



**HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFT  
UND KUNST**

**Fakultät Naturwissenschaft und Technik  
Göttingen**

-

**Studienarbeit im Fach  
Audio- und Videosysteme  
Thema:**

## **H.264 Codec und Medienformate**

Eingereicht von:  
Sebastian Basner  
Matr.Nr.:512242

6. Januar 2010

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ziel dieser Arbeit . . . . .	1
<b>2</b>	<b>Die Geschichte von H.264</b>	<b>2</b>
2.1	Die Entwicklung der Codecs . . . . .	2
2.2	Die H.XXX Codecfamilie . . . . .	3
2.3	Einsatzgebiete und Verbreitung . . . . .	4
<b>3</b>	<b>Funktionsweise</b>	<b>5</b>
3.1	Komprimierung . . . . .	5
3.1.1	Frames und Predictions . . . . .	6
3.1.2	Entropiekodierung . . . . .	6
3.2	Container . . . . .	7
3.3	Profile und Level . . . . .	8
<b>4</b>	<b>Alternativen zu H.264</b>	<b>9</b>
<b>5</b>	<b>Ausblick</b>	<b>10</b>
<b>6</b>	<b>Literaturverzeichnis</b>	<b>11</b>

# 1 Einleitung

H.264 –auch MPEG4-AVC (Advanced Video Codec)– ist ein besonders effizienter Videokompressionscodec, der von der ITU (International Telecommunication Union) entwickelt wurde. Er genießt heute große Verbreitung und wird bewusst oder unbewusst von beinahe jedem genutzt. Er ist sowohl im Bereich des Streaming als auch in der lokalen Videoverarbeitung oder auf dedizierten Geräten wie z.B. Blu-Ray Playern zu finden und auf allen gängigen PC Plattformen verfügbar.

## 1.1 Ziel dieser Arbeit

In dieser Arbeit wird der Videocodec H.264, seine Geschichte und heutige Bedeutung mit seinen verschiedenen Einsatzgebieten vorgestellt. Es wird herausgearbeitet wo die Abgrenzung zwischen Codec und der durch ihn entstehenden Datei ist, sowie auf welchem Prinzip seine besonders effiziente Komprimierung beruht. Des weiteren werden in Kapitel 4 die Alternativen zu H.264 aufgezeigt. Im letzten Kapitel wird aufgezeigt welche weiteren Einsatzmöglichkeiten für den Codec noch existieren, sowie vor welchen Problematiken er steht.

## 2 Die Geschichte von H.264

H.264 ist keine komplette Neuentwicklung, sondern steht am Ende eines langen Entwicklungsprozesses. Er hatte bereits diverse Vorgänger, wie z.B. den H.263 aus dem er schließlich hervor ging.

### 2.1 Die Entwicklung der Codecs

Die ersten für die breite Masse interessanten und zugänglichen Videocodecs waren Anfang der 90er Jahre Apples Quicktime und der MPEG1. Diese Codecs waren noch nicht besonders effizient und wurden so gut wie ausschließlich für kurze Clips, die lokal abgespeichert waren, genutzt. Der erste Quicktime Codec konnte eine Auflösung von  $156 \times 116$  Pixel produzieren und die Videos hatten eine Bildrate von ca. zehn Frames pro Sekunde [Har09].

Mit dem Einsetzen des Internetbooms ergaben sich andere Anforderungen an einen Codec als bis dahin angenommen. Aufgrund der damaligen Bandbreiten im Internet war es hauptsächlich wichtig, dass die Dateien möglichst klein waren. Das führte dazu, dass in der Mitte der 90er Jahre besonders die Produkte der Firma „RealNetworks“ besonders populär wurden [Pla]. Diese stachen dadurch hervor, dass sie besonders stark komprimierte Dateien produzierten, die optimal für die Verbreitung in zunächst Audio- und später Videostreams waren, obwohl sie nur eine mäßige Qualität hatten. Hiermit gelang auch erstmals eine „Echtzeitübertragung“ von Mediendaten über das Internet. Besonders im Bereich der Windows betriebenen Computer begann ungefähr gleichzeitig der Intel Indeo Codec populär zu werden, der lange Zeit das Maß der Dinge blieb und Grundlage für Microsofts „Video for Windows“ Technologie, die im AVI (Audio Video Interleave) Containerformat Anwendung fand [Eid99].

