Videokompressionsverfahren

von MPEG-1 bis H.264 und H.265

Martin Fiedler
Dream Chip Technologies GmbH

Wozu Videokompression?

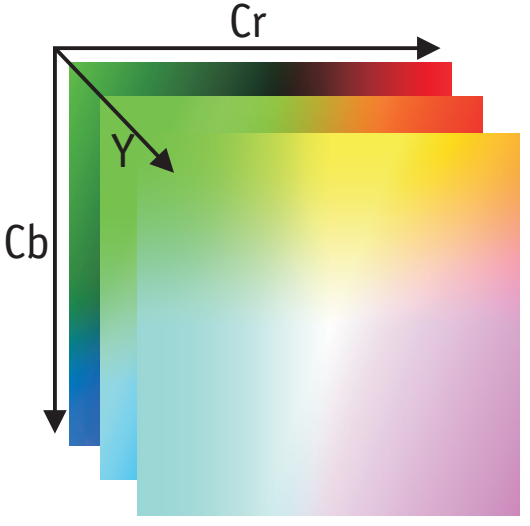
- Videokompression ist heutzutage alltäglich
  - sogar analoges Kabelfernsehen wird von digitalem Satellitenfernsehen gespeist

- Beispiel: unkomprimiertes digitales SD-(PAL-)Fernsehsignal
  720 Pixel × 576 Zeilen × 24 Bit/Pixel × 25 Bilder/Sekunde
  = 248 Mbps
  = 5 Satellitentransponder
  = 25 DVDs pro Stunde

- Grundlage der Videokompression: Standbildkompression nach dem JPEG-Verfahren

Chroma Subsampling

- Aufnahme und Wiedergabe erfolgt im RGB-Farbraum
  aber: RGB für effiziente Codierung ungeeignet!
  - für Empfinden ist die Bildhelligkeit am wichtigsten
  - bei RGB geht die Helligkeit in jeden Kanal ein

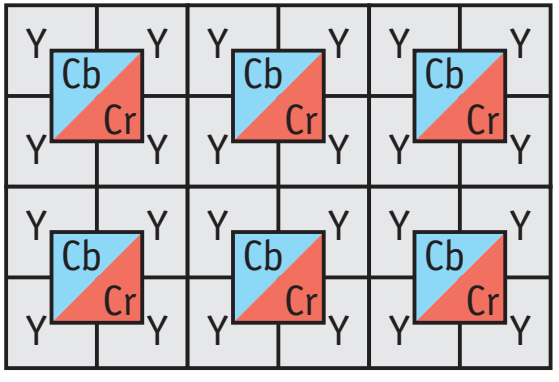


- daher Verwendung des **YCbCr**-Farbraums, ähnlich zum analogen Farbfernsehen
  - Kanal Y = Luminanz (Helligkeit)
  - Kanäle Cb und Cr = Chrominanz (Farbe)

- Konvertierung RGB/YCbCr ist linear und (theoretisch) verlustfrei
  - in der Praxis minimale Verluste durch Rundungsfehler

Chroma Subsampling

- menschliches Auge ist für Helligkeitsänderungen empfindlicher als für Farbänderungen
  – daher: Reduktion der Cb- und Cr-Kanäle auf halbe Auflösung (4:2:0-Subsampling):



- allein durch diese Maßnahme Datenreduktion um 50%

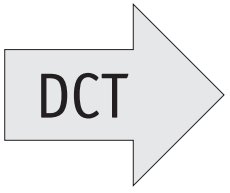
- Verluste nur an scharfen farbigen Kanten sichtbar

8x8 DCT

- Zerlegung des Bildes in 8x8 Pixel große Blöcke
- jeder Block wird einzeln codiert

- Transformation des Blocks mit einer separierbaren zwei-dimensionalen Diskreten Cosinustransformation (DCT)
- DCT hat ähnliche Eigenschaften wie Fouriertransformation
  - Zerlegung des Bildsignals in seine Ortsfrequenzen

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	155
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	161	160	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158



