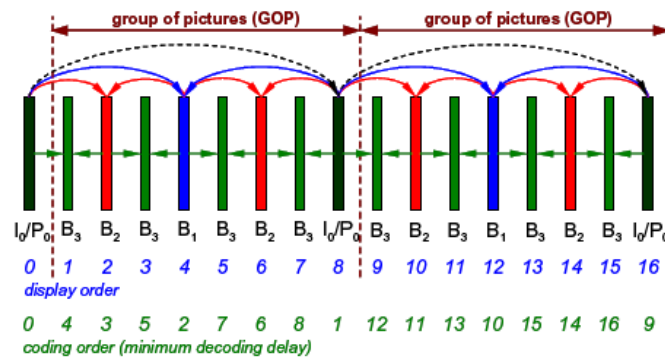


Abb. 16: Hierarchische B-Bild-Struktur mit 4 zeitlichen Auflösungen

Hierarchische GOP-Strukturen können auch so aufgebaut werden, dass sich kleinere Abstufungen in der Bildfrequenz ergeben.

Zusätzlich kann bei einer solchen hierarchischen Prädiktionsstruktur die Kompressions-Effizienz dadurch gesteigert werden, dass die Quantisierung bei jedem zusätzlichen Layer (dessen Bilder weniger oft als Referenz verwendet werden) vergrößert wird.

4.5.3 Örtliche Skalierbarkeit

Um verschiedene Bildauflösungen in einem Datenstrom unterzubringen, muss für jede erwünschte Bildgröße ein Layer mit eigenen Bewegungsinformationen separat codiert werden. Um SVC in diesem Bereich gegenüber einem Simulcast-Strom effizienter zu gestalten, können drei verschiedene Arten der Inter-Layer-Prädiktion verwendet werden, um Redundanzen zwischen den Layern zu nutzen.

Inter-Layer-Intra-Prädiktion

Die Werte eines Makroblocks in einem höheren Layer werden vorhergesagt, indem der rekonstruierte entsprechende Makroblock aus dem niedrigeren Layer hochgetastet (up-sampled) wird. Dazu wird derselbe Filter verwendet, der als Interpolationsfilter für die $\frac{1}{2}$ -Pixel-genaue Bewegungskompensation dient. Codiert wird dann der Prädiktionsfehler.

Inter-Layer-Bewegungsprädiktion

Bewegungsvektoren aus dem niedrigeren Layer können skaliert und als Vorhersagewerte für die aktuellen Bewegungsvektoren verwendet werden.

Inter-Layer-Prädiktionsfehler-Prädiktion

Hier wird der Vorhersagefehler, also das Differenzsignal, aus dem unteren Layer blockweise hochgetastet und zur Prädiktion des Differenzsignals des aktuellen Layers verwendet.

Welche Art der Inter-Layer-Prädiktion eingesetzt wird, kann jeweils der Encoder entscheiden, um eine optimale Effizienz zu erreichen.

4.5.4 SNR-Skalierbarkeit

Grobe Qualitätsabstufungen (Coarse-Grain Scalability)

Bei der SNR-Skalierbarkeit gibt es einerseits die Möglichkeit von groben Abstufungen durch Layer mit unterschiedlich feiner Quantisierung (Layered SNR Scalability oder Coarse-Grain Scalability, CGS). Für den Basis-Layer wird ein relativ grober Quantisierungsparameter eingesetzt. In jedem Enhancement Layer wird dann jeweils der Quantisierungsfehler codiert, jedoch mit einer feineren Quantisierung. Inter-Layer-Prädiktion kann genauso wie bei den örtlich skalierbaren Layern eingesetzt werden.

Feine Qualitätsabstufungen (Fine-Granular Scalability)

Eine feinere Abstufung bei der SNR-Skalierbarkeit lässt sich durch den Einsatz von so genannten Progressive Refinement (PR) Slices erreichen. Bei jedem Layer wird der Quantisierungsparameter um 6 vermindert, was einer Halbierung der Quantisierungsstufen entspricht. Innerhalb der NAL-Units werden zuerst die grob quantisierten Koeffizienten des Basis-Layers und danach die Daten zur Verfeinerung (Refinement) der Qualität codiert. Diese Daten sind absteigend nach ihrer Bedeutung für die Bildqualität angeordnet, sodass der Bitstrom an einer beliebigen Stelle abgebrochen werden kann, um eine gewünschte Qualität (und somit Datenrate) zu erreichen.

4.6 Was kommt nach H.264?

Ähnlich wie bei MPEG-2, bei dem die Kompressionseffizienz seit der Einführung erheblich gesteigert werden konnte, erwarten Experten, dass auch H.264 sein volles Potenzial im Laufe der kommenden Jahre erst nach und nach ausschöpfen wird (vgl. [14]).

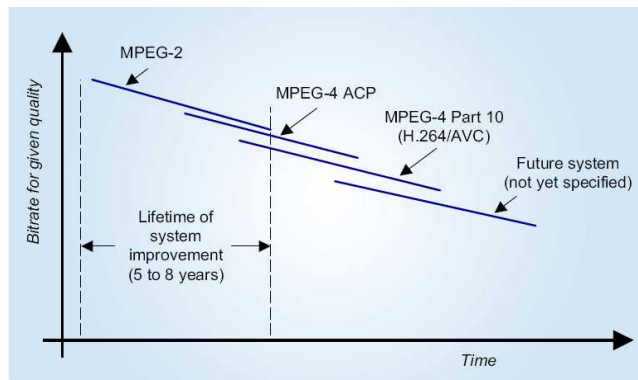


Abb. 17: Die Evolution von Systemen mit offenen Videokompressionsstandards

Obwohl H.264 bereits als fixer Bestandteil sowohl bei der nächsten Generation des digitalen Fernsehens als auch bei den Nachfolgern der DVD feststeht, wird sich die Übergangszeit bis zur vollständigen Ablösung wahrscheinlich über mehrere Jahre erstrecken. Für Broadcaster und Hersteller von DVD-Playern, Set-Top-Boxen etc. bedeutet die Umstellung einen hohen Investitionsaufwand, deshalb besteht seitens der Industrie keine Eile, mit der Entwicklung eines neuen Standards zu beginnen.

Die Forschungsarbeiten der Mitglieder von VCEG und MPEG gehen natürlich trotzdem ständig weiter. Die konkrete Ausschreibung der Entwicklungsziele, also sozusagen ein Startschuss für H.265 ist frühestens 2007 zu erwarten, mit einem fertigen neuen Standard rechnet man derzeit nicht vor 2008 – 2010 (vgl. [32]).

Innerhalb der MPEG gibt es beispielsweise eine Gruppe, die versucht, auf Wavelet-Transformation basierende Kompressionsmethoden zu optimieren (eine waveletbasierte Codierung war schon in MPEG-4 Visual vorgesehen, allerdings nur für Standbilder; auch JPEG 2000 arbeitet mit Wavelet-Transformation). Neben anderen Vorteilen wäre bei einer solchen Methode eine örtliche Skalierbarkeit schon allein durch den Transformationsmechanismus gegeben, außerdem würde die Unterteilung in Makroblocks bei der Transformation und damit auch die quantisierungsbedingten Block-Artefakte wegfallen.

Andere Ansätze, die in Zukunft helfen könnten, Datenraten weiter zu reduzieren, folgen dem objektbasierten Konzept von MPEG-4 Visual. Texturen mit vielen Details, die sehr schlecht komprimierbar sind, aber für die Darstellung des Bildes eigentlich unwichtig, können von den relevanten Bildteilen getrennt und mit Hilfe einer Texturanalyse und – Synthese zwar weniger korrekt, aber sehr viel datensparender übertragen werden.

Eine weitere Möglichkeit ist eine Kombination von modellbasierter (Model Based Coding) mit herkömmlicher Codierung. Dabei wird im Encoder eine dem Bildinhalt angepasste 3D-Struktur erstellt und durch Animation für die Bewegungsvorhersage verwendet. Diese Methode hat in Versuchen eine Datenreduktion um ca. 45% ermöglicht, funktioniert allerdings nur bei „regulären“ und einfachen Bildinhalten, für die 3D-Modelle vorgefertigt werden können, wie z.B. einem Nachrichtensprecher (vgl. [41]).

5 Lizenzierung von Offenen Standards

Obwohl die Bezeichnung „Offener Standard“ eigentlich das Gegenteil andeutet, ist die Verwendung von Videokompressionsstandards mit Lizenzierungskosten verbunden. Die Standardisierungsinstitutionen empfehlen ihren Gremien zwar den Einsatz von patentfreien Technologien, aber im Bereich der Videocodierung würde man damit nicht weit kommen.

Die Firmen, die ihr Know-How und ihre Forschungsarbeit in die Entwicklung von Standards stecken, möchten natürlich dafür auch entsprechende Gewinne erzielen. Um trotzdem einen „freien“ Zugang zu standardisierten Technologien zu ermöglichen, dürfen nur Patente verwendet werden, deren Inhaber ein sogenanntes „RAND Statement“ unterzeichnen. Der englische Wortlaut *“under reasonable terms and conditions that are demonstrably free of discrimination”* bedeutet, dass sich der Patentinhaber einverstanden erklärt, seine Technologie unter angemessenen Bedingungen und ohne Diskriminierung zur Verfügung zu stellen.

Diese „angemessenen“ Bedingungen sind aber nicht näher erläutert, und waren besonders bei der Lizenzierungslösung für MPEG-4 sehr umstritten.

Weil die Standardisierungsorganisationen sich nicht um Lizenzierungsangelegenheiten kümmern können, Einzelverhandlungen mit jedem Patentinhaber bei einem Videostandard aber zu aufwändig wären, wurde für MPEG-2-Lizenzen 1996 der Patentpool MPEG-LA gegründet. Er verwaltet im Fall von MPEG-2 mehrere hundert Einzelpatente, die sich auf über 20 Firmen aufteilen. MPEG-LA verwaltet auch MPEG-4 und H.264-Lizenzen, wobei es für H.264 auch einen zweiten Pool, Via Licensing, gibt.

Keiner der beiden Lizenzpools garantiert die Vollständigkeit, alle in einem Standard inkludierten Patente anbieten zu können – jedem Patentinhaber steht es frei, sich einem Pool anzuschließen oder nicht. Theoretisch ist es daher jederzeit möglich, dass ein Lizenzinhaber, der während der Entwicklung eines Standards nicht deklariert hat, dass seine Technologie verwendet wird, nachträglich Forderungen (die keineswegs angemessen oder nicht-diskriminierend sein müssen) an die Anwender eines Standards stellt.

Bei MPEG-2 war diese Vorgangsweise sehr erfolgreich. Die Lizenzierung wurde relativ unkompliziert gestaltet (ein bestimmter Betrag, heute US-\$ 2,50, pro hergestelltem DVD-

Player oder sonstigem Gerät, das einen Encoder und/oder Decoder enthält), und durch die breite Annahme des Standards erzielten die Patentinhaber große Gewinne.

Die Lizenzbedingungen für MPEG-4 waren um einiges komplizierter, und es wurden erstmals „usage fees“, also Benutzungsgebühren, eingeführt. Zusätzlich zu den Gebühren, die pro Encoder- und Decoder-Einheit anfallen, wird sozusagen pro übermitteltem Videoinhalt, je nach Länge des Videos und Anzahl der Empfänger, kassiert. Potenzielle Lizenznehmer waren mit diesen neuen Gebühren sehr unzufrieden und versuchten den MPEG-LA Pool durch öffentlichen Druck zu einer Änderung zu zwingen.

Besonders der Apple-Konzern, der stark in die Entwicklung von MPEG-4 eingebunden war, brachte seinen Streit um die Lizenzgebühren mit MPEG-LA an die Öffentlichkeit und verzögerte die Veröffentlichung seines Quicktime 6 Players, der einen MPEG-4-Codec enthält.

MPEG-LA änderte schließlich die ursprünglich geplanten Lizenzbedingungen, aber zu spät und zu wenig umfassend, um den entstandenen negativen Eindruck noch abzufangen. Diese Entwicklung ist einer von mehreren Gründen dafür, dass MPEG-4 nie an den Erfolg von MPEG-2 anschließen konnte.

Bei der Entwicklung von H.264 waren Lizenzierungsfragen schon im Vorfeld der Standardisierung ein Thema, weil alle Beteiligten ein Debakel wie bei MPEG-4 vermeiden wollten. Man bemühte sich sogar, das H.264 Baseline-Profil ganz frei von Lizenzgebühren zu halten, indem man dafür nur Technologien einsetzt, deren Patentinhaber eine Art Verzichtserklärung auf den Lizenzgebührenanspruch unterzeichnen.

Leider konnte sich diese Idee letztendlich nicht durchsetzen, und sogar die unpopulären „usage fees“ wurden beim MPEG-LA-Pool ähnlich wie bei MPEG-4 Visual geregelt. Die Lizenznehmer haben sich aber offensichtlich inzwischen mit diesem Konzept abgefunden.

Weder der MPEG-LA- noch der Via-Licensing-Pool decken alle Patente ab, die dem H.264-Standard zugrunde liegen. Lizenznehmer müssen als Alternative zu Einzelverhandlungen mit jedem der über 20 Patentinhaber Verträge mit beiden Pools abschließen und zusätzlich mit jenen Unternehmen verhandeln, die sich bis jetzt keinem der beiden Pools angeschlossen haben.

5.1 Sonderfall: SMPTE 421M / VC-1

Microsoft bietet mit der Windows Media Serie seit Jahren proprietäre Codecs für audiovisuelle Inhalte an, die besonders bei Streaming- und anderen Internet- und PC-basierten Anwendungen den offenen Standards starke Konkurrenz bieten.

Seit sich gezeigt hat, dass die Nachfolge für den MPEG-2-Standard keine klare Sache ist, war Microsoft bemüht, mit seinen Codecs auch in den Broadcast- und DVD-Bereich vorzudringen. Der jüngste Videocodec Windows Media Video 9 (WMV9) ist in Qualität und Kompressionsleistung mit H.264 vergleichbar, war aber schon früher und zu günstigeren Lizenzbedingungen verfügbar.