Vor allem um die Bandbreiten- und Ressourcenprobleme, die Videotelefone genauso wie mobile Geräte und PCs betrafen, in den Griff zu bekommen, begann das MPEG (Motion Picture Experts Group) Konsortium mit der Arbeit am MPEG-4 Standard. In diesem werden Verfahren zur Video- und Audiokkomprimierung festgeschrieben. Da das MPEG Konsortium aus relativ vielen Mitgliedern besteht, ist es allerdings immer schwierig zu einer schnellen Einigung zu kommen und so wurden der halb fertige Standard von Microsoft aufgegriffen und ein Codec dazu

entwickelt. Dieser Codec wurde von einem französischen Hacker freige patched und unter dem Namen „DivX“ veröffentlicht [Com09]. Dieser Codec hatte bereits eine sehr gute Komprimierung und wurde später offiziell in den MPEG-4 Standard aufgenommen.

## 2.2 Die H.XXX Codecfamilie

Wie bereits in der Einleitung erwähnt, wurde der H.264 Codec –wie alle H. Codecs– von der Internationalen Telekommunikationsunion entwickelt. Da es sich hierbei um eine Telekommunikationsorganisation handelt, waren die dort entwickelten Codecs zu Anfang auch für den Einsatz im Telefonnetz gedacht. Die Entwicklung dieser Codecs entstand aus dem Anspruch heraus auch hier Bilder über ein Netz übertragen zu wollen, aber aufgrund der schlechten Qualität und der nicht erfüllten Echtzeitanprüche kamen die oben erwähnten Codecs hier nicht in Frage. Die so entwickelten Vorgänger von H.264 (H.261 und H.263)[itwa] [Zö]sollten daher ursprünglich im ISDN (Integrated Services Digital Network) Netz für die Videoverarbeitung bei der Videotelefonie zuständig sein. Dieses Feature setzte sich bekanntermaßen nie durch, aber der Codec fand weiterhin Verwendung in der Videoverarbeitung außerhalb der Telefonnetze und so wurde H.263 aufgrund seiner Verfügbarkeit bei Verabschiedung des Standards, sowie der Einhaltung der dort beschriebenen Verfahren direkt unverändert in den MPEG-4 Standard aufgenommen.

Bei H.264 stand erstmals nicht mehr das Videotelefon im Fokus sondern es sollte ein Codec entwickelt werden, der, besonders im Hinblick auf HD Video, die Datenraten deutlich senken sollte. Für H.264 wurde nun hauptsächlich die Effizienz erhöht, indem man die Zeitspanne, in der ein voller Frame im Video erscheinen muss, verlängerte und eine Entropiekodierung einführte (mehr dazu in Kapitel 3.1.2). Der so entstandene Codec konnte folglich eine höhere Videoqualität mit weniger Speicherverbrauch ermöglichen und war so gut, dass man sich mit dem MPEG Konsortium zusammensetzte und ihn nachträglich in den MPEG4 Standard als „Part 10“ aufnahm und mit dem Kürzel AVC (Advanced Video Coding) versah [itwb] [Fac08]. Die restlichen Videocodecs sind im sogenannten „Part 2“ unter dem ASP (Advanced Simple Profile) zusammengefasst. Hierzu zählen Codecs wie XviD oder der verwandte DivX [Enc09b].