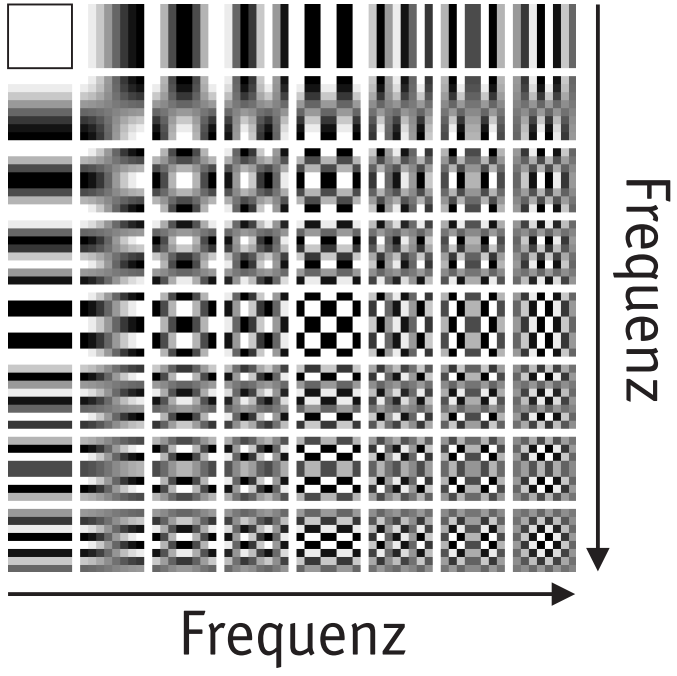
235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

8x8 DCT

- entspricht Darstellung des Blocks als Linearkombination der 64 charakteristischen Grundschwingungen:

DC-Koeffizient
= Maß für die mittlere Helligkeit

alle anderen Koeffizienten heißen **AC-Koeffizienten**



Quantisierung

- Reduktion der Datenmenge durch ganzzahlige Division der DCT-Koeffizienten

- Auge nimmt niedrige Ortsfrequenzen besser wahr als hohe
  - Divisor wird mit der Frequenz höher

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

÷

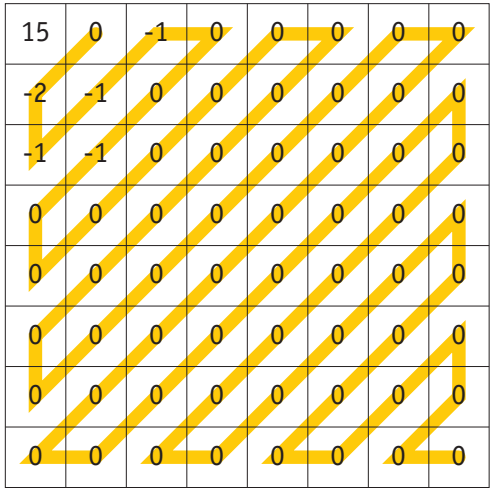
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

=

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Entropiecodierung

- Auslesen der quantisierten Koeffizienten in Zick-Zack-Reihenfolge von niedrigen zu hohen Frequenzen:



DC = 15

AC = 0, -2, -1, -1, -1, 2x 0, -1, Rest 0

spezielle Lauflängen-codierung

Sequenz kann vor-zeitig ab-gebrochen werden

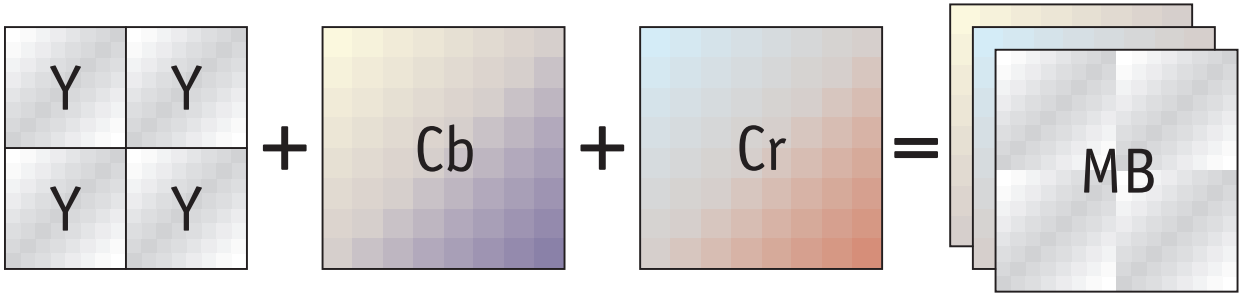
- Koeffizienten werden Huffman-codiert
  - niedrigere (häufigere) Koeffizienten haben kürzere Codes
- DC-Wert wird differenziell abgespeichert

Makroblöcke

- DCT arbeitet nur mit Graustufen
  - alle drei Farbkanäle werden getrennt DCT-codiert

- Chrominanzblöcke erscheinen im Ausgabebild wegen Subsampling in 16x16 Pixel Größe

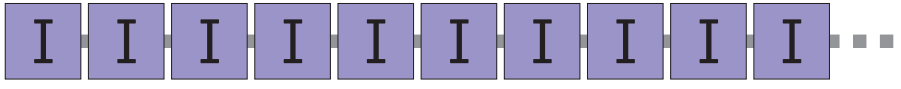
- Zusammenfassung von 4 Y, 1 Cb und 1 Cr-Block zu einem **Makroblock:**



- Grundeinheit des Bilddatenstroms sind die Makroblöcke!