Seit 2003 arbeitete Microsoft ausgehend von WMV9 an einer Standardisierungsspezifikation, die schließlich Anfang 2006 von der amerikanischen SMPTE (Society of Motion Picture and Television Engineers) als offizieller Standard unter dem Namen VC-1 bzw. SMPTE 421M veröffentlicht wurde. Die Standardisierung und damit Offenlegung des Codecs war u.a. eine Voraussetzung dafür, dass VC-1 neben H.264 als verpflichtender Standard für HD DVD und Blu-Ray Disc festgelegt wurde.

Die Lizenzgebühren für den WMV9-Codec bestimmt Microsoft alleine ohne Rücksicht auf eventuell dafür verwendete patentpflichtige Technologien, und kann damit die offenen Standards preislich unterbieten. Durch die Offenlegung ihrer Dokumente in der VC-1-Standardisierung können jedoch jetzt alle Patentinhaber Ansprüche erheben. Die Lizenzierung von VC-1 wird zukünftig so wie bei den MPEG-Standards über den MPEG-LA Lizenzpool möglich und daher auch nicht billiger als H.264 sein.

6 Qualität eines Videocodecs

Um verschiedene (verlustbehaftete) Codecs bewerten und vergleichen zu können, muss die Kompressionsleistung einem Qualitätsmaß gegenüber gestellt werden. Ein Problem dabei ist, dass die Bewertung nach objektiven Kriterien nicht immer mit dem subjektiven Qualitätsempfinden des menschlichen Auges übereinstimmt.

Sowohl der Kompressionsfaktor selbst als auch die objektive und subjektive Qualität eines Codecs sind außerdem sehr stark vom Bildinhalt abhängig. Tests sollten daher immer mit mehreren verschiedenartigen Videosequenzen durchgeführt werden, bzw. wenn es um ganz spezifische Anwendungsbereiche geht, sollte man die Testvideos dementsprechend auswählen.

6.1 Objektive Bewertung

Das am weitesten verbreitete objektive Maß zum Leistungsvergleich ist die „Rate-Distortion“-Kurve, die die Verzerrung des Videosignals im Verhältnis zur Bitrate zeigt.

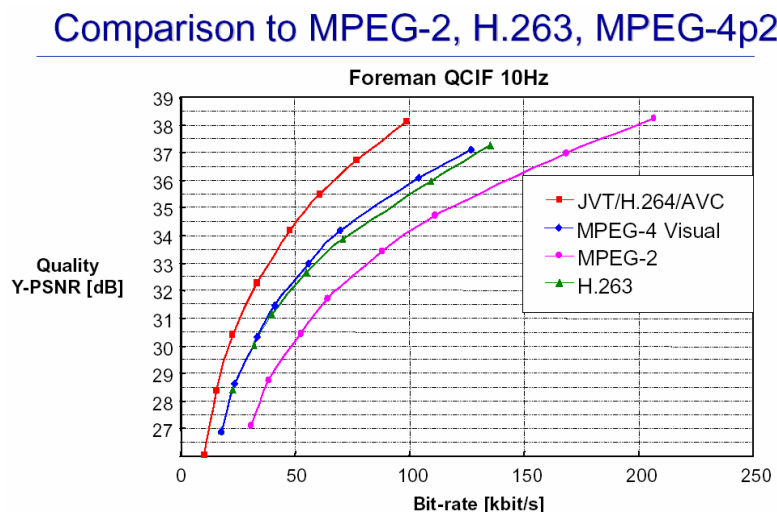


Abb. 18: Beispiel einer Rate-Distortion-Kurve

Die Verzerrung wird dabei als PSNR (peak-signal-to-noise-ratio, Spitzen-Signal-Rausch-Verhältnis) in dB angegeben.

$$PSNR = 10 \cdot \log_{10} \left(\frac{x_{pp}^2}{MSE} \right) \quad [\text{dB}]$$

x_{pp} ... *Spitzenwert*

MSE ist der Mittlere quadratische Fehler (mean square error):

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^N (x_i - \hat{x}_i)^2$$

$x[n]$... *Originalsignal mit N Abtastwerten*

$x[n] = (x_1 x_2 \dots x_i \dots x_N)$

$\hat{x}[n]$... *rekonstruiertes Signal*

Dieses und ähnliche objektive Qualitätsmaße können ohne weiteren Aufwand von Computerprogrammen berechnet werden, die das Original- mit dem rekonstruierten Signal vergleichen.

Besonders bei hohen Kompressionsraten treten deutlich wahrnehmbare Arten der Verzerrung verstärkt auf – sichtbar als Unschärfen, Blockstrukturen, Überschwinger an Helligkeitskanten („Ringing“), Verschmieren der Farbe, etc. – deren direkter Einfluss auf den subjektiven Qualitätseindruck nur unzureichend durch einen PSNR-Wert dargestellt werden kann.

6.1.1 Video Quality Experts Group

Es gibt seit Jahren sehr aufwändige Bemühungen, das menschliche Wahrnehmungssystem in objektive Qualitätsmessungen mit einzubeziehen, aber bisher konnte sich keine Methode durchsetzen, die einen vollständigen Ersatz für subjektive Tests bietet. Seit 1997 beschäftigt sich eine eigene Arbeitsgruppe innerhalb der ITU, die sog. Video Quality Experts Group (VQEG) mit diesem Thema.

Bis jetzt wurden zwei Standards für objektive Bemessungsmethoden von wahrnehmungsbezogener Videoqualität („objective perceptual video quality measurement techniques“) erarbeitet: ITU-T J.144 (2001 bzw. 2004) für digitales Kabelfernsehen, und ITU-R BT.1683 (2004) für digitales Fernsehen in Standard

Definition. Darin werden verschiedene Videoqualitätsmaße (Video Quality Metrics - VQM) beschrieben und Empfehlungen für deren Anwendung gegeben.

Die in diesen Standards beschriebenen Methoden beziehen sich auf digitales Fernsehen, also speziell auf Qualitätsvergleiche für MPEG-2-codiertes Videomaterial auf Fernsehbildschirmen, und sind daher für die unterschiedlichen Bedingungen im Multimedia-Bereich nur beschränkt anwendbar.

Die VQEG arbeitet zur Zeit daran, auch für Multimedia-Anwendungen entsprechende Testmethoden auszuwerten. Besonders bei Anbietern von mobilen Video-Services besteht ein großes Interesse an brauchbaren Instrumenten zum Qualitätsvergleich zwischen unterschiedlichen Videocodecs (vgl. [36]).

Die Schwierigkeit dabei ist die extreme Vielfalt an Parametern, die in Betracht gezogen werden müssen: Verschiedene Bildgrößen (CIF, QCIF, VGA,...), verschiedene Bildschirme und Displays, verschiedene Betrachtungsabstände, Bildfrequenzen zwischen 30 Bilder/s und 2,5 Bilder/s, verschiedene Codecs mit vielen verschiedenen Einstellungsmöglichkeiten. Dazu kommt, dass die wahrgenommene Qualität sehr stark vom Inhalt des Videos und auch von der individuellen Betrachtungssituation abhängt.

Unter Umständen können subjektive Tests auch völlig unvorhersehbare Ergebnisse hervorbringen, wie zum Beispiel einen höheren subjektiven Qualitätswert bei geringerer Bildfrequenz (vgl. [47]). Wie gut auch immer ein zukünftiger Standard zur objektiven Qualitätsbestimmung die menschliche Wahrnehmung nachempfinden kann, in Einzelfällen werden Testreihen mit Versuchspersonen abweichende Ergebnisse erzielen.

6.2 Subjektive Bewertung

Zur subjektiven Qualitätsbewertung gibt es verschiedene Verfahren, bei denen Testpersonen eine Reihe von Videosequenzen beurteilen.

Der ITU-R BT.500 Standard legt verschiedene subjektive Testmethoden für die Qualitätsbeurteilung von Fernsehbildern fest. Bei der am öftesten eingesetzten Methode DSCQS (Double Stimulus Continuous Quality Scale) werden jeweils zwei kurze Sequenzen gleichen Inhalts hintereinander einer Testperson präsentiert, eine in Originalqualität, die andere komprimiert, in zufälliger Reihenfolge. Die Qualität der Sequenzen wird auf einer kontinuierlichen Skala mit fünf Intervallen, von schlecht bis

exzellent (bad – poor – fair – good – excellent), beurteilt. Das Ergebnis einer solchen Testreihe nennt man Mean Opinion Score (MOS), das die relative Qualität der komprimierten und der Originalsequenz mit einem Wert zwischen 0 und 100 (volle Qualität) wiedergibt. Wird ein MOS-Wert von 80 erreicht, spricht man von „transparenter Qualität“, was bedeutet, dass die codierte Sequenz qualitätsmäßig nicht vom Original unterscheidbar ist.

Verschiedene andere Arten der subjektiven Bewertung unterscheiden sich nur unwesentlich von der DSCQS-Methode, es wird zum Beispiel keine Referenz in Originalqualität gezeigt, oder die Referenzsequenz explizit als solche gekennzeichnet. Es ist auch möglich, mehrere verschieden codierte Testsequenzen zu vergleichen, oder auch zwei Sequenzen gleichzeitig nebeneinander abspielen zu lassen.

Eine Projektgruppe der EBU (European Broadcasting Union), Broadcast/Video in Multimedia (B/VIM) beschäftigt sich seit 2002 mit der Bewertung von subjektiver Videoqualität speziell für den Multimedia-Bereich. Eine speziell dafür entwickelte Testmethode, SAMVIQ (Subjective Assessment Methodology for Video Quality), wurde 2003 der ITU-R zur Standardisierung vorgeschlagen.

7 H.264 in der Praxis

Schon knapp ein Jahr nach seiner Veröffentlichung wurde der H.264-Standard von einigen seiner Entwickler als *„weltweiter Nachfolgestandard von MPEG-2 für die Bereiche Digital Video-Broadcast, Video-on-Demand über xDSL und TV/HDTV/Film-Aufzeichnung auf DVDs“* angesehen, der darüber hinaus in *„nahezu allen Bereichen der Informations- und Kommunikations-Technologie, bei denen Video eine Rolle spielt, zum Einsatz kommen wird.“* (Pressemitteilung des Fraunhofer Instituts für Nachrichtentechnik / Heinrich Hertz Instituts, vgl. [53]).

Auch der Apple-Konzern setzte sehr früh auf eine solche Entwicklung. Noch vor der Veröffentlichung des neuen QuickTime 7 Players (April/Mai 2005), der mit dem Apple H.264 Codec den Standard als erster ins Internet und damit an eine große Anzahl von Anwendern brachte, prophezeite Frank Casanova, der Marketing-Chef von QuickTime, *„H.264 is the future of broadcast – it will replace the older MPEG technologies. Because it's a standard there are no other encroaching technologies – nothing else out there really matters.“* (vgl. [23])

Da es ähnliche große Erwartungen bereits für MPEG-4 Visual gegeben hatte, sind andere Stimmen vorsichtiger. Anfang 2005 meinte Tim Siglin in streamingmedia.com noch, dass H.264 in den folgenden Monaten erst seine Fähigkeiten unter Beweis stellen müsse: *„The next few months are critical for H.264 proponents, as claims about the scalability of H.264 from mobile telephony to high-definition broadcast need to be proven. Specifically, advocates will need to prove to sceptical users and developers that one codec has the potential to replace multiple codecs, scaling the spectrum from low-bandwidth to very high bandwidth while simultaneously encompassing videoconferencing, streaming, progressive downloads and physical media playback.“* (vgl. [20])

Selbst als H.264 schon als fixer Bestandteil der HD DVD- und Blu-Ray Disk- sowie der DVB-S2-Spezifikationen fest stand, waren manche Experten mit Prophezeiungen vorsichtig: *„Das Jahr 2006 könnte also der Auftakt für den Siegeszug von H.264 werden,“* meinte etwa Dominik Schöneberg im Dezember 2005 in netzwelt.de ([22]).

Aus den Zahlen der H.264-Lizenznehmer bei MPEG LA lässt sich ein solcher Siegeszug (noch) nicht eindeutig feststellen. Im Juli 2006 beträgt die Anzahl der Lizenznehmer 204. Zum Vergleich: Bei MPEG-4 Visual sind es zum selben Zeitpunkt 332, bei MPEG-2 1080.

In diesem Kapitel soll überblicksmäßig gezeigt werden, wie weit der H.264-Standard Mitte 2006 die großen Erwartungen, die seit seiner Veröffentlichung 2003 mit ihm verbunden waren, bereits erfüllt hat, bzw. in welchen Bereichen ein durchschlagender Erfolg schon absehbar ist. Die Liste an Beispielen von H.264 im praktischen Einsatz ist keinesfalls vollständig und soll in erster Linie einen treffenden Eindruck der bisherigen und zukünftigen breiten Akzeptanz des jüngsten Videokompressions-Standards bieten.

7.1 Vorgesehene Einsatzgebiete

Einfach gesagt, war der H.264-Standard von Anfang an für eine maximale Vielfalt an verschiedenen Anwendungsbereichen vorgesehen, sollte also überall dort zum Einsatz kommen, wo mit digitalem Video gearbeitet wird. Über die zum Zeitpunkt der Spezifikation etablierten Video-Anwendungen hinaus wurde eine Flexibilität angestrebt, die auch eine Anpassung an alle zukünftig entwickelten neuen Technologien möglich machen sollte.

Zu den ausdrücklich vorgesehen Einsatzgebieten von H.264 (vgl. [18]) gehören

- Digitales Fernsehen
- Interaktive oder serielle Speicherung auf optischen/magnetischen Datenträgern
- Kommunikationsanwendungen (ISDN, mobil, VoIP, ...)
- PC-basierte Anwendungen wie Multimedia-Streaming, Video-on-Demand
- Multimedia Messaging Services (MMS)

Die Bereiche überschneiden sich teilweise und basieren auf einer Reihe unterschiedlicher Übertragungswege – Kabel, Satellit, terrestrischer Rundfunk, DSL, ISDN, Ethernet, LAN, drahtlose und mobile Netzwerke, Modems, etc. oder einer Mischung davon.