## 2.3 Einsatzgebiete und Verbreitung

Der H.264 Codec gewann schlagartig an Bedeutung als er 2007 in die Version 9 von Flash aufgenommen wurde [Zot07] und in der Folge der Webdienst YouTube verkündete den Codec zu verwenden [Sch08]. Kurz vorher hatte bereits der Computerhersteller Apple verkündet auf den Codec zu setzen, alle seine Kunden per Update mit dem Codec versorgt und diesen in „Quicktime“ eingebunden [mob07]. In der Folge wurden Programme wie der Videochatclient iChat mit H.264 in sehr guter Qualität nutzbar. Auch in seinen Set-top Boxen und iPods setzte Apple den Codec in der Folge ein. Für Linux ist der Codec unter anderem im „ffmpeg Projekt“ vorhanden das in der Standardkomponente gStreamer enthalten ist und wird auf diesem Weg mitgeliefert. Einzig Microsoft weigerte sich den Codec zu unterstützen und versuchte mit seinem Konkurrenzformat Windows Media Video in den Markt zu drängen. So ist es selbst mit einem Windows Vista nicht möglich ohne die Installation von Drittsoftware ein H.264 Video zu schauen. Dies soll sich nun mit Windows 7 aber auch endgültig geändert haben.

Des weiteren findet man H.264 heute in Blu-Ray Playern, HDTV DVB Receivern, Smartphones, Videokameras und vielen anderen Geräten.

## 3 Funktionsweise

H.264 ist wie eigentlich alle modernen Videocodecs ein DCT Codec. DCT steht für Discrete Cosine Transformation und bedeutet, dass Signale aus dem Zeitbereich in den Frequenzbereich transformiert werden. Dieses Verfahren wird z.B. auch bei JPEG angewandt. Für H.264 wurde es dahingehend modifiziert, dass keine reine DCT auf 8x8 Pixel große Blöcke angewandt wird sondern eine sogenannte Integertransformation auf einen 4x4 Pixel großen Block. Das hat den Hintergrund, dass bei der Integertransformation keine Rundungsfehler auftreten können und es keine Fehlerfortpflanzung gibt. Auch was die Rechenintensität angeht, ist sie der reinen DCT überlegen, da sie nur mit Additionen, Subtraktionen und Bit Verschiebungen auskommt, aber keinen Divisionen und Multiplikationen benötigt [pd06].

Grundsätzlich lässt sich sagen, dass alle DCT einsetzenden Codecs sich ähnlich sind was die Algorithmen angeht und sich vor allem durch die eingesetzten Features –wie z.B. B-Frames– voneinander unterscheiden.

### 3.1 Komprimierung

Um hochqualitatives Videomaterial möglichst klein zu halten, ist eine entsprechend effektive Komprimierung erforderlich, die das sichtbare Bild dennoch soweit unverändert lässt, dass sich keine offensichtlichen Einschränkungen der Bildqualität ergeben. Es ist also nicht möglich (wie z.B. bei MP3) einfach ein paar Bildanteile abzuschneiden, da dies sofort sichtbar wäre. Stattdessen wurden einige spezielle Methoden entwickelt, die den Kern von H.264 bilden. Diese sind so effizient, dass sich eine 9GB DVD mit einigen optischen Abstrichen auf die Größe einer CD bringen lässt.

### 3.1.1 Frames und Predictions

Der Hauptteil der Kompression von H.264 erfolgt durch die Einsparung von möglichst vielen vollständigen Frames des Videos. In H.264 gibt es drei Sorten von Frames:

- I-Frames
- P-Frames
- B-Frames

Die I-Frames (Intra-Frames) sind einfache vollständige Frames wie beispielsweise ein JPEG auch. Von Ihnen werden die P- und B- Frames abgeleitet. P-Frames (Prediction-Frames) wurden bereits mit MPEG1 eingeführt, aber mit späteren Codecs immer weiter verbessert. Hier werden keine kompletten Frames mehr übertragen, sondern es wird vom letzten I-Frame ausgegangen und ab da nur noch eine Differenzinformation übertragen. Im P-Frame steckt also die Information was sich seit dem letzten I-Frame geändert hat. Eine Weiterentwicklung der P-Frames sind die mit MPEG-4 eingeführten B-Frames (Bidirectional-Frames). Wie der Name schon sagt beziehen sich die B-Frames nicht nur auf die vorhergehenden I-Frames, sondern können bereits Differenzinformationen zu zukünftigen I-Frames enthalten [m2s].

