

H.264/AVC Advanced Video Coding

Erwin*

22. Oktober 2007

Der H.264/AVC-Standard bietet gegenüber MPEG-2 eine Reduktion der Datenrate um den Faktor 2. Erreicht wird dieser Kompressionsgewinn durch die Optimierung im Detail. Jeder einzelne Schritt der Videokodierung wurde durch diverse gute Ideen effektiver aber auch komplexer, was unvermeidbar auch die benötigte Rechenleistung herausfordert. Es stehen deutlich mehr Möglichkeiten zur Verfügung. Das beginnt bei der Unterteilung der Makroblöcke, geht über neue Prädiktions-, Transformations- und Kodierverfahren und bietet zu guter Letzt noch eine Nachfilterung zur subjektiven Verbesserung der Bildqualität an. Ihren Nutzen aus diesem Standard ziehen nicht nur Fernsehsender oder die Anbieter von HD-DVDs. Auch für Videokonferenzen ist es geeignet. Aufgrund dieser großen Nachfrage von einem mächtigen Markt bleiben natürlich Konkurrenten nicht fern, vor denen sich der AVC-Standard aber nicht zu verstecken braucht.

1 Einleitung

Obwohl MPEG-4 bereits hervorragende Möglichkeiten zur Kodierung von Videosignalen bietet, bestand dennoch der Bedarf, die Effizienz zu erhöhen und die Datenrate weiter zu verringern. Profitieren können davon vor allem Fernsehsender, die gegenüber der herkömmlichen MPEG-2-Ausstrahlung bei DVB-T einiges an Bandbreite sparen können und somit vor allem im vom Frequenzen dicht besiedelten Europa gleichzeitig mehr Kanäle zur Verfügung haben. Weiterhin werden brauchbare Videoübertragungen für Systeme mit vergleichsweise niedrigen Datenraten, wie z.B. UMTS oder xDSL, überhaupt erst ermöglicht. Ein großes Interesse an einer verbesserten Datenreduktion besteht aber auch im Hinblick auf Speicherung und Archivierung von Videosignalen. Hier sei an die HD-DVD gedacht, die ohne effizientere Kodierung als die herkömmliche DVD ein Fünffaches an Speicherplatz aufweisen müsste.

Um diese Bedürfnisse zu befriedigen, startete die Video Coding Experts Group (VCEG) 1997 ein Projekt namens H.26L. Von Dezember 2001 an wurde es gemeinsam mit der

*Name aus redaktionellen Gründen geändert

ISO/IEC Moving Pictures Experts Group (MPEG) weiterentwickelt und mittlerweile unter dem Namen H.264/AVC fertig gestellt und als internationaler Standard definiert. Die ISO-Bezeichnung lautet ISO/IEC 14496 10 Advanced Video Coding (AVC). Und auch hinter dem Namen MPEG-4 Part 10 steckt H.264/AVC. Wie im Folgenden deutlich wird, kommen hierbei keine grundlegend neuen Verfahren zum Einsatz. Es sind viele kleine innovative Ideen und zum Teil auch das Inkaufnehmen eines höheren Speicherbedarfs in Encoder und Decoder, aus der immerhin eine Halbierung der Datenrate resultiert.[1]

2 Technik

Im Folgenden soll nun auf die eigentlichen technischen Neuheiten von H.264/AVC gegenüber MPEG-4 eingegangen werden. Grundsätzliche Verfahren der Bild- und Videokodierung, wie sie bei JPEG und MPEG zur Anwendung kommen, werden hier als bekannt vorausgesetzt und sind in [MPEG-4 Video im Vergleich zu MPEG-2] ausführlich beschrieben. Neben den eigentlichen Videodaten, die im Video Coding Layer (VCL) verarbeitet werden, bietet der AVC-Standard den Network Abstraction Layer (NAL), auf den hier nur kurz eingegangen werden soll.

2.1 NAL

Der Network Abstraction Layer (NAL) kümmert sich um die Formatierung der Videodaten, um deren Übertragung oder Speicherung an das gewünschte Medium anzupassen. Hierbei werden den reinen Videodaten bei Bedarf auch Zusatzinformationen angefügt. Dazu organisiert der NAL die Daten in so genannten NAL-Units, welche sowohl für paketorientierte als auch byteorientierte Übertragungsverfahren geeignet sind. Im zweiten Fall werden zur Unterscheidung lediglich Startcodes vor die einzelnen NAL-Units gesetzt. Die genaue Arbeitsweise des NAL wird ausführlich in [Übertragung von MPEG-4 Inhalten] erläutert.[1]

2.2 VCL

Der Video Coding Layer (VCL) verarbeitet die Videodaten und beinhaltet die wesentlichen Verbesserungen der Videokodierung. Es kommen dabei keine grundsätzlich neuen Erkenntnisse der Kodierung von Bildmaterial zur Anwendung. Grundlage ist, wie bei allen MPEG-Verfahren, das Prinzip der Redundanz- und Irrelevanzreduktion. Sämtliche Schritte wie z.B. Prädiktion, Bewegungskompensation, Quantisierung und Entropiekodierung wurden prinzipiell übernommen, jedoch in ihren Details so ausgefeilt, dass sich ein deutlicher Gewinn bemerkbar macht. Diese Verbesserungen sollen im Folgenden anschaulich erläutert werden. Es sei noch darauf hingewiesen, dass auch der H.264/AVC-Standard, wie alle vorhergehenden Standards von MPEG und VCEG, lediglich die Syntax und Semantik des Dekodierers und des Bitstroms festlegt. Unter Einhaltung dieser Voraussetzung ist es den Entwicklern der Encoder überlassen, wie sie diesen realisieren.