7.2 Systemstandardisierungen zur Einbettung von H.264

Standardisierungsarbeit zur Einbettung von H.264-codierten Videodaten, die für unterschiedliche Anwendungsbereiche maßgeblich ist, wurde u.a. von den folgenden Gremien geleistet: Von der MPEG zur Verpackung von H.264-Daten in MPEG-2-Transportströme sowie ins MPEG-4-Dateiformat; von der ITU-T im Standard H.241 zur Anwendung in H.320, H.323 und H.324-Transportströmen (Videotelefonie/-konferenz);

und von der IETF (Internet Engineering Task Force) im Dokument RFC 3984 zur Übertragung von H.264 mittels Real-time Transport Protocol (RTP).

7.3 Digitales Fernsehen

Von der ETSI (European Telecommunications Standards Institute) wurde die DVB- (Digital Video Broadcast-) Richtlinie TS 101 154 im Jänner 2005 dahingehend erweitert, dass neben MPEG-2 auch H.264-codierte Videoströme in den vorgeschriebenen MPEG-2-Transportströmen verwendet werden können. MPEG-2-Transportströme sind die Grundlage für alle DVB-Spezifikationen: DVB-S (Satellit), DVB-C (Kabel), DVB-T (terrestrisch) und DVB-H (handheld).

Ein weiteres ETSI-Dokument, TS 102 005, spezifiziert den Transport von Video über IP (Internet Protocol) für IPTV, also Fernsehen, das als Alternative zu Kabel oder Satellit über xDSL-Leitungen übertragen wird, und zwar mittels IP. Diese Form der Fernsehübertragung, die durch den automatisch vorhandenen Rückübertragungsweg auch für alle interaktiven Anwendungen ideal geeignet ist, ist noch nicht sehr weit verbreitet und stellt ein großes Wachstumspotenzial in den nächsten Jahren dar. In der ETSI-Spezifikation für IPTV wurde H.264 als einziger (verpflichtender) Standard zur Videocodierung festgesetzt.

Auch das amerikanische ATSC (Advanced Television Systems Committee) hat H.264 in seine Standards für digitales Fernsehen aufgenommen. Die ISMA (Internet Streaming Media Alliance, arbeitet unter anderem derzeit an einem internationalen Standard für IPTV, bei dem H.264-Kompression ebenfalls verpflichtend zum Einsatz kommen wird.

7.3.1 IPTV

Die weltweit erste Fernsehübertragung mittels H.264 wurde im April 2005 vom britischen IPTV-Anbieter HomeChoice durchgeführt [63], der anschließend bis August 2005 sein gesamtes Angebot auf H.264 umgestellt hat. Einige große Telecom-Unternehmen, die mittlerweile H.264 im Bereich IPTV einsetzen sind der niederländische KPN-Konzern (ehem. staatliche Telekom), die belgische Belgacom und der größte US-Telekom-Anbieter SBC Communications.

7.3.2 Satellitenfernsehen

Anfang 2005 kündigten große Satellitenbetreiber wie DirecTV (USA) und BSkyB (UK, Irland), und der erste europäische HDTV-Sender Euro 1080 an, ihr HDTV-Angebot zukünftig auf H.264-Codierung zu basieren. Seit Herbst 2005 werden bereits lokale Programme von DirecTV in Amerika, und die HD-Programme von ProSieben, Sat.1 und Premiere in Europa H.264-codiert ausgestrahlt, seit Mai 2006 Sky-HD in England. Vorerst können diese Programme allerdings noch von relativ wenigen Sehern empfangen werden, weil bis jetzt (Mitte 2006) das Angebot an H.264-kompatiblen Sat-Receivern und DVB-S2-Karten sehr gering ist.

7.3.3 DVB-T

Terrestrische Übertragung von H.264-codierten Fernsehsignalen wird seit Juni bzw. Juli 2006 in einzelnen Städten in England und Frankreich versuchsweise durchgeführt [66]. In mehreren anderen europäischen Ländern steht H.264 als Basis für geplante DVB-T-Netze bereits fest.

Allein zur besseren Kapazitätsnutzung, aber auch speziell zur Erweiterung des HD-Angebots ist zu erwarten, dass sich im Verlauf der nächsten Jahre H.264-codiertes Fernsehen, sowohl bei der Satelliten-, als auch Kabelausstrahlung zunächst neben MPEG-2 etabliert und dieses schließlich ablösen wird. In Märkten, wo DVB-T neu eingeführt wird, wird man in Zukunft vermutlich größtenteils von Anfang an auf H.264-Kompression setzen.

7.3.4 Mobiles Fernsehen

Die beiden konkurrierenden Standards zur mobilen Fernsehübertragung („Handy-Fernsehen“), DVB-H (handheld) und T-DMB (terrestrial Digital Multimedia Broadcast), basieren ausschließlich auf H.264-Codierung. DMB-Fernsehen gibt es in Südkorea schon seit Anfang 2005. Seit Mai 2006 überträgt watcha [67] in Deutschland 5 Fernsehkanäle im T-DMB-Standard. In Frankreich und England laufen Mitte 2006 DMB-Tests [68]. Auch DVB-H wird derzeit (Juni 2006) u.a. in einigen deutschen Städten getestet [69]. Obwohl laut Meinungsumfragen das Interesse am Handy-Fernsehen in

Deutschland vorerst nur gering ist ([70], Juni 2006), sehen Mobilfunkbetreiber hier ein großes Wachstumspotenzial.

Weitere Mobil-TV-Standards, die H.264-Codierung vorsehen, sind ISDB (Integrated Services Digital Broadcasting) der japanischen ARIB (Association of Radio Industries and Businesses) und MBMS (Multimedia Broadcast Multicast Service), das ein Teil der 3GPP-Spezifikationen ist (s. Mobile Anwendungen). Für mobiles Fernsehen kommt das H.264 Baseline Profil in den Levels 1b und 2 zum Einsatz.

7.4 Speichermedien

Für die beiden konkurrierenden Nachfolgeformate der DVD, HD DVD und Blu-Ray Disc (BD) die Videoinhalte in High Definition-Qualität liefern, steht H.264 (High Profil, Levels 4 und 4.1) neben VC-1 und MPEG-2 seit Juni (HD DVD) bzw. September (BD) 2004 als verpflichtender Videokompressions-Standard fest. Die Tatsache, dass bis Mitte 2006 kaum noch Abspielgeräte für oder Filme auf diesen neuen Datenträgern im Handel sind, liegt teilweise daran, dass die Industrie lange versuchte, sich auf eines der beiden Formate festzulegen, was allerdings nicht gelang. (Ein weiterer Grund für die Verzögerung waren Schwierigkeiten bei der Entwicklung eines wirksamen Kopierschutzmechanismus.)

Den ersten HD DVD Player gab es Ende März 2006 in Japan zu kaufen, Recorder soll es ab Juli 2006 geben. Erst im Juni 2006 kamen die ersten Blu-Ray Disc Brenner, der erste Blu-Ray Disc Player und Filme auf Blu-Ray Disc auf den Markt. Es ist sehr wahrscheinlich, dass es demnächst auch Bemühungen in die Richtung geben wird, Player, Laufwerke und Brenner zu entwickeln, die mit beiden Medien und Formaten umgehen können – einen ersten Schritt machte der Hersteller Ricoh, der ankündigte, im Juli 2006 erstmals ein System vorzustellen, dass sowohl HD DVD und Blu-Ray Disc als auch die herkömmlichen Medien CD und DVD lesen und beschreiben kann [57].

Ein proprietäres Speicherformat, das auf H.264-Komprimierung setzt, ist die UMD (Universal Media Disc), die eigens für die Sony PSP (PlayStation Portable) in erster Linie für Spiele entwickelt wurde, aber auch zum Abspielen von Video und Musik geeignet ist.

7.5 Aufzeichnung

Eine sehr neue Entwicklung ist AVCHD, das von Panasonic und Sony im Mai 2006 als neues Format für HD-Videoaufnahmen vorgestellt wurde. Es soll digitalen Camcordern durch H.264-Codierung die direkte Aufnahme von HD-Videosignalen auf herkömmlichen DVDs ermöglichen [78].

7.6 Kommunikation

Große Anbieter von Videokonferenzsystemen waren unter den ersten, die H.264 in ihren Produkten implementierten – darunter die norwegische Firma Tandberg, die unmittelbar nach der Ratifizierung des Standards im Mai 2003 ihre gesamte Produktpalette mit H.264-Unterstützung anbieten konnte [60]. Inzwischen ist H.264-Komprimierung bei allen Produkten im Bildtelefonie- und Videokonferenzbereich state-of-the-art. Zum Einsatz kommt hier das Baseline Profil mit Levels 1b bis 4.

7.7 Mobile Anwendungen

Im Bereich Mobilfunknetze haben die beiden großen Gruppen von nationalen Standardisierungsinstitutionen, 3GPP (3rd Generation Partnership Programme) und 3GPP2, H.264-Kompression in mehrere ihrer Richtlinien, die die Übermittlung von multimedialen Inhalten betreffen, mit einbezogen. Neben dem H.263-Standard, den 3GPP verpflichtend für alle Anwendungen (außer MBMS) vorschreibt, wird H.264 als optionaler Standard geführt. Für MBMS (Multimedia Broadcast Multicast Service) wird die Verwendung von H.264 empfohlen. Bei 3GPP2 ist H.264 ein optionaler Codec für Streaming-Services.

7.8 Computerbasierte Anwendungen

Die ersten professionellen H.264-konformen Software-Codecs gab es noch im Jahr der Veröffentlichung des Standards, 2003. Im September 2003 wurde beispielsweise von Vsofts (auch VSS, Vanguard Software Solutions) bereits ein H.264-Codec vorgestellt, der SD-Video in Echtzeit auf einem PC codieren und decodieren kann (vgl [87]).

Inzwischen gibt es über 30 verschiedene Software-Implementierungen des H.264-Standards, darunter auch einige Open-Source-Projekte ([88]). Der Open-Source-

Encoder x264 konnte in einem unabhängigen H.264-Codec-Vergleich der Moskauer Staatsuniversität im Dezember 2005 ([73]) gegenüber professionellen Produkten von Atime, ArcSoft, ATI, Elecard, Fraunhofer IIS und VSS hervorragende Werte erzielen.

Die beiden großen Grafikkartenhersteller ATI und Nvidia bieten seit Ende 2005 bzw. Anfang 2006 hardwarebeschleunigte H.264-Decodierung für ihre neueren Grafikkarten.

Den größten Verdienst bei der Bekanntmachung des H.264-Standards an eine große Öffentlichkeit hat wahrscheinlich der Apple-Konzern. Gleichzeitig mit dem neuen Betriebssystem OS X Tiger wurde Apple's H.264-Codec Ende April 2005 mit der Videokonferenzsoftware iChat und der Version 7 der QuickTime Mediensoftware eingeführt.

Den neuen Codec bewarb Apple als „Zukunft der Videotechnik“, der in allen Bereichen, von der Filmproduktion bis zur Videokonferenz, neue Maßstäbe an Qualität und Effizienz setzen würde ([27]). Tatsächlich konnten die auf der QuickTime-Homepage gezeigten Beispiele mit einer Bildqualität, die für im Internet veröffentlichte Videos neu war, beeindrucken. Mit dem Apple QuickTime 7 Player, der kurz nach seiner Veröffentlichung auch für Windows erhältlich war, ist der QuickTime H.264 Decoder inzwischen weltweit vermutlich auf mehreren hundert Millionen PCs und Macs installiert.

Neben der etablierten Verwendung von QuickTime in der professionellen Video- und Filmproduktion konnte man wie erhofft durch den neuen Codec einen vermehrten Einsatz von QuickTime für Video-Streaming im Internet erreichen, und damit erstmals einen offenen Standard erfolgreich in diesem Bereich platzieren. Die Konkurrenz ist hier allerdings nach wie vor sehr stark – Windows Media Video 9 bietet vergleichbare Qualität und Kompression, und der neueste RealVideo 10 Codec soll sogar um 15% effizienter sein als H.264 ([89]).

Seit Mai 2005 bietet Apple in seinem iTunes Music Store neben Musik auch H.264-codierte Musikvideos und Fernsehsendungen zum Download an, die man direkt am PC/Mac anschauen und/oder auf einen iPod übertragen und dort abspielen kann.

Die Plattform Google Video, auf der jedermann seine Videos frei oder gegen Bezahlung zum Streaming oder Download anbieten kann, verwendet einen eigenen Player, der auch H.264 unterstützt. Er basiert auf dem VLC-Media Player, der vom selben Team entwickelt wurde wie der x264-Codec.

Das sind nur einige Beispiele für computer- und internetbasierte Anwendungen des H.264-Standards. In diesem Bereich werden auch weiterhin eine Reihe proprietärer und standard-konformer Codecs nebeneinander existieren, von denen sich wahrscheinlich keiner als eindeutiger Marktführer durchsetzen wird. H.264 ist aber auch hier auf jeden Fall schon jetzt ein Standard, an dem kein Weg vorbei führt, und wird es für etliche Jahre bleiben.

7.9 Andere Anwendungen

Die 2003 gegründete Digital Living Network Alliance (DLNA), eine Organisation die sich um Standardisierung im Bereich digitaler Heimnetzwerke bemüht, hat in seinen im März 2006 veröffentlichten Interoperabilitäts-Richtlinien den H.264-Standard als verpflichtend für Video festgelegt.

Die Sony PSP (PlayStation Portable) und die neueren Versionen der Microsoft Xbox und des Apple iPod können H.264-komprimierte Videos abspielen.

Auch mehrere nationale und internationale Behörden haben H.264 zu ihrem favorisierten Videokompressionsstandard erklärt, darunter die NATO und das amerikanische Verteidigungsministerium ([37]).

8 H.264-Codecs im Test

Nach der ausführlichen theoretischen Beschreibung werden in diesem Kapitel zwei verschiedene Implementierungen des H.264-Standards, im Vergleich miteinander und zu konkurrierenden Codierungssystemen, in einer praktischen Testreihe untersucht.