Eine Besonderheit von H.264 ist die „Long Term Prediction“ die es den P- und B-Frames ermöglicht sich nicht nur auf den jeweils letzten oder nächsten I-Frame zu beziehen, sondern auch auf beliebig weit zurückliegende I-Frames, wodurch insgesamt I-Frames eingespart werden können. Einige Optimierungen von H.264 zielen darauf ab möglichst lange nur Differenzbilder und möglichst wenig I-Frames zu übertragen. Da dies sowohl bei lokal gespeicherten Videos als auch bei gepufferten Videostreams im Internet gemacht wird, ist so der Effekt zu erklären den man beobachten kann, wenn man in einem Onlinevideo zu einer Stelle weit in der Zukunft oder der Vergangenheit springt und anstatt des erwarteten Bildes noch einige Sekunden das Vorhergehende Bild sieht, in dem die neuen Bewegungen stattfinden. Es fehlt der passende I-Frame zu den Differenzbildern und es wird von dem I-Frame, den man vor dem Springen an die neue Stelle hatte, ausgegangen, bis ein neuer und aktueller I-Frame kommt.

### 3.1.2 Entropiekodierung

Entropiekodierung ist ein Verfahren zur verlustfreien Datenkompression. Es wird jedes Datensymbol mit einem Codewort kodiert. Tritt ein Datensymbol besonders häufig auf, bekommt es ein kurzes Codewort, tritt es dagegen selten auf, wird ihm



ein längeres Codewort zugeordnet. Auf diese Art und Weise kann eine sehr gute verlustfreie Komprimierung ermöglicht werden. Es gibt verschiedene Verfahren der Entropiekodierung.

Bis H.264 war die sogenannte CAVLC (Context Adaptive Variable Length Coding) Methode die effektivste Entropiekodierung. Mit H.264 wurde die sogenannte CABAC (Context-adaptive binary arithmetic coding) Methode eingeführt. Hier kann im Gegensatz zur CAVLC Methode die Kodiertabelle noch während des Kodierens dynamisch angepasst werden. Dies erfordert jedoch auch mehr Rechenleistung und ist für schwächere PCs oder mobile Geräte ungeeignet, daher sind heute beide Methoden im Standard enthalten [vfer]. Im direkten Vergleich bringt CABAC einen Performancegewinn von ca. 5 bis 15 % .

### 3.2 Container

Da H.264 nur ein Codec ist und kein eigenes Dateiformat mitbringt, müssen die codierten Dateien entsprechend verpackt werden um sie abspielbar zu machen. Ein Container gibt bestimmte Rahmenbedingungen vor, die an den Inhalt gestellt werden. Er kann z.B. bestimmen, dass ein Video grundsätzlich mit einer hohen Bitrate codiert wird, aber an einer bestimmten Stelle nur eine niedrigere eingesetzt wird. Das meistgenutzte Format ist Microsofts AVI Container. Ein Container sagt zunächst mal nichts über das darin enthaltene Material aus und so kann in einem AVI oder Quicktime Container z.B. H.264, H.263, oder aber auch ein DivX/XviD codiertes Video sein. Zum Abspielen der Videos braucht man dann entsprechend den richtigen Codec. Einen AVI Codec beispielsweise gibt es nicht [Falar]. Um nun festzustellen um was für ein Format es sich tatsächlich in dem Container handelt, verwendet AVI beispielsweise den FourCC (Four Character Code). Es handelt sich dabei um eine 32 Bit Sequenz aus 4 ASCII Zeichen, die über den Inhalt Auskunft gibt. Da AVI seit 96/97 von Microsoft nicht mehr weiterentwickelt wird, es aber natürlich neue Codecs gegeben hat, gibt es keine einheitliche Benennung der FourCCs mehr und hier sind einige Inkompatibilitäten entstanden [mLer].