I-Frames

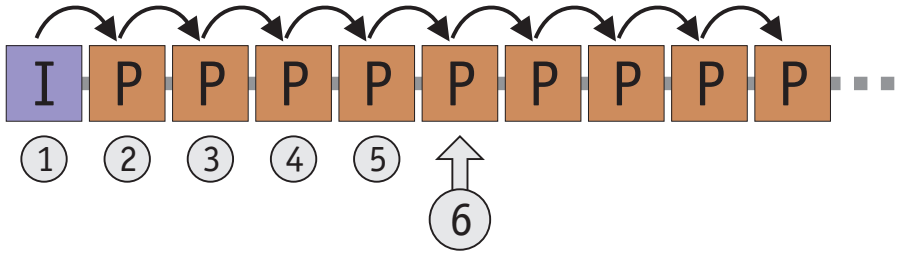
- Videodarstellung durch einfaches Aneinanderreihen von Einzelbildern (**Intra-Frames**, **I-Frames**, oder **Keyframes**)
  - **Motion-JPEG**, **DV**



- Vorteile:
  - ermöglicht direkten Zugriff auf jedes Einzelbild (ideal zum Editieren)
  - kann ein Frame nicht decodiert werden, wird er einfach ausgelassen
- Nachteil:
  - niedrige Codiereffizienz, da Abhängigkeiten zwischen Bildern nicht berücksichtigt werden

P-Frames

- es wird zunächst ein I-Frame codiert
- danach folgen Bilder, die nur die **Differenz** zum letzten Bild speichern
  - **P-Frames**, **Predicted Frames** oder **Deltaframes**

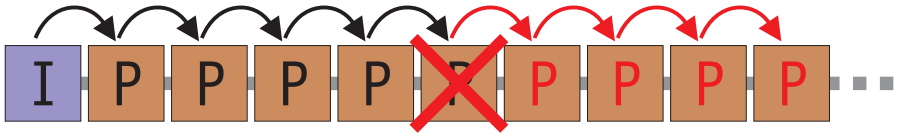


- Vorteil:
  - höhere Codiereffizienz

- Nachteil:
  - soll ein bestimmter Frame decodiert werden, so müssen zunächst alle vorherigen Frames decodiert werden!

P-Frames

- weiterer Nachteil:
  - kann ein P-Frame nicht decodiert werden, so kann auch der Rest des Videos nicht decodiert werden!



- (Teil-)Lösung für die Probleme:
  - periodisches Einstreuen von I-Frames
  - bei großen Perioden trotzdem nicht hilfreich

- **trotz allem:** Vorteil überwiegt Nachteile bei weitem!

MPEG-1

- 1992 standardisiert; zunächst nur per Hardware decodierbar

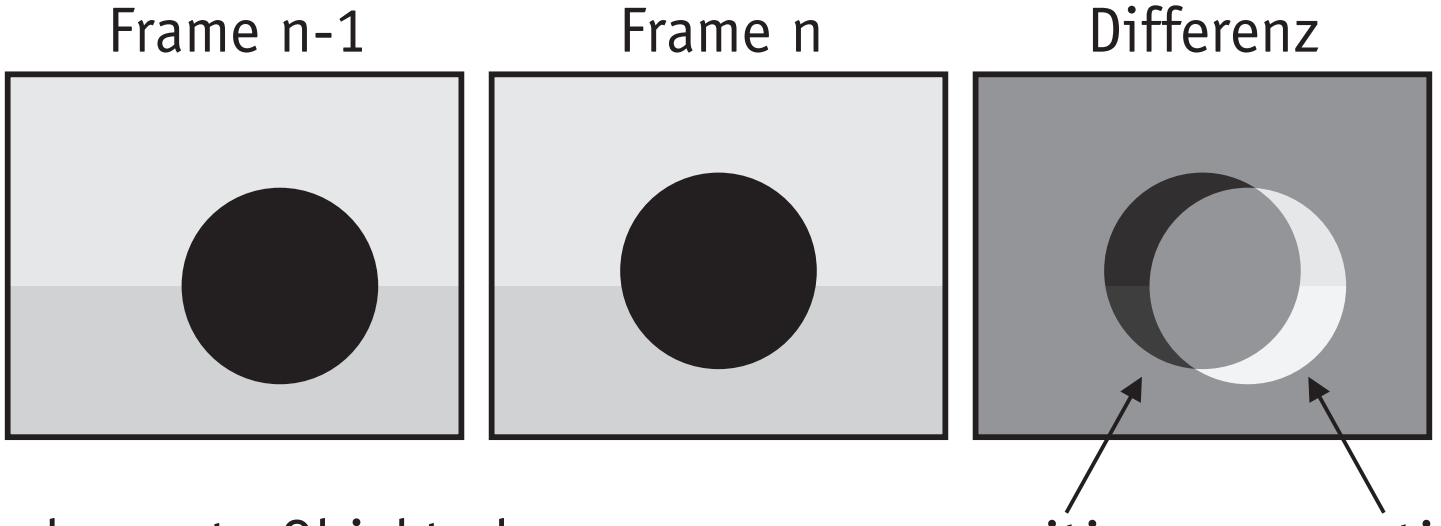
- typische Parameter: »VHS-Qualität« bei 1-1,5 Mbps
  PAL: 352x288, 25 fps; NTSC: 352x240, 30 fps