Für die Tests wurde ein vom Grafik- und Videolabor der Moskauer Staatsuniversität entwickeltes Instrument („MSU Video Quality Measurement Tool“, Version 1.2) verwendet, das für nicht-kommerzielle Zwecke frei zur Verfügung gestellt wird und auf http://www.compression.ru/video/quality_measure/video_measurement_tool_en.html erhältlich ist.

Dieses Tool ermöglicht den Vergleich von Original- mit komprimierten Videosequenzen durch die Berechnung von verschiedenen Video-Qualitätsmaßen. Eines dieser Maße ist das bereits beschriebene PSNR-Maß (peak-signal-to-noise-ratio), das zur Qualitätsbeurteilung von komprimierten Videodaten am häufigsten herangezogen wird.

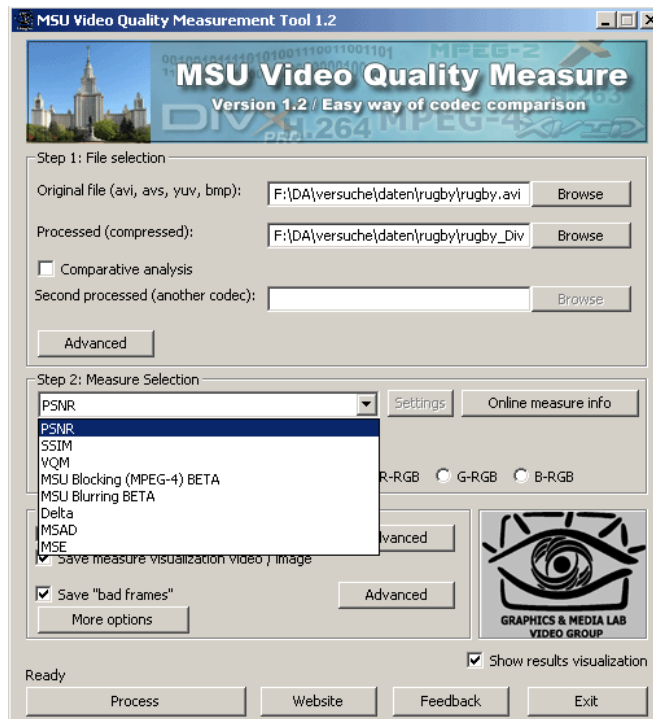


Abb. 19: MSU Video Quality Measurement Tool

Es können wahlweise ein einzelnes oder zwei verschieden codierte Videos gleichzeitig mit dem Original verglichen werden. Zur Aufzeichnung und Ausgabe der Daten gibt es

folgende Möglichkeiten: Unmittelbar nach der Berechnung können die Werte über die gesamte Sequenz grafisch dargestellt werden.

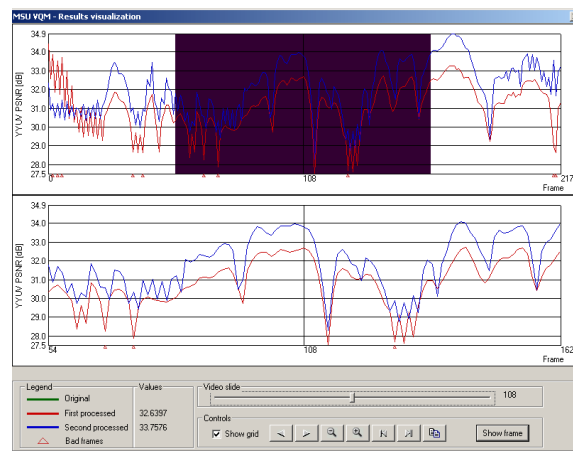


Abb. 20: Y-PSNR-Vergleich von zwei verschieden codierten Videosequenzen im MSU Video Quality Measurement Tool

Zusätzlich kann man sich bis zu 50 Frames anzeichnen lassen, bei denen die Werte besonders stark vom Original abweichen bzw. wo es die stärksten Unterschiede zwischen den beiden verglichenen codierten Sequenzen gibt. Diese „Bad Frames“ werden gleichzeitig auch als Einzelbilder abgespeichert. Alle errechneten Werte, sowohl die Einzelwerte für jedes Frame als auch der Durchschnittswert über die gesamte Sequenz, können in einer CSV-Datei zur weiteren statistischen Auswertung gespeichert werden.

Außerdem ist es möglich, das Ergebnis zu visualisieren, indem ein Video gespeichert wird, in dem für jedes einzelne Frame die errechnete Abweichung vom Original farblich gekennzeichnet ist – von schwarz für die beste Übereinstimmung über blau, grün, gelb bis zu rot für die stärkste Abweichung.

Weitere Maße, die das Tool berechnen kann, sind der „Structural Similarity Index“ (SSIM), und ein „Video Quality Metric“ (VQM), beides Qualitätsmaße, die näher an der menschlichen Wahrnehmung liegen als PSNR. SSIM wird aus einer Kombination von Helligkeits-, Kontrast- und struktureller Ähnlichkeit zwischen Original und komprimiertem Signal errechnet ([80]). Die Berechnung von VQM erfolgt mit Hilfe von DCT-Koeffizienten und ergibt im Gegensatz zu den beiden anderen Maßen einen geringeren Ergebniswert je höher die Qualität des codierten Videos ([48]).



Abb. 21: Frame 197 aus der Sequenz „City“, Original



Abb. 22: Frame 197 aus der Sequenz „City“, komprimiert mit DivX 6.2.5 bei 197 kbit/s

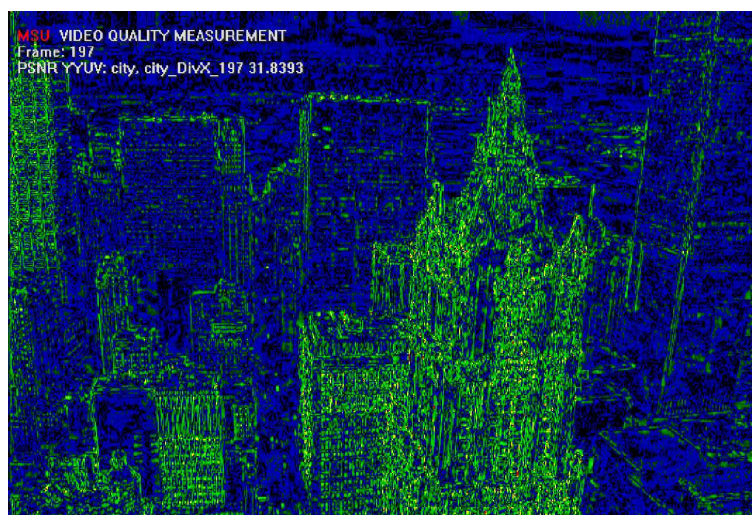


Abb. 23: Frame 197 aus der Sequenz „City“, komprimiert mit DivX 6.2.5 bei 197 kbit/s
Y-PSNR-Visualisierung mittels MSU Video Quality Measurement Tool

Nachdem selbst die SSIM- und VQM-Testergebnisse nicht mit dem subjektiven Qualitätseindruck der komprimierten Sequenzen übereinstimmt, besonders was störende Blockartefakte betrifft, wurden auch die Ergebnisse des im Video Quality Measurement Tool eingebauten Maßes für Blockartefakte („MSU Blocking“) überprüft, das sich laut Entwicklerteam noch im Beta-Stadium befindet, und über dessen Berechnungsalgorithmus abgesehen von der folgenden Beschreibung keine näheren Informationen zu finden waren:

“This metric was created to image visual measure of blocking. For example, in contrast areas of the frame blocking is not appreciable, but in smooth areas these edges are conspicuous. This metric also contains heuristic method for detecting objects edges, which are placed to the edge of the block. In this case metric value is pulled down, it allows to measure blocking more precisely. We use information from previous frames to achieve better accuracy.” ([90])

PSNR, SSIM und VQM können jeweils sowohl für die Y- (Helligkeits-) Komponente als auch für die einzelnen Farbkomponenten berechnet werden. Die durchgeführten Tests beschränken sich auf die Y-Komponente, da diese für die Wahrnehmung die größte Rolle spielt.

8.1 Testsequenzen

Die verwendeten Videosequenzen sind Testsequenzen im PAL- und NTSC-Format (720x625, bei 50 Halbbildern/s, 576 aktive Zeilen bzw. 720x525 bei 60 Halbbildern/s, 486 aktive Zeilen; Y/Cr/Cb-Komponentensignal mit 4:2:2-Abtastung, Länge jeweils 8 Sekunden + 20 Frames), die von der Video Quality Experts Group der ITU auf einem öffentlich zugänglichen Server zur Verfügung gestellt werden ([79]).



Abb. 24: Testsequenz „City“ (NTSC)

Hubschrauberflug über Manhattan. Langsame und relativ gleichmäßige seitliche Bewegung. Kamera ist etwas unruhig.



Abb. 25: Testsequenz „Rugby“ (PAL)

Sehr schnelle Bewegungen der Spieler und der Kamera.



Abb. 26: Testsequenz „Susi“ (NTSC)

Fast statisch außer Mimik (Augen, Mund), ein kurzes Kopfschütteln. Starkes Rauschen.



Abb. 27: Testsequenz „Wind“ (NTSC)

Sequenz beginnt mit gelber Pflanze in Großaufnahme, langsames gleichmäßiges Herauszoomen. Blätter wirbeln ständig durch das Bild.

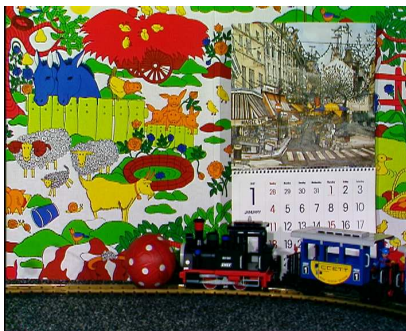


Abb. 28: Testsequenz „Zug“ (PAL)

Zug bewegt sich langsam, Ball rollt vor ihm her, Kamera folgt Zug. Kalender bewegt sich auf und ab. Sehr feine Details auf Kalender und Tapete.

8.2 Getestete Codecs

8.2.1 Apple QuickTime H.264

Mit der Integration des eigenen H.264-Codecs in das Betriebssystem OS X 10.4 Tiger und in den kostenlos auch für Windows-Systeme erhältlichen QuickTime 7 Player (Veröffentlichung Ende April 2005) hat Apple den H.264-Standard einer großen

Öffentlichkeit schmackhaft gemacht. Die kristallklaren Film-Trailer und andere Demo-Videos auf der QuickTime Homepage ([27]) übertrafen tatsächlich an Qualität und Kompressionsrate alles was bis zu diesem Zeitpunkt an komprimiertem Video im Internet abrufbar war. Bis jetzt, über ein Jahr später, stellt die QuickTime Videogalerie, die auch etliche Clips in hohen Auflösungen (bis 1980x1020) enthält, eine beeindruckende Demonstration der Leistungsfähigkeit von H.264 dar.

Mit einem Upgrade zur Pro-Version um € 29,99 bietet QuickTime einen relativ günstigen und sehr einfach zu bedienenden Encoder. Laut Apple unterstützt der QuickTime H.264 Codec das Main Profil des H.264-Standards, in Wirklichkeit ist es aber nur das Baseline Profil plus ein B-Frame ([81]). Auch der QuickTime-Decoder beherrscht nicht alle Features des Main Profils ([82]). Verwendet man also einen anderen als den QuickTime H.264 Codec zur Kompression, muss man die Encoder-Einstellungen besonders genau beachten, wenn die Videos mit dem QuickTime Player abspielbar sein sollen.

Für die Codierung der Testsequenzen wurden folgende Einstellungen des QuickTime H.264 Encoders (Version 7.0.4) verwendet:

Keyframes: automatisch

Bilder neu anordnen

Codierung: Beste Qualität (mehrere Durchgänge)

Datenrate: Beschränken auf ... kbit/s

Optimiert für: Laden

8.2.2 x264

Dieser freie H.264-Codec erreichte bei einem unabhängigen Test von verschiedenen H.264-Codecs, der vom Grafik- und Medienlabor der Moskauer Staatsuniversität im Dezember 2005 durchgeführt wurde, neben dem Atime H.264-Codec die besten Werte ([73]). Deshalb schien er als Repräsentant für die leistungsfähigsten Software-Implementierungen des H.264-Standards für einen Vergleich mit dem QuickTime H.264-Codec sehr gut geeignet.

Trotz der hervorragenden Testergebnisse wird von den Entwicklern von x264 eindeutig betont, dass sich der Codec noch in einem „frühen Entwicklungsstadium“ befindet („early development stage“, [72]). Es gibt beispielsweise noch keine Unterstützung für interlaced (Halbbild-) Videosignale.

Verwendet wurde die CLI-Version vom 4. August 2006 (rev. 551, [83]) gemeinsam mit der MeGUI-Benutzeroberfläche (Version 0.2.3.2185, [84]). Mit den x264-Standard Einstellungen und auch mit den von MeGUI „empfohlenen“ („recommended“) Einstellungen konnten zunächst keine optimalen Ergebnisse (im Vergleich zu DivX) erzielt werden. Die vielen einzeln einstellbaren Parameter wie etwa die Stärke des Deblocking-Filters, Anzahl der Referenzbilder, Keyframe-Intervall, GOP-Größe, verschiedene Makroblock-Unterteilungsoptionen etc. gestalten eine Suche nach den Einstellungen für die bestmögliche Qualität sehr mühsam.

Schließlich wurde das für MeGUI vorgefertigte Einstellungsprofil „HQ-Insane“ verwendet ([85], [86]), weil die damit erreichte Qualität mit keiner anderen ausprobierten Einstellung übertroffen werden konnte (die Ergebnisse wurden jeweils anhand ihrer Y-PSNR-Werte verglichen).