Trotz alledem setzte sich zunächst für viele neue Codecs das AVI Format durch, da dies weit verbreitet war und unterstützt wurde. Dadurch wurden einige Erweiterungen durch dritte nötig, so konnte die letzte offizielle Spezifikation z.B. nicht größer als 2GB sein. Auch B-Frames waren nicht möglich und diese Funktionalität musste nachgepatched werden.

Inzwischen entstehen mehr und mehr neuere/moderne Containerformate. Das MPEG Konsortium hat für MPEG-4 einen eigenen Container mit der Endung .mp4 spezifiziert [Enc09a], aber auch freie Formate, wie der quelloffene Matroska .mkv Container [mat], bieten heute sehr viel mehr Features als AVI.

### 3.3 Profile und Level

Da H.264 sowohl extrem hochqualitative Videos, die mehr Leistung für das Betrachten erfordern, als auch „schlechtere“ Qualitäten für nicht so leistungsstarke Geräte erzeugen kann, muss man sich beim Encoding für ein Profil und ein Level entscheiden. Die Profile legen dabei die genutzten Features fest (z.B. Art der Entropiekodierung) und die Level die Bitrate, die das abspielende Endgerät vertragen muss, um das Video abspielen zu können. Hierbei gilt je höher das Level desto höher die Bitrate. Ursprünglich gab es die vier Profile **Baseline**, **Extended**, **Main** und **High**. Nachträglich kam noch die drei **High 10**, **High 4:2:2**, **High 4:4:4** hinzu [itwb].

H.264/AVC Profiles			
Baseline Profile	Extended Profile	Main Profile	High Profile
I Slices	I Slices	I Slices	I Slices
P Slices	P Slices	P Slices	P Slices
CAVLC	CAVLC	CAVLC	CAVLC
4x4 Transform	4x4 Transform	4x4 Transform	4x4 Transform
	Weighted Prediction	Weighted Prediction	Weighted Prediction
	B Slices	B Slices	B Slices
		CABAC	CABAC
			Quantization Matrix
			8x8 Transform
ASO	ASO		
FMO	FMO		
Redundant Slices	Redundant Slices		
	Data Partitioning		
	SP Slices, SI Slices		

Abbildung 3.1: Die 4 Ursprungsprofile von H.264

## 4 Alternativen zu H.264

Microsoft hat sich wie bereits erwähnt lange dagegen gesträubt H.264 zu unterstützen und versuchte stattdessen im Alleingang das Windows Media Format für Audio und Video in den Markt zu bringen. Tatsächlich schaffte man es auch das WMV-HD Format auf HD-DVD und Blu-Ray Player zu bringen. Hier ist die Lage inzwischen aber deutlich zu Gunsten von H.264 ausgefallen. Einen weiteren Vorstoß wagt Microsoft jetzt mit der WMV-HD-DVD, bei der HD Bildmaterial auf normale DVD Rohlinge gepackt wird.