2.2.1 Unterteilung in Makroblöcke

H.264/AVC unterteilt jedes Bild zunächst wie gewohnt in gleichgroße quadratische Makroblöcke. Für das Helligkeitssignal (Luminanz) sind diese 16x16 Pixel groß, für die beiden Farbdifferenzsignale (Chrominanz) jeweils 8x8 Pixel. Nun werden diese Blöcke in Gruppen, so genannten Slices, zusammengefasst. Ein Slice beinhaltet also einen bestimmten Bildbereich, der beispielsweise einen rechteckigen Bildausschnitt, ein Schachbrettmuster oder auch eine völlig willkürliche Anordnung aufweisen kann. Einige Beispiele zeigt Abbildung 1. Es ist sogar möglich, dass sich ein Makroblock in mehreren Slices befindet. Das Verfahren zur Anordnung der Makroblöcke in Slices nennt sich Flexible Macroblock Ordering (FMO).

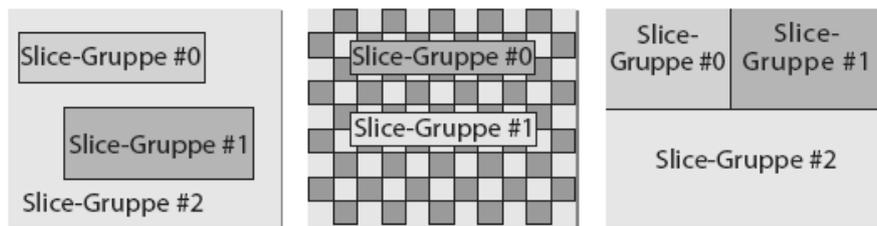


Abbildung 1: Beispiele für verschiedene Slice-Muster[1]

Da einzelne Slices komplett in einer NAL-Unit zusammengefasst und übertragen werden, kann dadurch z.B. bei dem Verlust eines einzelnen IP-Paketes eine sehr brauchbare Fehlerverdeckung durch benachbarte Slices erzielt werden. Abgesehen von der frei wählbaren Form eines Slices, die für die Kodierung eines einzelnen Bildes Vorteile bringt, definiert H.264/AVC fünf verschiedene Slice-Typen, die für die Bewegungskompensation interessant sind. Dies sind zunächst die drei Typen I, P und B, wobei I für Intra-Frame-prädiziert steht, P für prädiziert und B für Bi-prädiziert. Sie entsprechen denen einer Group of Pictures (GOP) wie sie bereits bei MPEG-2 für ganze Bilder verwendet wird. Zusätzlich gibt es noch die Typen SP als Kurzform für Switching P und SI für Switching I. Setzt sich ein Bitstrom aus mit verschiedenen Datenraten kodierten Videosignalen zusammen, so ermöglichen SP und SI das erfolgreiche Umschalten im Decoder.[1]

2.2.2 Intra-Frame-Prädiktion

Bei der Intra-Frame-Prädiktion werden aus bekannten Bildinhalten, den Makroblöcken, Vorhersagen über benachbarte Makroblöcke getroffen. Es handelt sich also um eine rein örtliche Prädiktion, deren Ziel es ist, den Unterschied zwischen vorhergesagtem und tatsächlichem Makroblock zu minimieren. Schließlich wird nur dieser Rest, der Prädiktionsfehler, weiterverarbeitet. Das Besondere bei H.264/AVC ist, dass hier verschiedene Prädiktoren zur Verfügung stehen, unter denen es den effektivsten auszuwählen gilt. Die größte Anzahl an Prädiktoren wird für die 4x4 Pixel großen Subblocks bereit gestellt, die aus der nochmaligen Unterteilung der 16x16 Pixel großen Luminanzblöcke entstehen. Hier gibt es neben einem Prädiktor, der allen Bildpunkten einfach den gleichen

Wert, nämlich den Mittelwert, zuordnet, noch acht Richtungsprädiktoren, die in Abbildung 2 anschaulich aufgeführt sind. Auch die Nummerierung der einzelnen Modi ist nicht willkürlich und für eine effektive Übertragung von Bedeutung.