Die Einstellungen des HQ-Insane-Profiles sehen im Einzelnen so aus (soweit sie von den x264-Standardwerten abweichen):

Automated 2-pass

AVC Profiles: High Profile

AVC Level: Unrestricted

Deblocking Strength: -2

Deblocking Threshold: -1

M.E. Algorithm: 3 - Multi hex (M.E = Motion Estimation)

Subpixel Refinement: 7 – RDO Level 2

Macroblock Options: Custom (All)

Trellis Quantization: 2 – Always

Number of Reference Frames: 16

Mixed References: Enabled

No Fast B-Skip: Enabled

Number of B-Frames: 3

B-Pyramid: Enabled

RDO for B-Frames: Enabled (RDO = Rate Distortion Optimization)

Weighted B-Prediction: Enabled

Bidirectional M.E.: Enabled

B-Frame Mode: Auto

Dazu wurde jeweils die gewünschte Bitrate eingestellt.

8.2.3 DivX 6.2.5

DivX ist einer der effektivsten (wahrscheinlich *der* effektivste) MPEG-4 Teil 2 ASP (Advanced Simple Profile) Codec. Im oben erwähnten H.264-Codec-Vergleich der Moskauer Staatsuniversität ([73]) wurde DivX in der Version 6.0 als Repräsentant für MPEG-4 Visual-Codecs mit einbezogen und konnte dabei teilweise ganz gut mit den H.264-Codecs mithalten, schnitt aber insgesamt schlechter als die meisten anderen ab.

Der DivX-Codec (Version 6.5.2) wurde zusammen mit VirtualDub (Version 1.6.15) mit folgenden Einstellungen verwendet:

Home Theater Profile

Einfacher Durchlauf (da kein automatischer Mehrfachdurchlauf möglich)

Kodiermethode: Atemberaubende Qualität

Bei den restlichen Einstellungen wurden die Standardwerte übernommen, die Bitraten wurden jeweils wie gewünscht eingestellt.

8.2.4 Windows Media Video 9 (WMV9)

WMV9 ist dem H.264-Standard sehr ähnlich, aber weniger komplex. In verschiedenen Qualitätstests war er H.264 nur knapp unterlegen oder sogar überlegen ([5], [39]).

Zur WMV9-Codierung in dieser Testreihe wurde eine Testversion von Sorenson Squeeze 4.3 verwendet. Dem komprimierten Video wird dadurch ein Wasserzeichen hinzugefügt, wodurch die Ergebnisse der durchgeführten Messungen verfälscht sind. Die Testresultate für WMV9 sind daher unter diesem Vorbehalt zu betrachten und nicht zu einem direkten Qualitätsvergleich mit den anderen Codecs geeignet.

8.2.5 Sorenson Video 3 Pro (SVQ3)

Sorenson Video 3 war vor der Einführung von H.264 der wichtigste QuickTime-Codec mit Apple als exklusivem Lizenznehmer. Die Nachfolgeversion Sorenson Spark (Sorenson H.263) wird in Macromedia Flash Video verwendet, und ist in dieser Form für die Kompression von Videoinhalten zur Veröffentlichung im Internet weit verbreitet.

Der in QuickTime Pro enthaltene SVQ3-Codec (Standard Edition V. 3.20.301) erzielte in Qualitätsvergleichen schlechtere Ergebnisse als der in Verbindung mit der Sorenson

Squeeze 4.3 Testversion verwendete Sorenson Video 3 Professional Edition (V. 3.20.305) - trotz des beeinträchtigenden Wasserzeichens.

Für die Testreihe wurde deshalb die SVQ3 Pro Version verwendet. Die Ergebnisse blieben ohnehin so weit hinter den anderen getesteten Codecs zurück, dass die Verfälschung durch das Wasserzeichen nicht weiter berücksichtigt werden muss. In den Tests mit den Sequenzen „Wind“ und „Zug“ wurde wegen der vorherigen schlechten Resultate auf den Sorenson-Codec ganz verzichtet.

8.3 Vorgangsweise bei den Tests

Alle Testsequenzen wurden vor der Kompression in einen Vollbildmodus gebracht, um die Videoqualität subjektiv besser beurteilen zu können. Danach wurden sie in voller zeitlicher und räumlicher Auflösung mit fünf (City, Rugby, Susi) bzw. vier (Wind, Zug) verschiedenen Codecs jeweils eingestellt auf folgende Bitraten codiert: 100, 200, 300, 450, 600, 800, 1000, 1300, 1600, 2000 und 2400 kbit/s.

Die codierten Sequenzen wurden dann mit Hilfe des MSU Video Quality Measurement Tools mit der Originalsequenz hinsichtlich der Qualitätsmaße Y-PSNR, Y-SSIM, Y-VQM und dem „MSU Blocking“-Maß verglichen. Aus den errechneten Mittelwerten wurden Vergleichsgraphen („Rate-Distortion“-Kurven) erstellt.

8.4 Testergebnisse im Einzelnen: PSNR, SSIM, VQM

8.4.1 Sequenz „City“

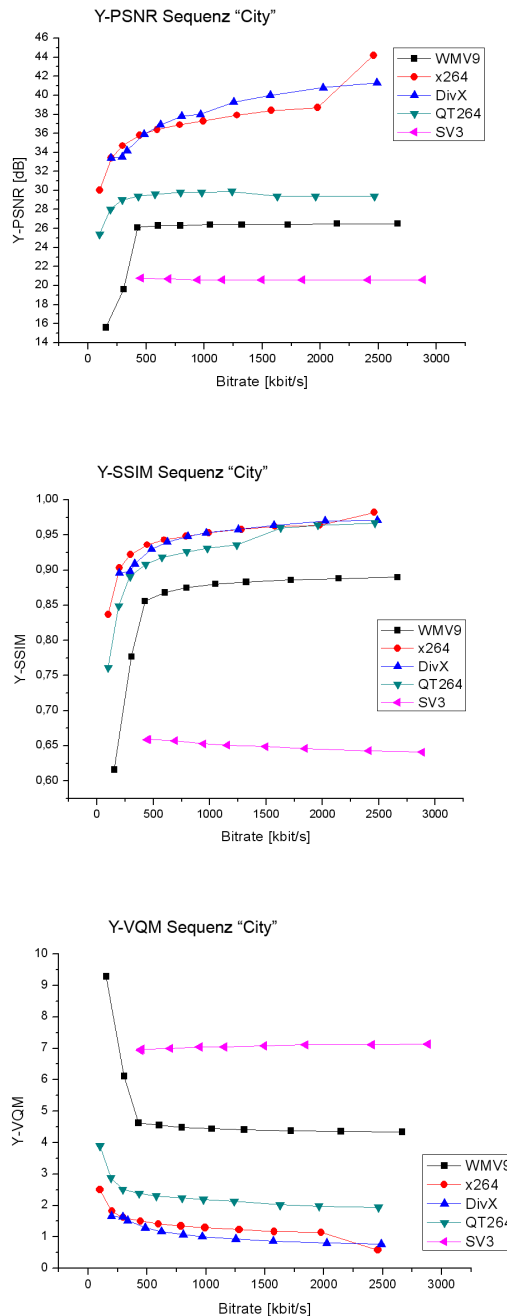


Abb. 29, 30 u. 31: Sequenz "City",
Y-PSNR, Y-SSIM, Y-VQM

Überraschenderweise schneidet DivX hier teilweise eindeutig besser ab als x264 (das war aufgrund der Testergebnisse der Moskauer Staatsuniversität [73] nicht zu erwarten).

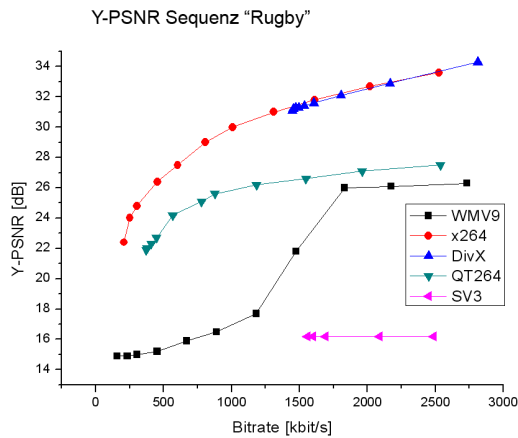
Der steile Qualitätsanstieg in der x264-Kurve zwischen 2000 und 2500 kbit/s findet sich in ähnlicher Form auch bei den Ergebnissen für die Sequenz „Susi“.

Bemerkenswert ist, dass der QuickTime H.264-Codec bei SSIM und VQM, die mit der menschlichen Wahrnehmung besser übereinstimmen als PSNR, sehr nahe an x264 und DivX herankommt – SV3 dagegen setzt sich bei SSIM noch deutlicher von den Konkurrenten ab.

Abgesehen von den im Vergleich zu den anderen Codecs ohnehin schlechten Qualitätsmaßen von SV3 war es sogar bei dieser „unkomplizierten“ Sequenz mit relativ ruhiger Bewegung nicht möglich, mit SV3 eine geringere Bitrate als etwa 450 kbit/s zu erreichen.

Bei QuickTime H.264 und WMV9 konnte ab einer Bitrate von etwa 500 kbit/s der PSNR-Wert nicht mehr gesteigert werden, SSIM und VQM aber schon – besonders deutlich SSIM bei QuickTime H.264

8.4.2 Sequenz „Rugby“



Die Sequenz „Rugby“ ist wegen der schnellen Bewegungen sehr schwierig zu komprimieren. Weder mit DivX noch mit SV3 war eine Kompression unter etwa 1500 kbit/s möglich.

Wieder kommt QuickTime H.264 in den wahrnehmungsähnlicheren Maßen näher an x264 heran als bei PSNR.

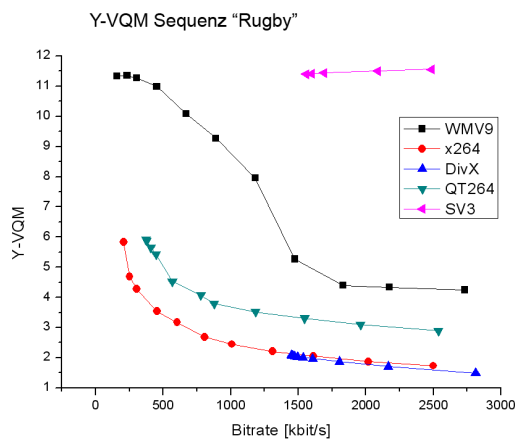
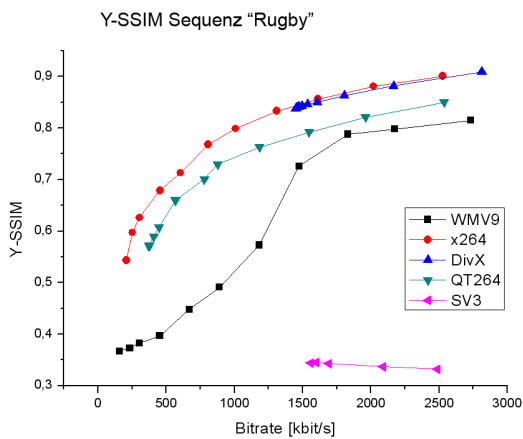
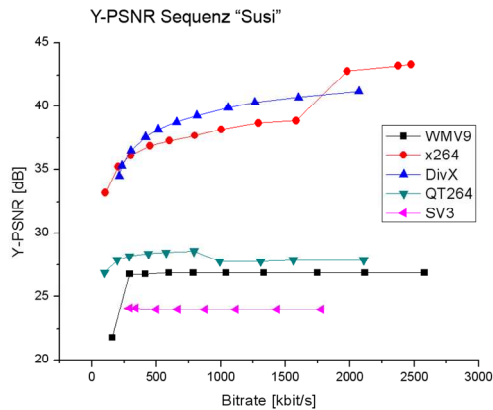


Abb. 32, 33 u. 34: Sequenz „Rugby“,
Y-PSNR, Y-SSIM, Y-VQM

8.4.3 Sequenz „Susi“



Ähnliche Verläufe wie bei der Sequenz „City“. Steiler Anstieg der PSNR-Kurve bei x264 zwischen 1500 und 2000 kbit/s.

Beim QuickTime H.264-Codec ist hier der Unterschied zwischen den PSNR- und SSIM-Werten im Vergleich zu den Konkurrenten am deutlichsten.

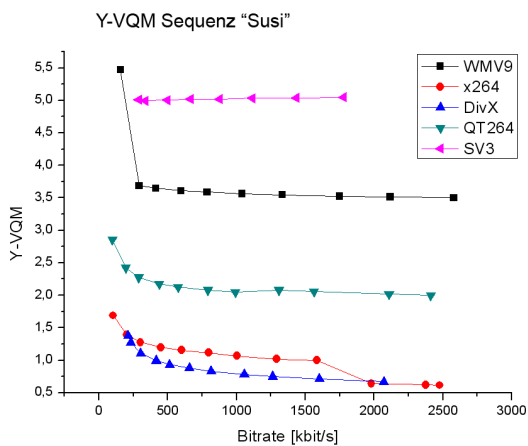
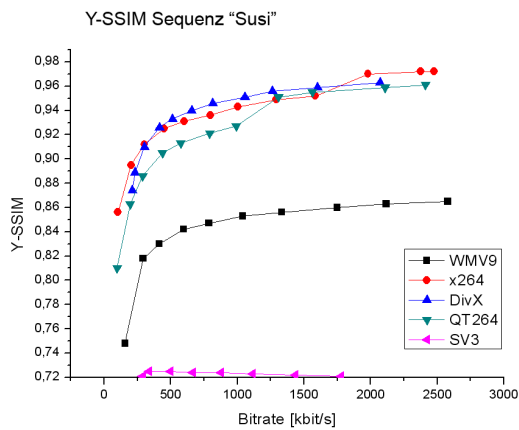
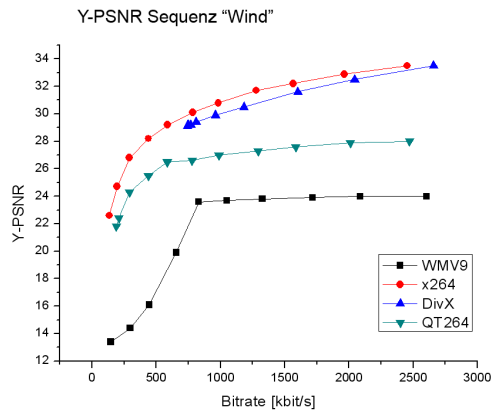


Abb. 35, 36 u. 37: Sequenz „Susi“,
Y-PSNR, Y-SSIM, Y-VQM

8.4.4 Sequenz „Wind“



Die wirbelnden Blätter in der „Wind“-Sequenz bereiten dem DivX-Codec offenbar schon solche Schwierigkeiten, dass er nicht unter eine Bitrate von etwa 600 kbit/s kommt.