Der vielleicht größere Konkurrent ist der „freie“ Ogg-Theora Codec. Der Codec war ein Geschenk der Firma On2 an die Opensource Gemeinde. Inzwischen wurde er stark weiterentwickelt. Er ist qualitativ zwischen H.263 und H.264 im Simple Profile anzusiedeln. Bislang fehlen einige Features wie die B-Frames, dafür erzeugt er bei niedrigen Bitraten kaum „Klötzchen“ sondern eher ein unscharfes Bild, was die Meisten als angenehmer empfinden. [Zot09]

Der Codec ist nicht wirklich frei, denn die genutzten Verfahren sind durch die Firma On2 patentrechtlich geschützt. Bei der Übergabe des Codecs wurde allerdings vertraglich festgehalten, dass aus diesen Patenten niemals Ansprüche abgeleitet werden. [Fro08]

## 5 Ausblick

H.264 war aktuell zusammen mit Theora im Gespräch als Standardcodec für die HTML5 Spezifikation des W3C (World Wide Web Consortium). Hierbei geht es konkret um die Einführung von Video und Audio Tags in HTML. Es sollte möglich sein ohne Techniken wie Flash ein Video, wie heute ein Bild in eine Website, einzubauen und dafür wurde der kleinste gemeinsame Nenner gesucht. Leider scheint hier keine Einigkeit erzielbar zu sein. Apple unterstützt bedingungslos H.264 und lehnt Theora ab, da dieser zu schlecht von Hardware unterstützt sei, Mozilla und Opera haben nur Theora aufgenommen und argumentieren, dass man bei H.264 Bedenken bezüglich der Lizenzsituation habe, wenn man den Browserquellcode freigeben will. Google schließt sich bei H.264 der Mozilla Argumentation an, hat aber beide Codecs aufgenommen, obwohl sie Theora für noch zu unausgereift halten. Der H.264 soll allerdings nur in Chrome und nicht im quelloffenen Chromium Einzug halten. Einzig Microsoft hat in den Internet Explorer bislang gar keine Unterstützung für das Video Tag implementiert und sich somit weder für den einen noch für den anderen entschieden. Das Problem sind die Patente von H.264. Lizenzzahlungen für die Patente fließen in einen Patent-Pool, da die Aufschlüsselung wer was bekommt sonst zu kompliziert für den einzelnen wäre. Probleme gibt es allerdings wenn ein Lizenznehmer den Codec dritten zur Verfügung stellt und dieser damit irgendwie Einnahmen generiert. In diesem Fall müssten wieder Lizenzzahlungen geleistet werden. [gol09]

Ein Ausweg wäre wohl nur in Sicht, wenn die Patente für H.264 auslaufen oder Theora technisch aufschließt.

Für künftige Codecs wird derzeit daran gearbeitet das DCT Verfahren durch eine Wavelet-Transformation zu ersetzen. Diese Art der Transformation wird heute bereits beim JPEG2000 Bildformat eingesetzt. Vorteile liegen unter anderem darin, dass mit Wavelets sehr steile Flanken besonders exakt dargestellt werden können und Über- und Unterschwinger vermieden werden. Derzeit scheitert der Einsatz dieser Transformation aber an dem beachtlichen Rechenaufwand, der für derzeitige CPUs noch nicht ohne weiteres zu leisten ist.