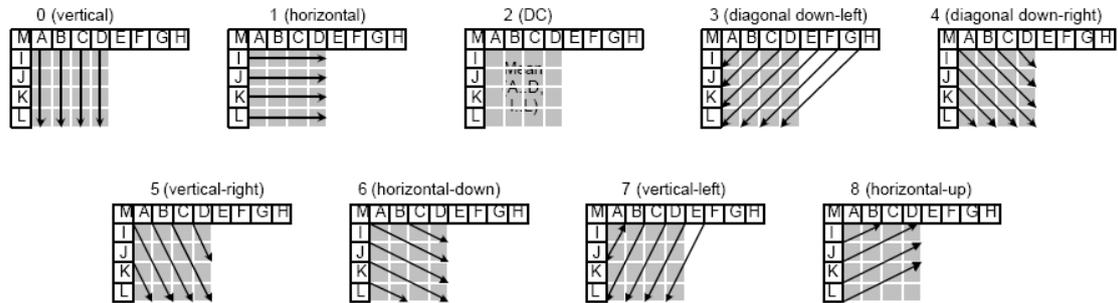


Abbildung 2: Die verschiedenen Prädiktions-Modi

Um den optimalen Prädiktor zu ermitteln, wird zunächst jeder einmal auf den zu prädizierenden (Sub-)Block angewendet. Anhand der jeweiligen Prädiktionsfehler kann für jeden Fall eine Sum of Absolute Errors (SAE) errechnet werden. Der beste Prädiktor ist der mit der kleinsten SAE. Da dieses Verfahren auf jeden einzelnen Makro- oder sogar Subblock angewendet werden muss, scheint eine Datenreduktion auf den ersten Blick als nicht besonders vielversprechend, da für jeden Block auch der verwendete Prädiktor übertragen werden muss. Doch hier wurde sich die Eigenschaft der hohen Korrelation zwischen benachbarten Bildinhalten zu Nutze gemacht. Zwei nebeneinander liegende Makroblöcke haben also mit großer Wahrscheinlichkeit denselben optimalen Prädiktor, wovon der Decoder zunächst auch einfach erst einmal ausgeht. Voraussetzung ist jedoch stets, dass sich die Makroblöcke in demselben Slice befinden. [evtl. detailliertere Beschreibung der Prädiktorzuweisung]. Für homogene Flächen ist es ausreichend und somit effektiver, die Makroblöcke nicht weiter zu unterteilen. Allerdings stehen für einen 16x16 Pixel Luminanzblock nur noch vier Prädiktionsmodi zur Verfügung. Dies sind eine vertikale, eine horizontale und eine Art diagonale Prädiktion sowie der Mittelwert aus den oberhalb und links vom Block liegenden Bildpunkten. Ebenfalls vier Modi stehen für die Prädiktion der Chrominanzsignale zur Verfügung. Es sind im wesentlichen die Gleichen wie die der Luminanz, jedoch unterscheidet sich die Reihenfolge geringfügig. Beide Chrominanzsignale verwenden stets denselben Prädiktionsmodus.[2]

2.2.3 Bewegungskompensation

Bei der Inter-Frame-Kodierung werden sich die hohen Ähnlichkeiten von zeitlich aufeinander folgenden Bildern zu Nutze gemacht. Betroffen sind hiervon natürlich nur die P- und B-Slices. Es wird weiterhin mit der Unterteilung des Bildes in Makroblöcke gearbeitet. Jeder 16x16 Pixel große Luminanzblock kann je nach Bedarf in verschiedene MxN-Blöcke aufgeteilt werden, wobei diese nicht unbedingt quadratisch sein müssen. Wird ein Makroblock in vier 8x8-Blöcke zerlegt, so kann jeder dieser Blöcke noch einmal

weiter unterteilt werden. Anschaulich wird dies in Abbildung 3 gezeigt.

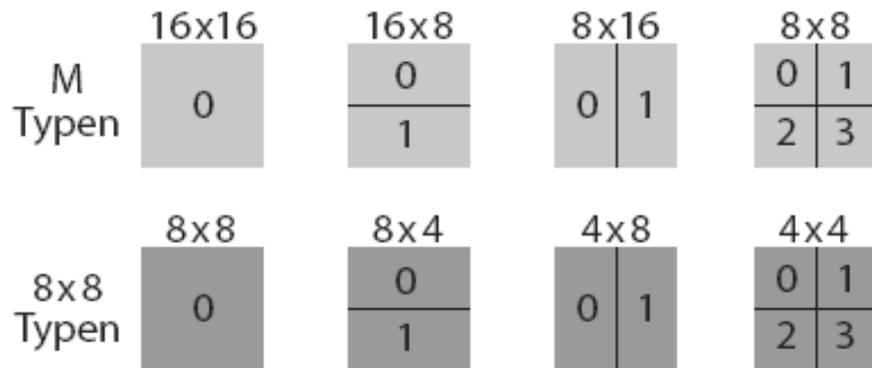


Abbildung 3: Blocktypen für die Bewegungskompensation

Sinnvoll ist diese feine Unterteilung jedoch nur, wenn der entsprechende Bildausschnitt besonders detailreich ist, da für jeden Block eigene Bewegungsvektoren ermittelt und übertragen werden müssen. Es liegt also in der Hand des Encoders eine optimale Zerlegung der Makroblöcke vorzunehmen. Abbildung 4 zeigt eine solche Zerlegung anhand eines Beispiels.