- Einsatzgebiete:
  Video-CD / CD-i

- die MPEG-1-Syntax unterstützt Videos bis 4096x4096, 60 fps; die meisten Decoder unterstützen nur **CPB**-konforme Datenströme (Constrained Parameter Bitstreams):
  max. 704 Pixel breit, max. 576 Pixel hoch, Durchsatz von 352x288/25 fps oder 352x240/30 fps, max. 1,8 Mbps

Motion Compensation

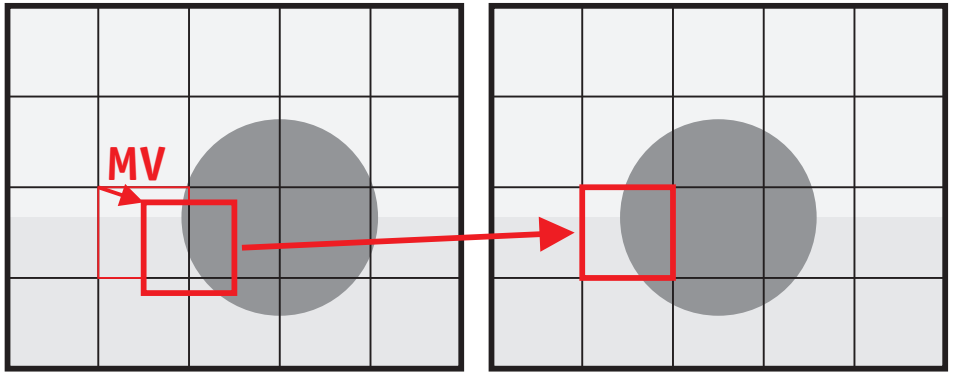
- Problem bei P-Frames: Was ist mit Bewegungen?



- um bewegte Objekte herum entstehen »Schatten«, die jedoch mitcodiert werden müssen!

Motion Compensation

- Lösung: **Motion Compensation**
  - jeder Makroblock erhält einen **Bewegungsvektor** (**Motion Vector**, **MV**), der angibt, welche Stelle im Bild als Referenz für die Differenzbildung dienen soll

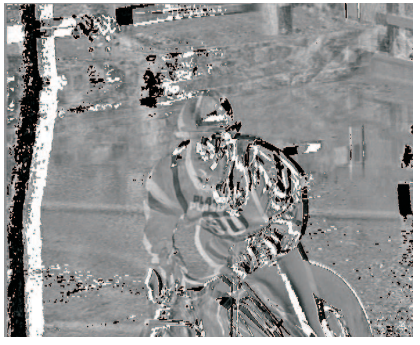


- Bewegungsvektoren bei MPEG sind auf 1/2 Pixel genau
  - Zwischenpixel werden durch bilineare Interpolation berechnet

Motion Compensation

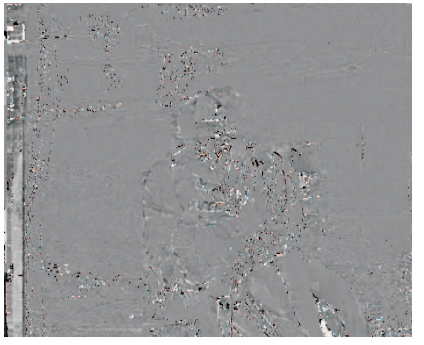
Ein Beispiel

zu codierendes Bild



Differenz zum Vorgänger (ohne MC)

Bewegungsvektoren



Differenz zum Vorgänger (mit MC)

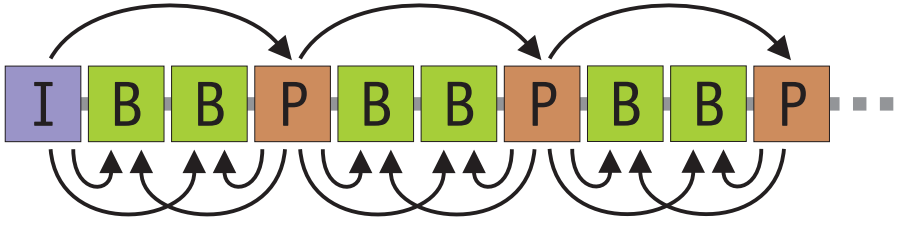
- Vorteil:
  - deutlich gesteigerte Codiereffizienz
- Nachteil:
  - Encoder wird erheblich komplexer!
    Motion Estimation = Suche nach dem optimalen MV