## 6 Literaturverzeichnis

- [Com09] Computerbase.de. DivX. Dezember 2009. <http://www.computerbase.de/lexikon/DivX>, Stand 02.01.10.
- [Eid99] Hermann Eiden. Das AVI-Format. August 1999. <http://www.tomshardware.com/de/digital-video-guide-iii-videoformate-und-kompressionstechniken,testberichte-61-2.html>, Stand 02.01.10.
- [Enc09a] Brother Johns Encodingwissen. Die Containerformate. September 2009. <http://encodingwissen.de/formate/container.html#mp4>, Stand 03.01.10.
- [Enc09b] Brother Johns Encodingwissen. Video- und audioformate. September 2009. <http://encodingwissen.de/formate/standards.html>, Stand 03.01.10.
- [Fac08] BET Fachwörterbuch. Mpeg-4/part 10. August 2008. <http://www.bet.de/lexikon/begriffe/MPEG4Part10.htm>, Stand 03.01.10.
- [Falar] Alexander Falk. AVI: Pattex für Bild und Ton. 2005 Januar. <http://www.netzwelt.de/news/69163-avi-pattex-bild-ton.html>, Stand 03.01.10.
- [Fro08] Oliver Frommel. Der Video-Codec Theora. April 2008. <http://www.linux-user.de/ausgabe/2006/07/092-theora/>, Stand 03.01.10.
- [gol09] golem.de. HTML 5 wird weder H.264 noch Ogg Theora vorschreiben. Juli 2009. [http://www.golem.de/showhigh2.php?file=/0907/68147.html&wort\[\]=html&wort\[\]=theora&wort\[\]=264](http://www.golem.de/showhigh2.php?file=/0907/68147.html&wort[]=html&wort[]=theora&wort[]=264), Stand 03.01.10.
- [Har09] PC Games Hardware. PCGH Retro 2. Dezember. Dezember 2009. <http://www.pcgameshardware.de/aid,668916/Das-erste-Quicktime-und-ein-beruehmter-id-Shooter-PCGH-Retro-2-Dezember-Retro/Wissen/>, Stand 03.01.10.
- [itwa] itwissen.info. H261. <http://www.itwissen.info/definition/lexikon/H-261-H-261.html>, Stand 03.01.10.

- [itwb] itwissen.info. H.263. <http://www.itwissen.info/definition/lexikon/H-264-advanced-video-coding-H-264-AVC-H-264.html>, Stand 03.01.10.
- [m2s] m2solutionsinc.com. LEAD H264 Encoder User Interface (4.0). [http://www.m2solutionsinc.com/help/m2pro/Filters/LEADH264Encoder%284.0%29\\_UI.htm](http://www.m2solutionsinc.com/help/m2pro/Filters/LEADH264Encoder%284.0%29_UI.htm), Stand 03.01.10.
- [mat] matroska.org. Matroska. <http://www.matroska.org/>, Stand 03.01.10.
- [mLer] msdn Library. FOURCC Codes. 2009 Oktober. <http://msdn.microsoft.com/en-us/library/dd375802%28VS.85%29.aspx>, Stand 03.01.10.
- [mob07] mobilemacs.de. Das Schweizer Taschenmesser für Videoformate. Februar 2007. <http://mobilemacs.de/2007/02/das-schweizer-taschenmesser-fur-videoformate.html>, Stand 03.01.10.
- [pd06] paradiso design.net. H.264/AVC. Oktober 2006. [http://www.paradiso-design.net/H264\\_AVC.html](http://www.paradiso-design.net/H264_AVC.html), Stand 03.01.10.
- [Pla] Player.com. Real Player Geschichte. <http://www.player.de.com/geschichte.html>, Stand 02.01.10.
- [Sch08] Simon Schmid. YouTube Qualität verbessern. März 2008. <http://netzlogbuch.de/videos/youtube-qualitat-verbessern/>, Stand 03.01.10.
- [vfer] video flash.de. H.264-Encoding CAVLC vs. CABAC. 2009 Oktober. <http://www.video-flash.de/index/h-264-encoding-cavlc-vs-cabac/>, Stand 03.01.10.
- [Zot07] Volker Zota. Kommender Adobe Flash Player unterstützt HD-Videos in MPEG-4 AVC. August 2007. <http://www.heise.de/newsticker/meldung/Kommender-Adobe-Flash-Player-unterstuetzt-HD-Videos-in-MPEG-4-AVC.html>, Stand 03.01.10.
- [Zot09] Dr. Volker Zota. Freier Videocodec Ogg Theora ist H.264 auf den Fersen. Mai 2009. <http://www.heise.de/newsticker/meldung/Freier-Videocodec-Ogg-Theora-ist-H-264-auf-den-Fersen-Update-21797.html>, Stand 03.01.10.

- [Zö] Patrick-Emil Zörner. H.263. <http://www.wikiweise.de/wiki/H.263>, Stand 03.01.10.