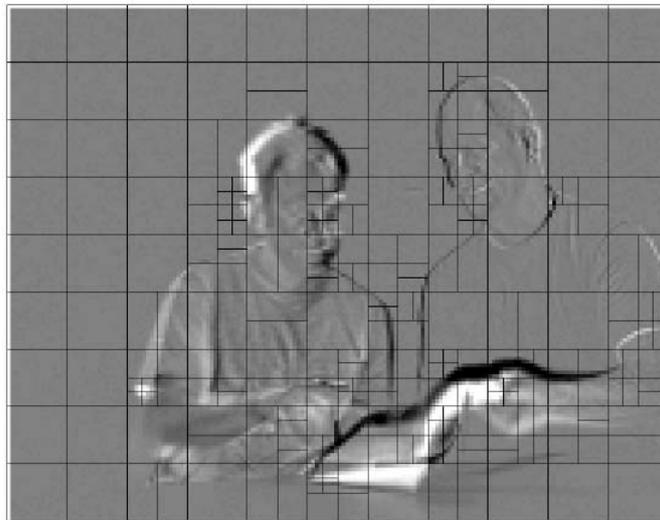


Abbildung 4: Restfehlersignal und optimale Unterteilung der Makroblöcke

Übertragen wird wie gewohnt ein Referenzbild und dessen Verschiebung. Allerdings kann beim H.264/AVC-Standard zwischen mehreren Referenzbildern ausgewählt werden um die Bewegungsschätzung zu verbessern. Für diese so genannte bewegungskompensierte Langzeitprädiktion ist allerdings ein größerer Speicher in Kodierer und Dekodierer

erforderlich, welcher bei den heutigen Speicherpreisen aber kein Problem darstellt. Eine weitere Neuheit ist durch den Skip-Kodiertyp gegeben. Sollte sich ein Makroblock in zwei aufeinander folgenden Bildern gar nicht bewegen oder verändern, so wird dieser Block lediglich aus dem ersten Bild übernommen. Bisher war es nur möglich, P-Bilder aus I-Bildern sowie B-Bilder aus I- oder P-Bildern zu schätzen. Der AVC-Standard erlaubt jedoch auch, die Bewegungskompensation aus B-Bildern, oder genauer gesagt B-Slices, durchzuführen. Doch damit noch nicht genug. Ein weiteres Manko waren bisher Bewegungen, die kein ganzzahliges Vielfaches an Bildpunkten aufweisen konnten. Dazu zählen auch besonders langsame Bewegungen. Um diesem Problem entgegenzuwirken interpoliert H.264/AVC die Werte zwischen den Bildpunkten und überträgt sie mit einer Genauigkeit von $1/4$ (Quarter Sample). Zu guter Letzt sind sogar Bewegungen möglich, die Bereiche außerhalb des Bildes mit einbeziehen. [3]

2.2.4 Transformation, Skalierung, Quantisierung

Wie bei allen Schätzungen üblich, treten auch bei der Intra- und der Inter-Frame-Prädiktion Fehler auf. Ziel der Prädiktionen war es, diesen Restfehler zu minimieren. Zu vollständigen Wiederherstellung des Originalbildes muss dieser Fehler dem Dekoder mitgeteilt werden. Dabei wird mittels geeigneter Transformation, Skalierung und Quantisierung dessen Datenrate reduziert.

Abbildung 5 zeigt, in welcher Reihenfolge ein Makroblock zerlegt und übertragen wird. Zunächst wird ein 4x4-Block (-1) mit den Gleichanteilen der 4x4-Blöcke des intrakodierten Luminanz-Makroblockes übertragen. Darauf folgen diese Blöcke (0-15) in der dargestellten Reihenfolge, wobei ihre Gleichanteilkoeffizienten zu Null gesetzt wurden. Anschließend werden entsprechend zur Luminanz die 2x2 Gleichanteils-Matrizen (16-17) der beiden Chrominanzsignale übertragen, welchen sich die 4x4-Chrominanzblöcke (18-25) mit ebenfalls zu Null gesetzten Gleichspannungs-Koeffizienten anschließen. Die beiden verschiedenen Arten von Gleichspannungs-Matrizen und die restlichen 4x4-Blöcke werden bei der anschließenden Transformation einzeln berücksichtigt.

Im Folgenden soll lediglich auf die Transformation für die 4x4-Restfehlerblöcke (0-15, 18-25) eingegangen werden. Das ist auch schon der erste Unterschied gegenüber der sonst üblichen Transformation von 8x8-Blöcken. Ebenfalls neu gegenüber allen vorherigen MPEG-Versionen ist, dass keine Cosinus-Transformation mehr verwendet wird. Um Berechnungsungenauigkeiten vorzubeugen wird stattdessen eine ihr durchaus verwandte Integertransformation benutzt. Diese ist wie die Cosinus-Transformation separierbar, d.h. für jede Dimension einzeln berechenbar, ihre Inverse lässt sich jedoch exakt durch die Verwendung von 16 Bit-Integeroperationen bestimmen. Die reine Transformation ist multiplikationsfrei und kommt allein mit Additionen und Shift-Operationen aus, was eine Berechnung auf kostengünstigen DSPs ermöglicht. Die inverse Transformation ist in dem AVC-Standard vollständig definiert, so dass es keine Probleme zwischen Kodierer und Dekodierer geben sollte.