B-Frames

- Zusätzlich zu I- und P-Frames noch ein dritter Frame-Typ: **B-Frames** (Bi-Directional Predicted Frames)

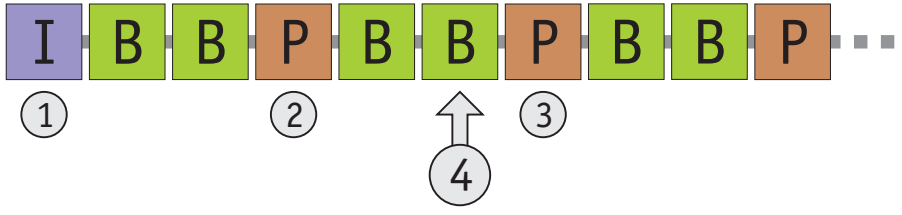
- B-Frames codieren den Unterschied zum vorigen **und nächsten** I- oder P-Frame
  - B-Frames werden jedoch ihrerseits **nicht** als Grundlage für andere Frames verwendet

- typischerweise 2-3 B-Frames zwischen zwei P-Frames

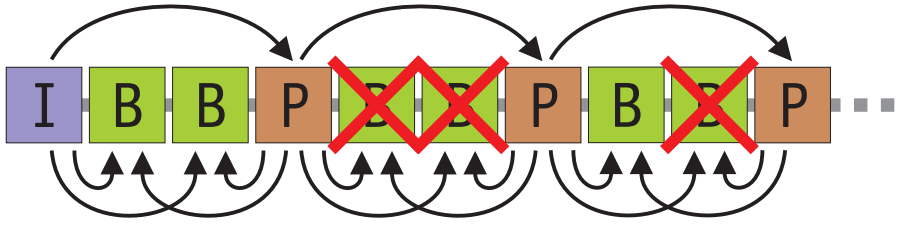


B-Frames

- Vorteile:
  - höhere Codiereffizienz
  - schnellerer Zugriff auf beliebige Bilder

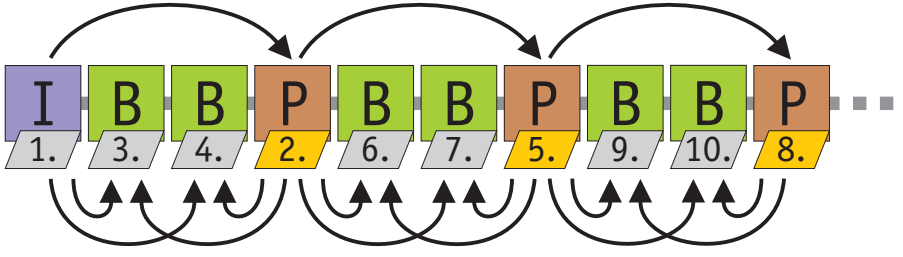


  - B-Frames können bei Problemen ohne Folgen ausgelassen werden



B-Frames

- Nachteile:
  - Codierreihenfolge entspricht nicht mehr der Anzeige-reihenfolge (nächster P-Frame muss vor B-Frames codiert werden!)



  - B-Frames leisten keinen Beitrag zu anderen Bildern (»verschwendete Bits«)
  - höhere algorithmische Komplexität, Speicherplatz- und Bandbreitenanforderungen in Encoder und Decoder

- B-Frames sind unter Experten immer noch umstritten

Codierung von P- und B-Frames

- Encoder kann für jeden Makroblock eine von zwei bzw. vier Möglichkeiten auswählen:

  P/B: Prädiktion aus letztem I/P-Frame + MV
  B: Prädiktion aus nächstem I/P-Frame + MV
  B: Prädiktion aus einer Mischung aus letztem und nächsten I/P-Frame + 2 MVs
  P/B: Intra-Codierung des Makroblocks (falls keine passende Bildstelle in den Referenzbildern gefunden wurde)
  P/B: gar keine Codierung des Makroblocks (**Skipped MB**)
  – Prädiktion wird mit Standardwerten durchgeführt

Bitratenkontrolle, Prediction

Bitratenkontrolle

- JPEG's Matrixquantisierung kann nur »konstante Qualität« codieren
  - für Video ist aber eine Steuerung der **Bitrate** (also der Ausgabegröße) wünschenswert!