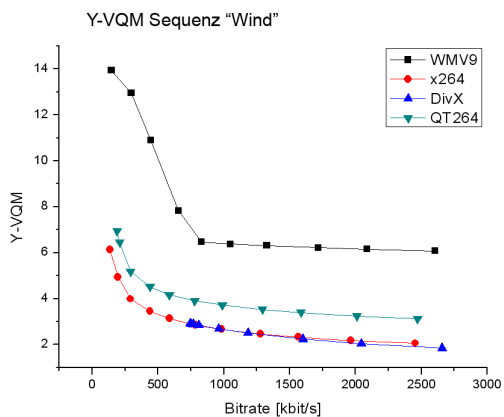
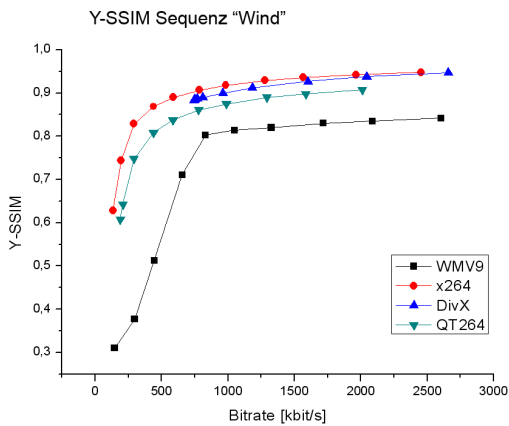
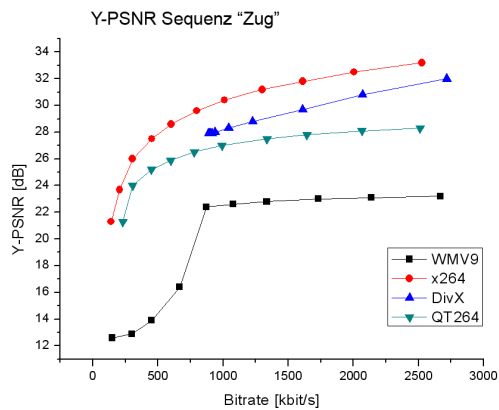


Abb. 38, 39 u. 40: Sequenz „Wind“,
Y-PSNR, Y-SSIM, Y-VQM

8.4.5 Sequenz „Zug“



Bei dieser wegen der vielen Details und feinen Linien schwierigen Sequenz lag bei DivX das untere Bitraten-Limit bei etwa 700 kbit/s.

Wie auch bei den anderen Sequenzen sind die Ergebnisse für den WMV9-Codec durch das von Sorenson Squeeze aufgebraute Wasserzeichen verfälscht und kämen normalerweise näher an die der Konkurrenten heran.

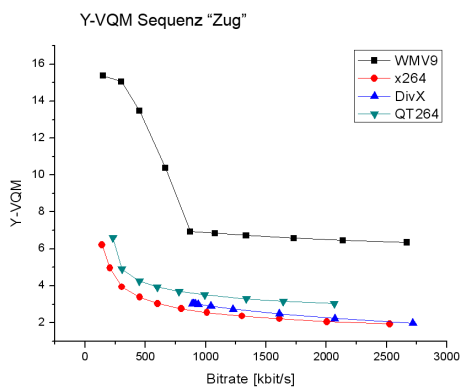
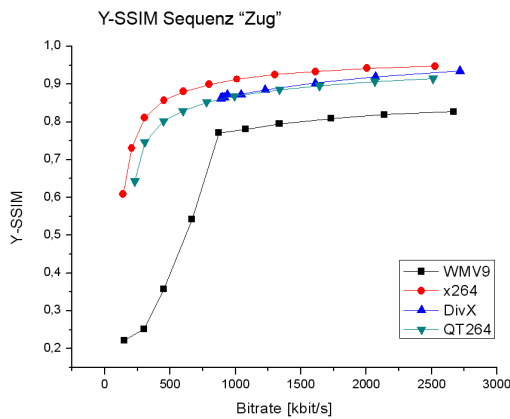


Abb. 41, 42 u. 43: Sequenz „Zug“,
Y-PSNR, Y-SSIM, Y-VQM

8.5 Testergebnisse im Einzelnen: Blocking

Der unmittelbare subjektive Qualitätseindruck war bei den DivX-codierten Sequenzen bei niedrigeren Bitraten durch die auftretenden Blockartefakte im Vergleich zu den H.264-Codecs und auch zu WMV9, wesentlich schlechter. Das liegt einerseits daran, dass bei H.264 und WMV9 zur Bewegungskompensation bei Bedarf die Makroblocks in kleinere Subblocks unterteilt werden können, aber vor allem an deren Deblocking-Filter, der in die Prädiktionsschleife eingebaut ist.

Das im Video Quality Measurement Tool der Moskauer Staatsuniversität enthaltene „Blocking“-Maß stellt einen Versuch dar, die Ausprägung dieser das subjektive Qualitätsempfinden sehr störende Blockartefakte wiederzugeben. Weder das PSNR-Maß noch die der subjektiven Wahrnehmung besser entsprechenden SSIM- und VQM-Werte können das Ausmaß der subjektiven Qualitätsminderung durch Blockartefakte korrekt abbilden.

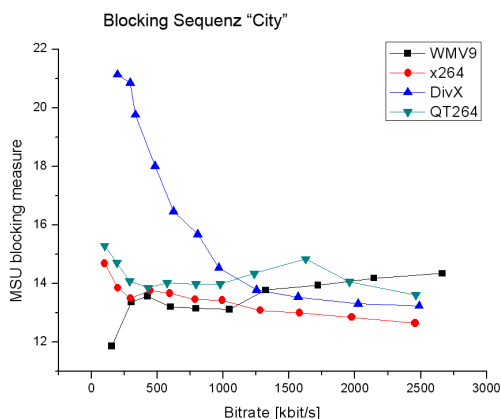


Abb. 44: Sequenz „City“, MSU Blocking

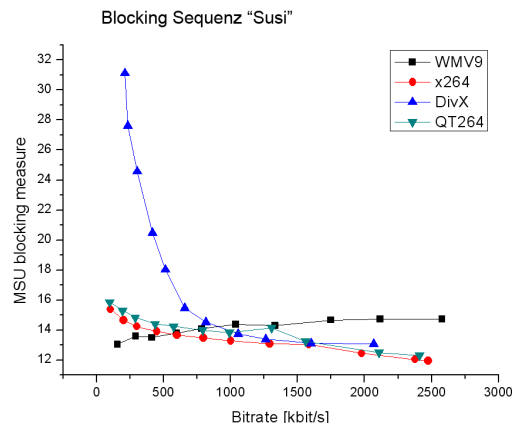


Abb. 45: Sequenz "Susi" MSU Blocking

In den beiden eher ruhigen und daher „einfach“ zu komprimierenden Sequenzen „City“ und „Susi“ zeigt das MSU Blocking-Maß eine wesentlich stärkere Blockbildung bei DivX nur im niedrigen Bitratenbereich.

Bei den schwierigeren Sequenzen mit viel Bewegung und Details ergeben sich für DivX bis hin zu sehr hohen Bitraten sehr viel schlechtere Werte als für die H.264-Codecs und WMV9.

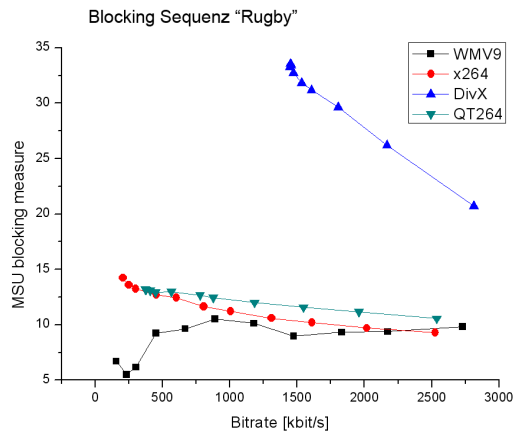


Abb. 46: Sequenz „Rugby“, MSU Blocking



Abb. 47: Detail „Rugby“, Original



Abb. 48: Detail „Rugby“, DivX , 1534 kbit/s



Abb. 49: Detail „Rugby“, x264, 807 kbit/s



Abb. 50: Detail „Rugby“, DivX, 1534 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes



Abb. 51: Detail „Rugby“, x264, 807 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes

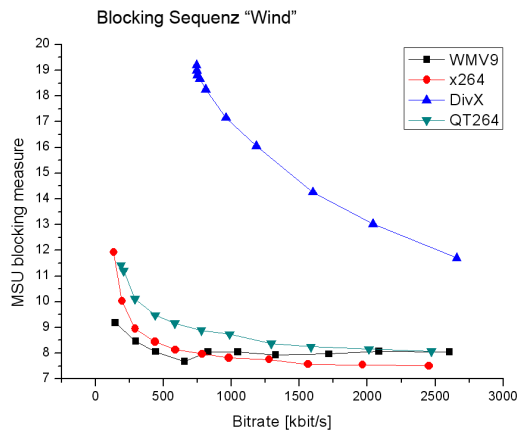


Abb. 52: Sequenz „Wind“, MSU Blocking



Abb. 53: Detail „Wind“, Original



Abb. 54: Detail „Wind“, DivX, 810 kbit/s



Abb. 55: Detail „Wind“, x264, 291 kbit/s



Abb. 56: Detail „Wind“, DivX, 810 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes



Abb. 57: Detail „Wind“, x264, 291 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes

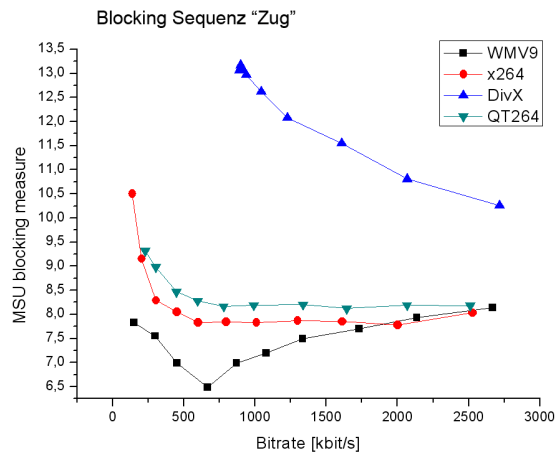


Abb. 58: Sequenz „Zug“, MSU Blocking



Abb. 59: Detail „Zug“, Original



Abb. 60: Detail „Zug“, DivX, 889 kbit/s



Abb. 61: Detail „Zug“, x264, 452 kbit/s



Abb. 62: Detail „Zug“, DivX, 889 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes



Abb. 63: Detail „Zug“, x264, 452 kbit/s
Visuelle Darstellung des MSU Blocking-Maßes

8.6 Zusammenfassung der Testergebnisse

	x264	QT H.264	DivX	WMV9	SVQ3
Susi Statische Kamera, wenig Bewegung, einheitlicher Hintergrund, starkes feinkörniges Rauschen	(PSNR: beste Werte ab 2000 kbit/s) Blocking: beste Werte		PSNR: beste Werte (bis 2000 kbit/s)		schlechteste Werte
City ruhige, gleichmäßige seitliche Kamerabewegung, leichte Auf- und Ab- Bewegung, feine Details (Fenster)	(PSNR: beste Werte ab 2500 kbit/s) Blocking: beste Werte		PSNR: beste Werte (bis 2500 kbit/s)		schlechteste Werte
Wind Kamera statisch, langsames Herauszoomen, schnelle Bewegung der wirbelnden Blätter	PSNR: beste Werte Blocking: beste Werte				nicht getestet
Zug langsame Translations- bewegung von Kamera, Zug und Kalender, viele feine Details (Schrift, Zeichnung, Tapetenmuster)	PSNR: beste Werte			Blocking: beste Werte	nicht getestet
Rugby sehr schnelle und komplexe Bewegungen der Kamera und der Spieler	PSNR: beste Werte (gemeinsam mit DivX)		PSNR: beste Werte (gemeinsam mit x264)	Blocking: beste Werte	schlechteste Werte
	beste PSNR- Werte bei schnellen Bewegungen und feinen Details beste Blocking- Werte gemeinsam mit WMV9	bei PSNR, SSIM, VQM durchgehend schlechter als DivX, besser als WMV9 Blocking: durchgehend schlechtere Werte als WMV9	beste PSNR- Werte bei langsamer/wenig Bewegung Blocking-Werte bei schnellen Bewegungen und feinen Details sehr schlecht bei schwierigen Inhalten lässt sich Bitrate nicht unter Mindestwert reduzieren	bleibt abgesehen von den guten Blocking-Werten hinter allen anderen (außer SVQ3) zurück niedrige Bitraten können nur durch Einfrieren des Bildes erreicht werden	durchgehend schlechteste Ergebnisse bei allen Sequenzen und Messwerten

Tab. 5: Übersicht Testergebnisse

Trotz der technischen Unterlegenheit und der durchwegs schlechteren PSNR-Ergebnisse kommt der QuickTime H.264-Codec in der subjektiven Qualität (repräsentiert durch die SSIM-, VQM-, und Blocking-Werte) sehr nahe an x264 heran. Wer Wert auf einfache Bedienbarkeit legt und bereit ist, die rund 30 € für QuickTime Pro zu bezahlen, ist damit durchaus gut bedient. Hinsichtlich ihrer Geschwindigkeit wurden die Codecs zwar nicht untersucht, trotzdem sollte bemerkt werden, dass der QuickTime Encoder mit großem Abstand der langsamste von allen getesteten war.

Beide getesteten H.264-Encoder verfügen (im Gegensatz zu den anderen Codecs) über eine sehr exakte Bitratenkontrolle und ermöglichen auch bei schwierigen Inhalten eine starke Datenreduktion (bei dementsprechend niedrigerer Qualität).