Nachdem die Blöcke transformiert wurden, können sie bereits quantisiert werden. An diesem Punkt lässt sich auch die Datenrate kontrollieren, da die Skalierung bei H.264/AVC in die Quantisierung integriert ist. Dieser Sachverhalt und der Anspruch,

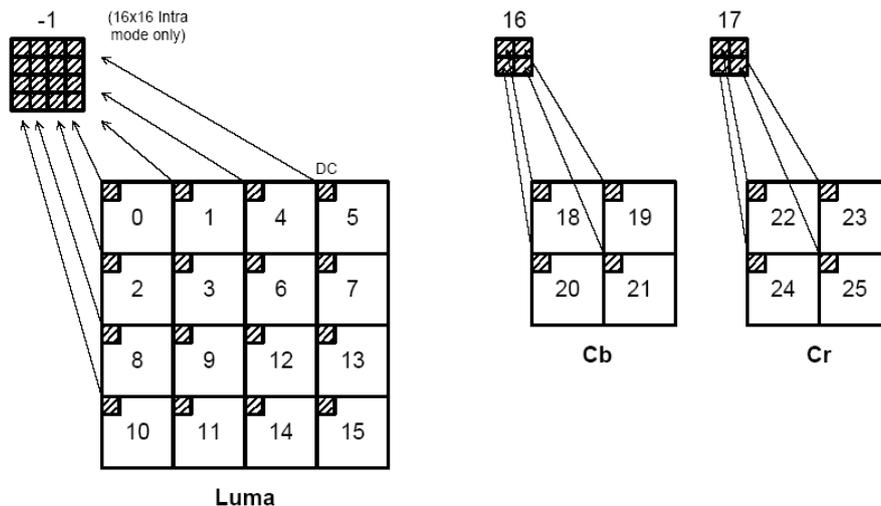


Abbildung 5: Bearbeitungsreihenfolge innerhalb eines Restfehler-Makroblockes

auch hier mit ganzzahligen Berechnungen auszukommen, gestaltet den Quantisierungsvorgang recht komplex. Zwar handelt es sich um eine lineare Quantisierungsskala, diese ist jedoch in 52 verschiedenen Quantisierungsstufen verfügbar. Die jeweils einem Makroblock zugeordnete Stufe Qstep wird durch den Parameter QP festgelegt. Das Interessante dabei ist, dass eine Erhöhung des Parameters QP um eins eine Bitraten-Reduktion von 12,5 Prozent bewirkt. In Tabelle 1 ist diese Zuordnung auszugsweise dargestellt.[4]

QP	0	1	2	3	4	5	6	7	8	9	10	11	12
Qstep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5
QP	18	...	24	...	30	...	36	...	42	...	48	...	51
Qstep	5		10		20		40		80		160		224

Tabelle 1: Quantisierungsstufen bei verschiedenen QPs

2.2.5 Entropiekodierung

Um die zu übertragenden Daten nochmals zu komprimieren werden diese kodiert. Im Detail betrifft dies z.B. den Restfehler aus den Prädiktionen, die Differenz zu den geschätzten Bewegungsvektoren, den Unterschied gegenüber dem vorigen Quantisierungsparameter QP, Informationen über die verwendeten Referenzbilder sowie Angaben zu den verwendeten Prädiktionsmodi. Zur Kodierung stehen bei H.264/AVC zwei verschiedene Verfahren zur Verfügung. Es handelt sich dabei um das Variable Length Coding (VLC) und das so genannte Context-based Adaptive Binary Arithmetic Coding (CABAC). Diese werden in den folgenden Abschnitten erläutert.

2.2.6 Variable Length Coding

Beim Variable Length Coding unterscheidet der H.264/AVC-Standard zwischen zwei verschiedenen Codes. Es handelt sich dabei um einen so genannten Exp-Golomb Code und einen Context-Adaptive Variable Length Code (CAVLC) Zunächst soll hier aber der Exp-Golomb Code erläutert werden.

Dem Exponential Golomb Code liegt eine bestimmte Erzeugungsregel zu Grunde. Diese lautet in Kurzform [M Nullen][1][INFO], wobei das erste Codewort nur aus der einzelnen Eins besteht. Die hinten angefügten INFO-Bits zählen im Prinzip einfach nur binär hoch, wenn man sich die Eins als MSB vorstellt, und vorne wird jeweils eine gleiche Anzahl an Nullen vorangestellt. Tabelle 2 zeigt die nach dieser Regel erzeugten ersten neun Codewörter. Der Index codenum ist den jeweiligen Codewörtern fest zugeordnet.

codenum	Codewort
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001

Tabelle 2: Die ersten Exp-Golomb Codewörter

Der Exp-Golomb Code wird zur Kodierung der vielen Parameter verwendet, die bei den voran gegangenen Methoden angefallen sind. Dies sind z.B. die Differenz der Bewegungsvektoren und des Quantisierungsparameters sowie die Typen der verwendeten Makroblöcke und die Prädiktionsmodi. Für die verschiedenen zu kodierenden Parameter gibt es sinnvoller Weise drei unterschiedliche Zuweisungsmethoden. Zunächst gibt es eine einfache vorzeichenlose Zuweisung, bei der der Parameter direkt dem Index codenum zugewiesen wird. Da diese Art der Zuordnung für Differenzen ungeeignet ist, steht für solche Daten die vorzeichenbehaftete Zuweisung zur Verfügung. Dabei wird alternierend hochgezählt, also 0, 1, -1, 2, -2, 3, usw.. Dadurch wird gewährleistet, dass statistisch häufiger auftretende Daten, also kleinere Differenzen, mit kürzeren Codewörtern kodiert werden. Schließlich ist das das Grundprinzip eines Variable Length Code. Die dritte Möglichkeit der Parameterzuordnung nutzt eine Tabelle, bei der, natürlich auch wieder nach Häufigkeit sortiert, bestimmte Daten direkt einer codenum zugewiesen werden.