- Lösung: Quantisierungmatrix wird von einem Parameter *quantizer_scale* linear skaliert
  - kann für jeden Makroblock verschieden sein

DC-Prediction, Motion Vector Prediction

- DC-Werte und Bewegungsvektoren werden nicht absolut codiert, sondern als Differenz zum letzten codierten Block

GOPs und Slices

- MPEG-Sequenz besteht aus **GOP**s (Group of Picture)
  - eine GOP ist eine Sequenz von typischerweise ca. ½ bis 1 Sekunde Länge
  - jede GOP hat ein I-Frame und eine Anzahl von P/B-Frames
  - typische GOP-Struktur: IBBPBBPBBPBBPBBIBBP...
  - Encoder kann (z.B. bei Szenenwechseln) GOP auch kürzen
  - Schnitt von MPEG-Video ist nur an GOP-Grenzen möglich!

- Frames sind in **Slices** unterteilt
  - erhöhen die Robustheit gegen Fehler
  - »Container« für eine bestimmte Anzahl von Makroblöcken
  - **Startcodes** zwischen Slices ermöglichen Resynchronisierung nach Decodierfehlern

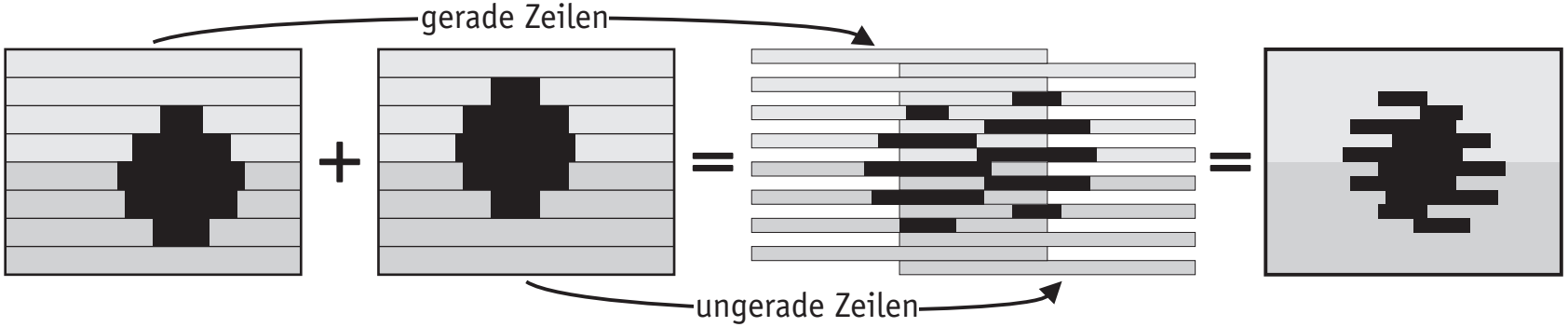
MPEG-2

- 1994 standardisiert; zunächst nur per Hardware decodierbar
- typische Leistungsmerkmale (PAL):
  720x576, 25 fps bei 2-8 Mbps
- Einsatzgebiete:
  SVCD, DVD, DVB (Digitales Fernsehen), Blu-ray Disc
- abwärtskompatibel zu MPEG-1
  - Syntax nur erweitert, nicht völlig neu konstruiert
  - jeder MPEG-2-Decoder kann MPEG-1 decodieren

Interlaced Encoding

- wichtigstes neues Feature von MPEG-2: die Möglichkeit, **Interlaced** Video zu codieren
  - nur so für Digitales Fernsehen qualifiziert!

- bei Interlaced Video besteht jeder Frame (Vollbild) aus zwei ineinander verwobenen Fields (Halbbildern):



- Problem: Fields können zu unterschiedlichen Zeiten aufgenommen sein → **Kammartefakte** bei Bewegungen

Interlaced Prediction Modes

- Encoder entscheidet für jeden **Frame** den Codierungsmodus:
  Frame Picture = Vollbild aus zwei verwobenen Halbbildern
  Field Picture = zwei Halbbilder hintereinander

- Encoder entscheidet für jeden **Makroblock** in P- oder B-Frames einen von drei Makroblock-Prediction-Modi:
  Frame Picture: Frame, Field oder Dual Prime
  Field Picture: Field, 16x8 oder Dual Prime

- Außerdem kann optional für jeden Makroblock eine **Interlaced DCT** ausgewählt werden

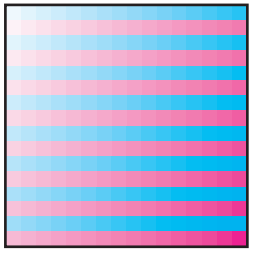
Frame Picture Prediction

Frame Prediction

- identisch mit MPEG-1-Prediction

Field Prediction

- für die ungeraden und geraden Zeilen eines MB werden jeweils eigene Predictions getroffen



- MVs erhalten zusätzlich eine Kennung, in welches Field des Referenzbildes sie zeigen
- normalerweise 2 MVs, bis zu 4 MVs in B-Frames