Abgesehen von der fehlenden Halbbild-Unterstützung ist der einzige Nachteil von x264 gegenüber dem QuickTime-Codec die vergleichsweise komplizierte Handhabung, die in erster Linie die anfängliche Installation und Konfiguration mehrerer notwendiger Komponenten betrifft. Sehr gute Anleitungen dazu gibt es auf Doom9's Forum (<http://forum.doom9.org>), das auf Videokompression spezialisiert ist ([83]). Seit kurzem ist dort auch eine Reihe vorgefertigter Einstellungsprofile für die MeGUI-Benutzeroberfläche erhältlich, die den Umgang mit x264 erleichtern. Mehrere dieser Profile sind auf eine Kompatibilität mit dem QuickTime 7 Player ausgerichtet ([85]).

Mit DivX konnten bei PSNR, SSIM und VQM vergleichbar gute und teilweise bessere Ergebnisse erzielt werden als mit x264, bei schwierigen Inhalten war allerdings eine Datenreduktion nur bis zu einer bestimmten Untergrenze möglich. Die niedrigste eingestellte Bitrate von 100 kbit/s konnte mit DivX bei keiner einzigen Testsequenz erreicht werden. Die im Vergleich zu den H.264-Codecs überraschend gute Leistung lässt sich damit erklären, dass DivX einen wesentlich längeren Entwicklungsprozess hinter sich hat. Für x264 und alle anderen H.264-Implementationen sind in den nächsten Jahren noch wesentliche Leistungssteigerungen zu erwarten.

Die Messergebnisse für Blockartefakte zeigen eine eindeutige Unterlegenheit von DivX (und damit MPEG-4 Visual) gegenüber H.264 und WMV9, die sich durch den bei MPEG-4 Visual fehlenden Deblocking-Filter und die Anwendung der DCT-Transformation auf größere Blocks ergeben. Andererseits zeigen die x264-codierten Sequenzen, wie auch in den Detail-Beispielbildern deutlich zu sehen, eine stärkere Unschärfe. Welche der beiden Verzerrungserscheinungen im Einzelnen die wahrgenommene Qualität stärker beeinträchtigt, müsste durch subjektive Qualitätstests überprüft werden, und variiert mit Sicherheit je nach Bildinhalt.

Im Gegensatz zu DivX sind daher die H.264-Codecs und WMV9 besonders gut geeignet, um beim Abspielen des codierten Videos ohne eine deutliche Qualitätsminderung die ursprüngliche Bildauflösung zu vergrößern – unter Umständen kann dabei sogar die subjektive Qualität des im selben Ausmaß vergrößerten Originals übertroffen werden.

Der WMV9-Encoder erreicht zwar (wie in den Kurven zu sehen) die gewünschten niedrigen Datenraten, jedoch nur durch Einfrieren des Bildes bei starker Bewegung, was auch die vergleichsweise extrem schlechten Qualitätsmaße im niedrigen Bitratenbereich erklärt. Wie schon gesagt, können bei den Ergebnissen für WMV9 (außer dem Blocking-Maß) wegen dem Sorenson-Wasserzeichen zwar der Verlauf der Kurve, aber nicht die absoluten Werte zu Schlussfolgerungen herangezogen werden.

Der Sorenson Video 3 Codec liegt in allen Ergebnissen – besonders deutlich bei SSIM und VQM - so weit zurück, dass er eigentlich nicht als Konkurrent für H.264 betrachtet werden kann.

10 Abschließende Zusammenfassung und Reflexion

MPEG-2 Visual ist bis heute durch seinen Einsatz im digitalen Fernsehen und bei DVDs der am weitesten verbreitete und auch finanziell erfolgreichste Videokompressions-Standard. In seinen unmittelbaren Nachfolger MPEG-4 Visual, der ein Höchstmaß an Flexibilität bietet, setzten die Entwickler große Hoffnungen, MPEG-2 durch die erweiterten Einsatzmöglichkeiten noch zu überbieten.

Verschiedene Umstände haben dieses Erfolgsszenario jedoch verhindert. Der große Umfang und die Komplexität des Standards erschwerten die Entwicklungsarbeit, die neuen unpopulären Lizenzierungsbedingungen führten zu öffentlichen Auseinandersetzungen. MPEG-2-Encoder wurden laufend verbessert und damit der Effizienzvorsprung von MPEG-4 Visual verkleinert.

Der jüngste Standard, H.264 (MPEG-4 AVC) erfüllt alle Voraussetzungen, um nicht nur MPEG-2 in seinen Einsatzgebieten nachzufolgen, sondern sich auch in anderen Bereichen durchzusetzen.

Die Frage, ob H.264 den ihm prophezeiten Siegeszug schon angetreten hat, lässt sich eindeutig mit „Ja“ beantworten. Der „große Durchbruch“ ist vielleicht noch nicht deutlich spürbar, solange HD DVD- und Blu-Ray-Player und H.264-fähige Set-Top-Boxen für den DVB-Empfang noch nicht im großen Stil zum Verkauf angeboten werden. Bei den DVD-Nachfolgern ist zwar VC-1 bzw. WMV9 ein gleichwertiger Konkurrent, aber im Broadcastbereich wird eindeutig die H.264-Codierung dominieren.

Im Bereich Videotelefonie und Videokonferenztechnik ist H.264 schon seit einiger Zeit etablierter Standard. Für mobile Videoanwendungen, die selbst noch eine relativ junge Entwicklung sind und großes Wachstumspotenzial zeigen, wird ebenfalls H.264 bevorzugt zum Einsatz kommen. Was Video-Streaming im Internet betrifft, wird H.264 vermutlich auch in näherer Zukunft nicht wie in den meisten anderen Bereichen eindeutig dominieren, aber wahrscheinlich wird er neben mehreren proprietären Codecs auch dort eine immer wichtigere Rolle spielen.

Mit der Möglichkeit zur Einbettung in Datenströme und damit Anpassung an verschiedene Übertragungswege ist H.264 bestens für alle Anwendungen gerüstet, die sich auch in Zukunft für Videodaten ergeben könnten.

Ähnlich wie die laufende Weiterentwicklung und Verbesserung von MPEG-2-Encodern zeigt, werden auch H.264-konforme Implementierungen erst in mehreren Jahren an die Grenzen ihrer erreichbaren Effizienz stoßen.

Würde man den Erfolg an der Vielfalt der Einsatzbereiche messen, wäre H.264 wahrscheinlich schon jetzt der erfolgreichste Videokompressionsstandard aller Zeiten. Misst man ihn an der Anzahl der weltweit verbreiteten standardkonformen Encoder- und Decoder-Chips und -Softwareeinheiten, kann es noch mehrere Jahre dauern, bis H.264 an MPEG-2 herankommt.

Anhang

Quellenverzeichnis

- [1] Tilo Strutz: „Bilddatenkompression: Grundlagen, Codierung, Wavelets, JPEG, MPEG, H.264“, 3. Auflage, Wiesbaden, 2005
- [2] Peter Symes: „Digital Video Compression“, New York, 2004
- [3] Iain E. G. Richardson: „H.264 and MPEG-4 video compression“, Chichester, 2003
- [4] Paola Sunna: „AVC / H.264 – an advanced video coding system for SD and HD broadcasting“, EBU Technical Review, April 2005
- [5] Klaus Diepold: „SMPTE VC-1 und MPEG-4 AVC im Vergleich“, Digital Production 04/2005
- [6] Franc Kozamernik, Paola Sunna, Emmanuel Wyckens, Dag Inge Pettersen: „Subjective quality of internet video codecs – Phase 2 evaluations using SAMVIQ“, EBU Technical Review, Jänner 2005
- [7] Karsten Sühling, Heiko Schwarz, Thomas Wiegand: „Effizienter kodieren – Details zum kommenden Videostandard H.264/AVC“, c't magazin für computertechnik, Heft 6, 2003
- [8] Gary J. Sullivan, Pankaj Topiwala, Ajay Luthra: „The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions“, August 2004
- [9] Detlev Marpe, Thomas Wiegand, Stephen Gordon: „H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas“, September 2005
- [10] Heiko Schwarz, Detlev Marpe, Thomas Wiegand: „MCTF and Scalability Extension of H.264/AVC“, Dezember 2004
- [11] Heiko Schwarz, Detlev Marpe, Thomas Wiegand: „Basics concepts for supporting spatial and SNR scalability in the Scalable H.264/MPEG4-AVC Extension“, September 2005
- [12] Ralf Schäfer, Thomas Wiegand, Heiko Schwarz: „The emerging H.264/AVC standard“, EBU Technical Review, Jänner 2003
- [13] Dr. S.R. Ely: „MPEG video coding – A simple introduction“, EBU Technical Review, Winter 1995
- [14] David Wood, „Everything you wanted to know about Video codecs – but were too afraid to ask“, EBU Technical Review, Juli 2003

- [15] Wen-Hsiao Peng, Chia-Yang Tsai, Tihao Chiang, Hsueh-Ming Hang: „Advances of MPEG Scalable Video Coding Standard“, 2005
- [16] Thomas Wiegand, Heiko Schwarz, Detlev Marpe, T. Shierl, K. Grüneberg: „Skalierbare Videocodierung (SVC): Technologie und Anwendungen“, Mai 2006
- [17] Daniel Budka: „MPEG-4 Der neue Standard in der Videotechnik“, 2002
- [18] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra: „Overview of the H.264 / AVC Video Coding Standard“, IEEE Transactions on circuits and systems for video technology, Juli 2003
- [19] EBU Technical Statement-D 96-2003 „Joint H.264/MPEG-4 AVC Patent Licence“, 2003
- [20] Tim Siglin: „H.264 Begins Its Ascent“, streamingmedia.com, Jänner 2005
<http://www.streamingmedia.com/article.asp?id=9015>
- [21] Franklin Douglas: „MPEG licensing basics“, Video/Imaging Desing Line, März 2005
<http://www.videsignline.com/showArticle.jhtml?printableArticle=true&articleId=159902209>
- [22] Dominik Schöneberg: „H.264: Der Video-codec der Zukunft vorgestellt“, netzwelt.de, Dezember 2005
<http://www.netzwelt.de/news/73129-h264-der-videocodec-der-zukunft.html>
- [23] Jim Dalrymple: „Apple: Video's future is Tiger, QuickTime 7, H.264“, Macworld, Jänner 2005
<http://www.macworld.com/news/2005/01/31/h264/index.php>
- [24] WBU-TC, „WBU-TC expresses concerns about MPEG-4 AVC licensing“, Juni 2004
http://www.abu.org.my/public/documents/WBU_TC_MPEG_0306l.pdf
- [25] Apple, „QuickTime and MPEG-4: Now Featuring H.264“, 2005
http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf
- [26] Apple, „H.264 Frequently Asked Questions“, 2006
<http://www.apple.com/quicktime/technologies/h264/faq.html>
- [27] Apple, „H.264 Stunning video quality from 3G to HD“, 2006
<http://www.apple.com/quicktime/technologies/h264/>
- [28] MPEG LA, „Summary of H.264/AVC License Terms“, 2006
http://www.mpegla.com/avc/AVC_TermsSummary.pdf
- [29] MPEG LA, „AVC Patent Portfolio License Briefing“, 2006
<http://www.mpegla.com/avc/avc-agreement.cfm>

- [30] Fraunhofer HHI image processing, Publications Detlev Marpe (Stand Juni 2006)
<http://iphome.hhi.de/marpe/pub.htm>
- [31] Fraunhofer HHI image processing, Publications Thomas Wiegand (Stand Juni 2006)
<http://iphome.hhi.de/wiegand/pubs.htm>
- [32] ITU-T Study Group 16 – Question 6/16 (Stand Juni 2006)
<http://www.itu.int/ITU-T/studygroups/com16/sg16-q6.html>
- [33] ITU-T Workshop on Video and Image Coding and Applications (VICA), Juli 2005
<http://www.itu.int/ITU-T/worksem/vica/aprogram.html>
- [34] Gary J. Sullivan: “Overview of digital video coding standards prior to H.264”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S0_P2_Sullivan.pdf
- [35] Jean-François Mulé, Mayumi Matsumoto: “Video and image compression in cable networks”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S1_P2_Mule.pdf
- [36] Stephan Wenger: „Video and image coding in mobile networks”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S1_P3_Wenger.pdf
- [37] Thomas Wiegand: “H.264/MPEG-4 AVC Application & Deployment Status”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S2_P1_Wiegand.pdf
- [38] Gary J. Sullivan, “Overview of H.264/MPEG-4 Advanced Video Coding (AVC)”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S3_P1_Sullivan.pdf
- [39] Mark Jeffrey: “The SMPTE 421M “VC-1 Standardization Project”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S3_P3_Jeffrey.zip
- [40] David Hands, “Towards a standardised perceptual quality metric for multimedia”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S5_P1_Hands-V2.pdf
- [41] Thomas Wiegand, “New techniques for improved video coding”, VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S6_P1_WiegandRev1.pdf