Um das zickzack abgetastete Restfehlersignal zu kodieren, wird ein Context-adaptive Variable Length Code verwendet. Dieser macht sich diverse Eigenschaften der vorhandenen Daten zu Nutze. Das Restfehlersignal weißt üblicherweise sehr lange Folgen von Nullen auf. Hier genügt es also, lediglich die Anzahl der Nullen zu übertragen. Weiterhin bilden die Koeffizienten, die nicht Null sind, meist Folgen von +1 und -1. Es besteht auch

eine Korrelation zwischen den benachbarten 4x4-Blöcken, außerdem neigen die Koeffizienten für niedrigere Frequenzen dazu, größer zu sein, sofern sie vorhanden sind. Dieser Eigenschaft begegnet der CAVLC mit einer Auswahl von sieben verschiedenen Look-Up Tables. Je nach Betrag der Koeffizienten wird anhand festgelegter Schwellen die optimale Tabelle für eine Variable Length Kodierung ausgewählt. [5]

2.2.7 Context-based Adaptive Binary Arithmetic Coding

CABAC weist gute Komprimierungseigenschaften auf, da es verschiedene Verfahren zur optimalen Codierung verwendet, die sich auch in dem langen Namen wiederfinden lassen. Unter Berücksichtigung des Zusammenhanges der zu kodierenden Daten wird ein günstiges Wahrscheinlichkeitsmodell ausgewählt, wobei sich die Schätzer noch einmal den aktuellen Statistiken anpassen. Dem Ganzen folgt eine arithmetische Codierung.

Da es sich, wie dem Namen zu entnehmen ist, um einen binären Code handelt, müssen nichtbinäre Daten zunächst umgewandelt werden. Dies geschieht im Prinzip wie bei der Zuweisung der Codewörter des Variable Length Code. Die eigentliche Kodierung folgt aber erst später.

Es folgt die Wahl eines Wahrscheinlichkeitsmodells. Dabei gilt es ein Modell zu finden, welches möglichst gut das Auftreten von Nullen und Einsen voraussagt. Anhand der zuletzt kodierten Bits wird dieses aus einer Auswahl verschiedener Modelle ausgesucht. Für das erste Bit eines Codewortes kann nur zwischen wenigen Modellen gewählt werden, wobei sich die Entscheidung hierbei auf die bereits kodierten entsprechenden Codewörter aus den Blöcken rechts und oberhalb des aktuellen Blockes stützt. Danach stehen zusätzliche Modelle bereit. Für jedes syntaktische Element bietet der AVC-Standard sowohl spezielle Verfahren zur Binarisierung als auch Wahrscheinlichkeitsmodelle. Im September 2002 waren bereits 267 verschiedene Modelle definiert.[6]

2.2.8 Deblocking-Filter

Bei einer vorgegebenen Datenrate kann es passieren, dass für die korrekte Kodierung einiger Makroblöcke nicht genügend Bits zur Verfügung stehen. Betroffen sind davon vor allem Bildinhalte mit vielen Details oder ungleichmäßigen Bewegungen, da dort die oben aufgeführten Kompressionsverfahren kein gutes Ergebnis erzielen können. Die einzige Möglichkeit, die Datenrate trotzdem einzuhalten, ist die Verstärkung der verlustbehafteten Quantisierung. Dadurch werden jedoch die Blockränder deutlich im Bild sichtbar, was subjektiv als grobe Störung empfunden wird.

Als Gegenmaßnahme beinhaltet der AVC-Standard ein Deblocking-Filter, welches auf jeden dekodierten Makroblock angewendet wird. Das geschieht im Kodierer noch bevor der Makroblock für weitere Prädiktionen gespeichert wird. Dadurch lassen sich zwei entscheidende Vorteile ausschöpfen. Zunächst erreicht man durch die Glättung der Blockkanten vor allem bei hohen Kompressionsraten eine deutliche Reduktion der Blockartefakte. Diese sind für den menschlichen Betrachter besonders störend, so dass das Deblocking-Filter eine erhebliche subjektive Qualitätssteigerung bietet. Desweiteren wird der gefilterte Makroblock zur Bewegungskompensation benutzt, woraus eine bessere

Prädiktion und somit ein kleineres Restfehlersignal resultiert. Bei der Intra-Prädiktion wird jedoch der ungefilterte Makroblock verwendet!

Zur Filterung wird jeder Makroblock in 4x4-Blöcke zerlegt. Die Luminanz also in 16 Blöcke, die Chrominanz in 4 Blöcke. Abbildung 6 zeigt anhand der gestrichelten Linien, wo die zu filternden Kanten bzw. die Grenzen zwischen den Blöcken liegen.

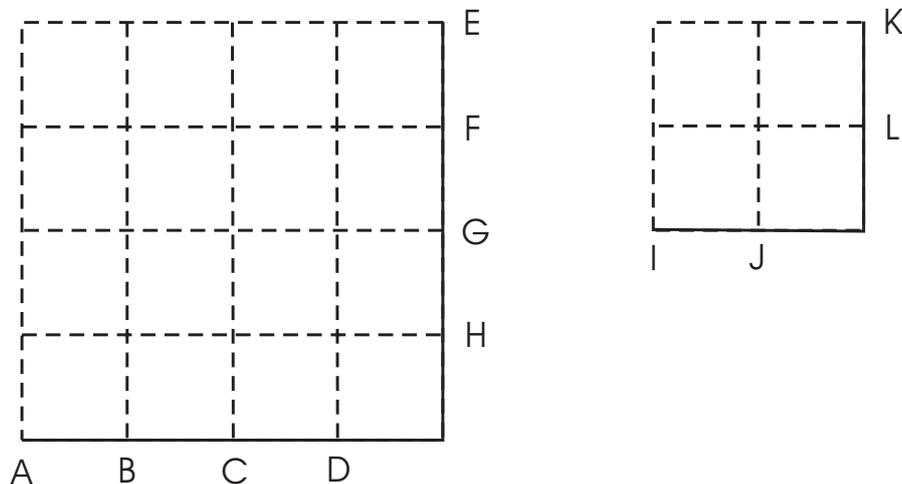


Abbildung 6: Kantenfilterung beim 16x16 Luminanz- und 8x8 Chrominanzblock

Es gibt also jeweils vier vertikale und horizontale Grenzen im Luminanzblock, sowie jeweils zwei vertikale und horizontale Grenzen im Chrominanzblock. Auf diese Kanten wird das Filter angesetzt. Dabei können bis zu drei Pixel rechts und links neben einer vertikalen, bzw. oberhalb und unterhalb einer horizontalen Grenze, zur Berechnung herangezogen werden. Um die benötigte Stärke der Filterung zu ermitteln, werden zwei wichtige Parameter berücksichtigt. Das ist zum einen die boundary strength (Bs). Sie kann Werte zwischen 0 und 4 annehmen und beinhaltet damit beispielsweise, ob es sich bei der entsprechenden Kante um eine Grenze innerhalb des Makroblocks oder der Grenze zu einem Nachbarblock handelt. Für Intra-kodierte Blöcke mit einer Kante am Rand des Makroblocks wird $B_s = 4$ und es bedarf somit einer starken Filterung. Der zweite Parameter, der zur Filterwahl genutzt wird, ist der Gradient. Dazu wird die Differenz der Pixel auf beiden Seiten der Grenze ermittelt und anhand bestimmter Schwellen der entsprechende Filter gewählt.

Eine wichtige Entscheidung ist vorallem, ob eine scharfe Kante, also ein großer Gradient, überhaupt gefiltert werden darf. Schließlich sind bei Weitem nicht alle scharfen Kanten Blockfehler, sondern gehören meistens in das Originalbild hinein. Unter Berücksichtigung des Quantisierungsparameters QP werden die Entscheidungsschwellen so angepasst, dass mit größtmöglicher Wahrscheinlichkeit richtig entschieden wird. Leicht nachvollziehen kann man das anhand eines Beispiels: Ist die Quantisierung sehr fein eingestellt, ist es äußerst unwahrscheinlich, dass es sich bei einem großen Gradienten um einen Quantisierungsfehler handelt. In diesem Fall liegt es näher, dass diese Kante einen wichtigen

Bildinhalt darstellt, der nicht durch die Filterung geglättet werden darf. [7] Abbildung 7 verdeutlicht die sichtbare Qualitätsverbesserung durch das Deblocking-Filter.



Abbildung 7: Links ohne Filter, rechts mit Deblocking-Filter

2.3 Profiles und Levels

H.264/AVC ist gleichermaßen für die Videotübertragung auf schmalbandigen Kanälen also auch für hochauflösendes HD-TV geeignet. Um diesen sehr unterschiedlichen Ansprüchen gerecht zu werden, sind auch für diesen Standard verschiedene Profile definiert, wie es bei MPEG bisher immer üblich war. Insgesamt werden drei Profile angeboten. In jedem Profil ist festgelegt, welche Kodierungsmethoden unterstützt werden müssen. Sie heißen Baseline, Main und Extended.

Baseline ist für geringe Datenraten spezifiziert, so dass diverse Verfahren in diesem Profil nicht enthalten sind. Betroffen sind davon B-, SP- und SI-Slices, die gewichtete Prädiktion und CABAC. Außerdem stehen keine Methoden für die Kodierung von Videosignalen mit Zwischenzeilenverfahren zur Verfügung.

Auf dem Main Profile liegt sicherlich das Hauptaugenmerk. Es ist vor allem für die digitale Fernsehübertragung gedacht und definiert bereits die meisten Kodierverfahren. Ausgenommen sind jedoch die SP- und SI-Slices. Außerdem kommt das Flexible Makroblock Ordering nicht mehr vor. Im Baseline Profile ist es enthalten.

Das Extended Profile ist für die hochauflösenden Anwendungen geeignet und bietet nahezu die volle Bandbreite der Kodiermethoden. Lediglich CABAC wird nicht unterstützt.

Die Level gelten für alle Profile gleichermaßen und legen bestimmte Parameter wie beispielsweise die Auflösung fest. Diese Level müssen vom Dekodieren unterstützt werden. Insgesamt existieren elf verschiedene Level mit zunehmenden Anforderungen an die benötigte Hardware. Sie umfassen alle wichtigen Bildgrößen von QCIF bis zu Kino geeigneten Formaten. [1]

3 Neue Anwendungsmöglichkeiten

Aufgrund der definierten Profile und Level deckt H.264/AVC ein breites Anwendungsspektrum ab. Systeme mit relativ niedrigen Bitraten wie z.B. UMTS erhalten dadurch erstmals eine Möglichkeit, Videos in brauchbarer Qualität zu übertragen und auch bei Videokonferenzen wird es voraussichtlich zur Anwendung kommen. Für tragbare Geräte wie beispielsweise Handys oder PDAs wird allerdings eher die Verwendung von DVB-H angepeilt. Im Zuge von hybriden Netzen könnte Pay-TV über UMTS bestellt und über DVB-H geguckt werden. Für die Übertragung von DVB-H-Inhalten ist H.264/AVC angedacht, eine endgültige Abstimmung gab es allerdings noch nicht.