- [42] Jens-Rainer Ohm, "Standardization in JVT: Scalable Video Coding", VICA, Juli 2005
http://www.itu.int/ITU-T/worksem/vica/docs/presentations/S3_P2_Ohm.pdf
- [43] DrunkenBlog, "Deconstructing H.264/AVC", Juli 2004
<http://www.drunkenblog.com/drunkenblog-archives/000312.html>
- [44] Via Licensing, H.264 License Fees
<http://www.vialicensing.com/products/AVCH264VC/license.terms.html>
- [45] Chris Adamson: "Magnificent Seven: What's New for Users in QuickTime 7", O'Reilly Network, Mai 2005
<http://www.macdevcenter.com/pub/a/mac/2005/05/10/qt7.html>
- [46] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli: "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, April 2004
<http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>
- [47] Olivia Nemethova, Michal Ries, Markus Rupp, Eduard Siffel: "Quality Assessment for H.264 coded low rate and low-resolution video sequences", November 2004
http://publik.tuwien.ac.at/files/pub-et_8787.pdf
- [48] Feng Xiao: "DCT-based Video Quality Evaluation", Winter 2000
http://www-ise.stanford.edu/class/ee392j/projects/projects/xiao_report.pdf
- [49] Alan Clark: "Clarifying Video Quality Metrics", TMCnet, April 2006
<http://news.tmcnet.com/news/-voip-iptv-video-audio-ip-telephony-telchemy-inc-/2006/04/11/1561303.htm>
- [50] MPEG LA, Liste von MPEG-2 Lizenznehmern, Juli 2006
<http://www.mpegla.com/m2/m2-licensees.cfm>
- [51] MPEG LA, Liste von MPEG-2 Visual Lizenznehmern, Juli 2006
<http://www.mpegla.com/m4v/m4v-licensees.cfm>
- [52] MPEG LA, Liste von H.264/AVC Lizenznehmern, Juli 2006
<http://www.mpegla.com/avc/avc-licensees.cfm>
- [53] Fraunhofer Heinrich Hertz Institut, „ITG-Preis an HHI-Mitarbeiter vergeben für ein Verfahren zur signifikanten Beschleunigung der Videoübertragung“, 2004
http://www.hhi.fraunhofer.de/german/gf/press_archive/pr2004/pr2004-5.html
- [54] The RFC Archive, "RFC 3984 – RTP Payload Format for H.264 Video", Feber 2005
<http://www.rfc-archive.org/getrfc.php?rfc=3984>

- [55] Ed Frauenheim, "Next-generation DVD moves ahead", CNET news.com, Juni 2004
http://news.com.com/Next-generation+DVD+moves+ahead/2100-1040_3-5233211.html
- [56] Blu-Ray.com, Juli 2006
<http://www.blu-ray.com/>
- [57] Michael Knott: "Ricoh bekommt Blu-Ray-Disc und HD-DVD unter einen Hut", netzwelt.de, Juli 2006
<http://www.netzwelt.de/news/74331-ricoh-bekommt-bluraydisc-und-hddvd.html>
- [58] DVD Forum, Juli 2006
<http://www.dvdforum.org/>
- [59] Michael Knott: „High Definition: Hochauflösendes Übel, gute Besserung“, netzwelt.de, Juni 2006
<http://www.netzwelt.de/news/74272-high-definition-hochaufloesendes-uebel-gute.html>
- [60] Tandberg Annual Report 2003
http://www.tandberg.net/annual_report2003/r_and_d.htm
- [61] DVB Project, "About DVB", Juli 2006
http://www.dvb.org/about_dvb/
- [62] ETSI, "TS 101 154 adds support of MPEG-4 in DVB Compliant Transport Streams", Feber 2005
<http://www.etsi.org/pressroom/previous/2005/2005%5F02%5Fbroadcast.htm>
- [63] Mike Slocombe, Simon Perry: "World's First MPEG-4/AVC Broadcast on HomeChoice", April 2005
http://digital-lifestyles.info/display_page.asp?section=platforms&id=2122
- [64] John Borland, "Satellite TV snubbing Microsoft", CNET news.com, Jänner 2005
http://news.com.com/Satellite+TV+snubbing+Microsoft/2100-1025_3-5544324.html?tag=nefd.lede
- [65] Erste DVB-S2-Karte mit integriertem H.264-Decoder-Chip
<http://www.heise.de/newsticker/meldung/70681>
- [66] AREA DVD, "England und Frankreich testen HDTV via DVB-T", areadvd.de, Juni 2006
http://www.areadvd.de/news/2006/200606/02062006003_HDTV_DVB-T.shtml
- [67] watcha – TV goes mobile, "TV/Radio Kanäle", Juli 2006
<http://www.watcha.de/index.php?id=5>
- [68] Digiradio.ch, "DMB News", Mai 2006
<http://www.digiradio.ch/dmb/news/index.html>

- [69] heise mobil, "DVB-H: Handy-Fernsehen angetestet", Juni 2006
<http://www.heise.de/mobil/newsticker/meldung/74210>
- [70] Heise mobil, "Interesse an Handy-TV gering", Juni 2006
<http://www.heise.de/mobil/newsticker/meldung/74114>
- [71] Doom9's Forum, "List of MPEG-4 AVC/H.264 Video Codecs", April 2006
<http://forum.doom9.org/showthread.php?t=95939>
- [72] VideoLAN developers, "x264 - a free h264/avc encoder"
<http://developers.videolan.org/x264.html>
- [73] MSU (Moscow State University) Graphics and Media Lab, "Second Annual MSU MPEG-4 AVC/H.264 Video Codec Comparison", Dezember 2005
http://compression.ru/video/codec_comparison/pdf/msu_mpeg_4_avc_h264_codec_comparison_2005_eng.pdf
- [74] MSU (Moscow State University) Graphics and Media Lab, "MSU Video Quality Measurement Tool"
http://www.compression.ru/video/quality_measure/video_measurement_tool_en.html
- [75] MSU (Moscow State University) Graphics and Media Lab, "MSU Perceptual Video Quality Tool"
http://www.compression.ru/video/quality_measure/perceptual_video_quality_tool_en.html
- [76] Junko Yoshida, "H.264 decoder chips begin to hit the market", EE Times online, November 2004
<http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=54200808>
- [77] Junko Yoshida, "H.264 video coding surges into view", EE Times online, September 2003
<http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=18309407>
- [78] Panasonic Pressemeldung, „Panasonic und Sony entwickeln gemeinsam ein neues HD Digital Videokamera-Format für Aufnahmen auf DVD“, Mai 2006
http://www.panasonic.de/presse/detail/ct_detail.aspx?newsID=396d6ead-13ce-4ce4-9da3-d5e2f5c98346&refer=search&suche=&catID=16&year=&month=
- [79] Video Quality Experts Group, Downloads
<http://www.its.bldrdoc.gov/vqeg/downloads/downloads.php>
- [80] Zhou Wang: "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, April 2004
<http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>
- [81] Wikipedia, "h.264", Juli 2006
<http://en.wikipedia.org/wiki/H.264>

- [82] Doom9's Forum, "REAL Ultimate Guide to x264 and QuickTime", Juli 2006
<http://forum.doom9.org/showthread.php?t=102609>
- [83] Doom9's Forum, "x264 daily builds", August 2006
<http://mirror05.x264.nl/Sharktooth/?dir=./x264>
- [84] MeGUI – open source encoding frontend
<http://megui.sourceforge.net/>
- [85] Doom9's Forum, "MeGUI h.264/AVC custom video profiles", August 2006
<http://forum.doom9.org/showthread.php?t=101813>
- [86] MeGUI x264 custom video profiles download, August 2006
<http://mirror05.x264.nl/Sharktooth/?dir=./MeGUI/profiles>
- [87] golem.de, "MPEG-4 H.264 in Echtzeit auf PC komprimieren und dekodieren", September 2003
<http://www.golem.de/0309/27374.html>
- [88] Doom9's Forum, „List of MPEG-4 AVC / H.264 Video Codecs", August 2006-08-<http://forum.doom9.org/showthread.php?t=95939>
- [89] Real Networks, "RealVideo 10"
<http://www.realnetworks.com/products/codecs/realvideo.html>
- [90] MSU (Moscow State University) Graphics and Media Lab, "MSU Quality Measure: Information about Metrics"
http://www.compression.ru/video/quality_measure/info_en.html
- [91] Wikipedia (Mai – Aug. 2006)
http://en.wikipedia.org/wiki/Main_Page (englisch)
<http://de.wikipedia.org/wiki/Hauptseite> (deutsch)
Einträge für ARIB, UMTS, 3GPP, 3GPP2, H.264, MPEG, Videokompression, H.261, H.262, H.263, Video codec design, VC-1, ITU-T, UMD, u.v.m.

Abbildungsverzeichnis

Abb. 1	Chronologie der internationalen Videokompressionsstandards Quelle: Gary J. Sullivan, "Overview of International Video Coding Standards", VICA, Juli 2005	16
Abb. 2	Differenzbilder Quelle: Tilo Strutz, „Bilddatenkompression“, aktualisierte Auflage 2005 (S. 226)	12
Abb. 3	DCT Quelle: Iain E. G. Richardson: „H.264 and MPEG-4 video compression“, Chichester, 2003	13
Abb. 4	Zick-Zack-Abtastung Quelle: Sullivan, „Video Coding Standards prior to H.264“, VICA, Juli 2005	14
Abb. 5	Intra-4x4-Prädiktion Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	26
Abb. 6	Die acht möglichen Richtungen der Vorhersage Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	26
Abb. 7	Fünf der neun Intra-4x4-Prädiktions-Modi Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	27
Abb. 8	Makroblock-Unterteilung zur Bewegungskompensation Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	28
Abb. 9	Differenzbild (ohne Bewegungskompensation) mit Unterteilung der Blockgrößen Iain E. G. Richardson: „H.264 and MPEG-4 video compression“, Chichester, 2003	28
Abb. 10	Interpolation bei der Bewegungsvorhersage Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	29
Abb. 11	Umwandlung eines Vollbild-Makroblock-Paars (frame mode) in ein Halbbild-Makroblock-Paar (field mode) Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	32

Abb. 12	Pixelwerte an den Blockgrenzen	37
Abb. 13	Leistung des Deblocking-Filters: linkes Bild ohne Filter, rechtes mit Filter Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	37
Abb. 14	Unterteilung eines Bildes in Slices (ohne FMO) Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	38
Abb. 15	Unterteilung eines Bildes in Slices mit FMO Quelle: Wiegand, Sullivan, Bjontegaard, Luthra: "Overview of the H.264 / AVC Video Coding Standard", 2003	38
Abb. 16	Hierarchische B-Bild-Struktur mit 4 zeitlichen Auflösungen Quelle: Schwarz, Marpe, Wiegand, "Analysis of hierarchical B pictures and MCTF", 2006, S. 1	47
Abb. 17	Die Evolution von Systemen mit offenen Videokompressionsstandards Quelle: David Wood, "Everything you wanted to know about Video codecs", 2003	49
Abb. 18	Beispiel einer Rate-Distortion-Kurve Quelle: Gary J. Sullivan, „The H.264/MPEG-4 Advanced Video Coding (AVC) Standard“, VICA, Juli 2005	54
Abb. 19	MSU Video Quality Measurement Tool	66
Abb. 20	Y-PSNR-Vergleich von zwei verschieden codierten Videosequenzen im MSU Video Quality Measurement Tool	67
Abb. 21	Frame 197 aus der Sequenz „City“, Original Quelle: Video Coding Experts Group	68
Abb. 22	Frame 197 aus der Sequenz „City“, komprimiert mit DivX 6.2.5 bei 197 kbit/s	68
Abb. 23	Frame 197 aus der Sequenz „City“, komprimiert mit DivX 6.2.5 bei 197 kbit/s, Y-PSNR-Visualisierung mittels MSU Video Quality Measurement Tool	68
Abb. 24	Testsequenz „City“ Quelle: Video Coding Experts Group	69
Abb. 25	Testsequenz „Rugby“ Quelle: Video Coding Experts Group	69
Abb. 26	Testsequenz „Susi“ Quelle: Video Coding Experts Group	70

Abb. 27	Testsequenz „Wind“ Quelle: Video Coding Experts Group	70
Abb. 28	Testsequenz „Zug“ Quelle: Video Coding Experts Group	70
Abb. 29	Y-PSNR Sequenz „City“	75
Abb. 30	Y-SSIM Sequenz „City“	75
Abb. 31	Y-VQM Sequenz „City“	75
Abb. 32	Y-PSNR Sequenz „Rugby“	76
Abb. 33	Y-SSIM Sequenz „Rugby“	76
Abb. 34	Y-VQM Sequenz „Rugby“	76
Abb. 35	Y-PSNR Sequenz „Susi“	77
Abb. 36	Y-SSIM Sequenz „Susi“	77
Abb. 37	Y-VQM Sequenz „Susi“	77
Abb. 38	Y-PSNR Sequenz „Wind“	78
Abb. 39	Y-SSIM Sequenz „Wind“	78
Abb. 40	Y-VQM Sequenz „Wind“	78
Abb. 41	Y-PSNR Sequenz „Zug“	79
Abb. 42	Y-SSIM Sequenz „Zug“	79
Abb. 43	Y-VQM Sequenz „Zug“	79
Abb. 44	Sequenz „City“, MSU Blocking	80
Abb. 45	Sequenz „Susi“, MSU Blocking	80
Abb. 46	Sequenz „Rugby“, MSU Blocking	81
Abb. 47	Detail „Rugby“, Original	81
Abb. 48	Detail „Rugby“, DivX, 1534 kbit/s	81
Abb. 49	Detail „Rugby“, x264, 807 kbit/s	81
Abb. 50	Abb. 50: Detail „Rugby“, DivX, 1534 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	81
Abb. 51	Detail „Rugby“, x264, 807 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	81
Abb. 52	Sequenz „Wind“, MSU Blocking	82
Abb. 53	Detail „Wind“, Original	82
Abb. 54	Detail „Wind“, DivX, 810 kbit/s	82
Abb. 55	Detail „Wind“, x264, 291 kbit/s	82
Abb. 56	Detail „Wind“, DivX, 810 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	82
Abb. 57	Detail „Wind“, x264, 291 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	82
Abb. 58	Sequenz „Zug“, MSU Blocking	83

Abb. 59	Detail „Zug“, Original	83
Abb. 60	Detail „Zug“, DivX, 889 kbit/s	83
Abb. 61	Detail „Zug“, x264, 452 kbit/s	83
Abb. 62	Detail „Zug“, DivX, 889 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	83
Abb. 63	Detail „Zug“, x264, 452 kbit/s, Visuelle Darstellung des MSU Blocking-Maßes	83
Tab. 1	Übersicht Videokompressions-Standards	22
Tab. 2	Profile bei H.264 Quelle: Gary J. Sullivan, Pankaj Topiwala, Ajay Luthra, „The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions“, 2004	42
Tab. 3	Levels bei H.264 Quelle: Gary J. Sullivan, Pankaj Topiwala, Ajay Luthra, „The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions“, 2004	43
Tab. 4	FRExt Profile Quelle: Gary J. Sullivan, Pankaj Topiwala, Ajay Luthra, „The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions“, 2004	45
Tab. 5	Übersicht Testergebnisse	84