Gleichzeitig profitieren die Rundfunkanstalten und damit einhergehend auch alle TV-Zuschauer, da nicht nur die Qualität der übertragenen Fernsehprogramme steigt, sondern auch zusätzliche Kanäle durch kleinere Bandbreiten geschaffen werden. Besonders in Deutschland existiert aufgrund der hohen Anzahl an Nachbarländern und vieler Regionalsender eine sehr große Auslastung der verfügbaren Sendefrequenzen. Dies führt schon bei der Einführung von DVB-T soweit, dass simultan analoge Fernsehprogramme abgeschaltet werden müssen. Die geplante Übertragung von HDTV-Inhalten würde bei herkömmlicher MPEG-2-Kodierung gleich fünf normale Kanäle belegen. Und nicht nur die Übertragung, sondern auch die Speicherung von Videos im HDTV-Format hat ihren Nutzen am neuen Standard. Sie ist nicht mehr allein von der Neuentwicklung von DVDs mit höherer Kapazität, z.B. Blue-ray Disc, abhängig. Time Warner hat es sich beispielsweise zum Ziel gesetzt, Videos im HD-Format auf einer herkömmlichen DVD9, also mit Doppellayer, unterzubringen. Dazu befinden sich H.264/AVC sowie WMV9 als Kompressionsverfahren in der engeren Auswahl.[1]

4 Vergleich zur Konkurrenz

Da auch die Konkurrenz wie in den meisten Fällen nicht schläft, gibt es einige weitere ernstzunehmende Mitstreiter unter den Videocodierverfahren. Als besonders bedeutsam gilt dabei Microsofts WMV9, für welches bereits gut funktionierende Implementierungen existieren. Auf der Cebit 2004 wurde sogar bereits eine Demo-HD-DVD mit einer WMV9-Kodierung vorgestellt, während H.264/AVC zurzeit noch zu aufwendige Hardware benötigt, um als marktreif bezeichnet werden zu können. MainConcept bietet beispielsweise schon funktionierende Software als Demoversion an, diese ist aber noch nicht geschwindigkeitsoptimiert, zeigt aber bereits eine vielversprechende Qualität AVC-kodierter Videos. Viele Firmen sträuben sich außerdem gegen das proprietäre Microsoftformat und würden einen Standard wie AVC bevorzugen. Leider standen zum Zeitpunkt der Fertigstellung dieses Dokumentes noch keine aussagekräftigen Vergleichstests mit Konkurrenzprodukten zur Verfügung.

5 Fazit

Der H.264/AVC-Standard ist eine intelligente Verbesserung vorheriger MPEG-Standards. Viele neue Ideen, aber auch einfach mehr Möglichkeiten bei der Auswahl geeigneter Verfahren zur Prädiktion oder Kodierung, führen zu einer signifikanten Datenreduktion bei gleicher Bildqualität. Dieses Dokument kann nur einen Überblick über diese vielen und teilweise sehr komplexen Verbesserungen geben. Doch nicht nur für den menschlichen Betrachter erscheint der Standard sehr aufwändig. Für die Kodierung und Dekodierung in Echtzeit sind zurzeit nur die aktuellen Prozessoren fähig. Es sind jedoch noch Laufzeit-optimierungen zu erwarten. Der Konkurrenz scheint dieser offene Standard im Moment noch etwas hinterher zu hinken, was auf unzureichende Implementierungen zurück zu führen ist. Technisch hat H.264/AVC offensichtlich bereits viele Anwender überzeugt. Große Rundfunkanstalten werden aber sicherlich noch einige Zeit abwarten, ob sich dieser Standard tatsächlich etabliert.

Literatur

- [1] Thomas Wiegand Karsten Sühning, Heiko Schwarz. Effizienter kodieren. *c't*, (6):266–273, 2003.
- [2] H.264/mpeg-4 part 10 prediction of intra macroblocks. White Paper, <http://www.vcodex.com>, April 2003.
- [3] H.264/mpeg-4 part 10 prediction of inter macroblocks in p-slices. White Paper, <http://www.vcodex.com>, April 2003.
- [4] H.264/mpeg-4 part 10 transform and quantisation. White Paper, <http://www.vcodex.com>, March 2003.
- [5] H.264/mpeg-4 part 10 variable-length coding. White Paper, <http://www.vcodex.com>, October 2002.
- [6] H.264/mpeg-4 part 10 context-based adaptive arithmetic coding. White Paper, <http://www.vcodex.com>, October 2002.
- [7] H.264/mpeg-4 part 10 reconstruction filter. White Paper, <http://www.vcodex.com>, April 2003.
- [8] Heiko Schwarz Ralf Schäfer, Thomas Wiegand. The emerging h.264/avc standard. *EBU Technical Review*, January 2003.
- [9] H.264/mpeg-4 part 10 overview of h.264. White Paper, <http://www.vcodex.com>, October 2002.