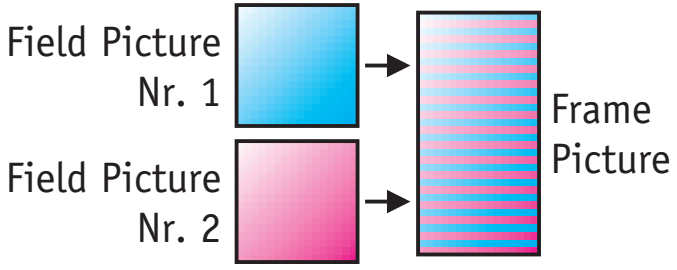
Field Picture Prediction

Field Prediction

- entspricht der MPEG-1-Prediction – allerdings auf Field Pictures bezogen

16x8 Prediction

- in Field Pictures erscheint jeder 16x16-Makroblock als 16x32 Pixel großer Bereich im Bild:



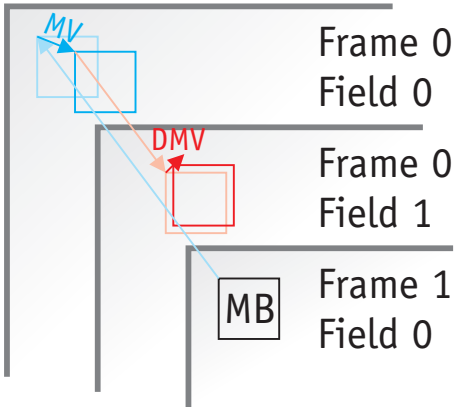
- 16x8 Prediction unterteilt den MB vertikal in zwei Hälften, um der Reduktion der »Auflösung« entgegenzuwirken

Dual Prime Prediction

Dual Prime oder **Differential Motion Compensation (DMC)**

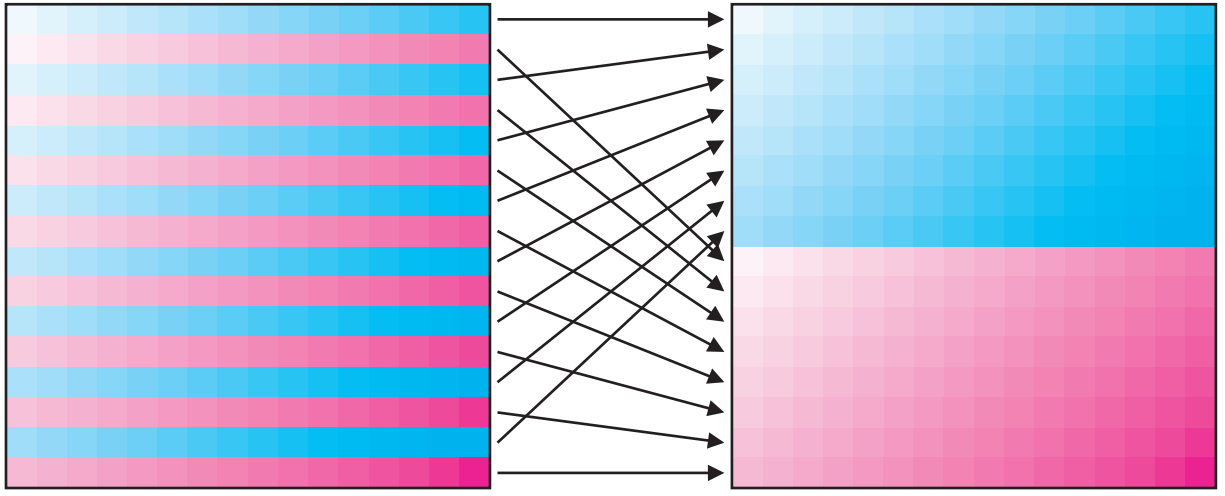
- nur in P-Frames möglich
- mischt Anteile der zwei vergangenen Fields zusammen:
  1. letztes Field gleicher Parität, per MV verschoben
  2. letztes Field entgegengesetzter Parität, mit dem selben MV und zusätzlich einem kleinen Differenz-MV verschoben

- in Frame Pictures wird für jede MB-Field-Hälfte ein getrenntes MV-DMV-Paar codiert



Interlaced DCT

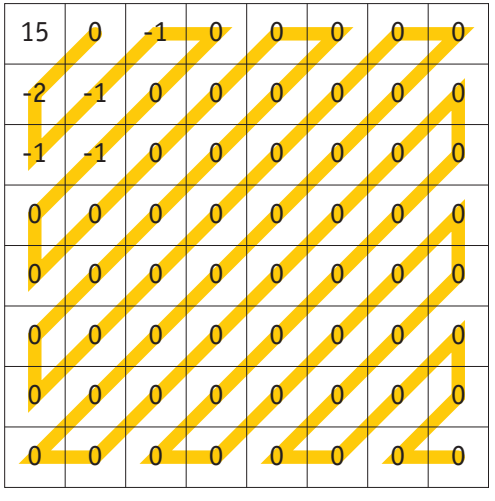
- um interlaced Material besser zu codieren, kann optional vor der DCT eine Permutation der Zeilen im Block durchgeführt werden:



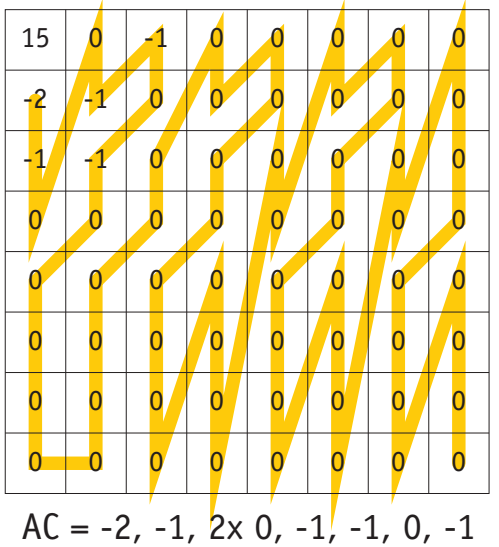
- Vorteil: Interlacing-Kammartefakte müssen nicht codiert werden

Alternative Scan-Sequenz

- alternativ zur Standard-JPEG-Zickzack-Auslesesequenz für DCT-Koeffizienten ist eine andere Reihenfolge definiert:



AC = 0, -2, -1, -1, -1, 2x 0, -1



AC = -2, -1, 2x 0, -1, -1, 0, -1

- Alternative Sequenz verbessert Entropiecodierung von Interlaced-Material

Scalability

- MPEG-2 ermöglicht die Unterteilung der Daten in mehrere verschieden priorisierte Datenströme:
  - Basisdatenstrom enthält alle Informationen, um das Video in minimaler Qualität zu rekonstruieren
  - Erweiterungsdatenströme verbessern die Qualität

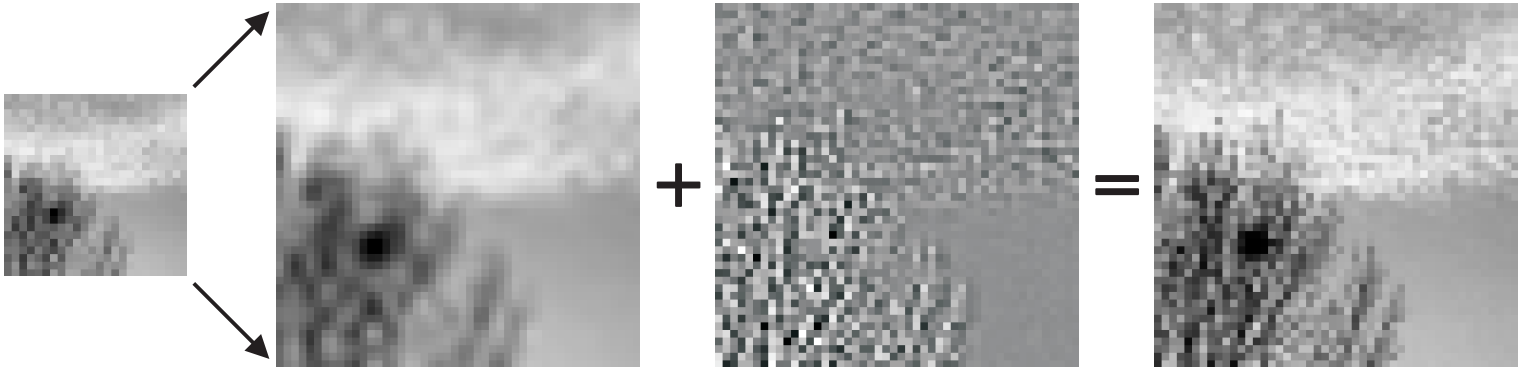
- Datenströme können mit verschieden starken Fehlerschutz-maßnahmen codiert werden

- es gibt vier Scalability-Modi

Scalability

Spatial Scalability

- Codierung des Videos in mehreren gestaffelten Auflösungen (»Bildpyramide«)
- zusätzliche Ebenen nutzen das Signal der Basisebene, skalieren es auf ihre Auflösung hoch und codieren nur den Unterschied



Basis-bild | skalierte Version | Differenz-information | hochauflösen-des Bild

Scalability

Data Partitioning

- entspricht in etwa dem Frequency Progressive Mode von JPEG
- Daten werden in zwei (oder mehr) Teile geteilt:
  1. Basisinformationen (Header, MB-Modi, MVs, ...) und niederfrequente AC-Koeffizienten
  2. höherfrequente AC-Koeffizienten

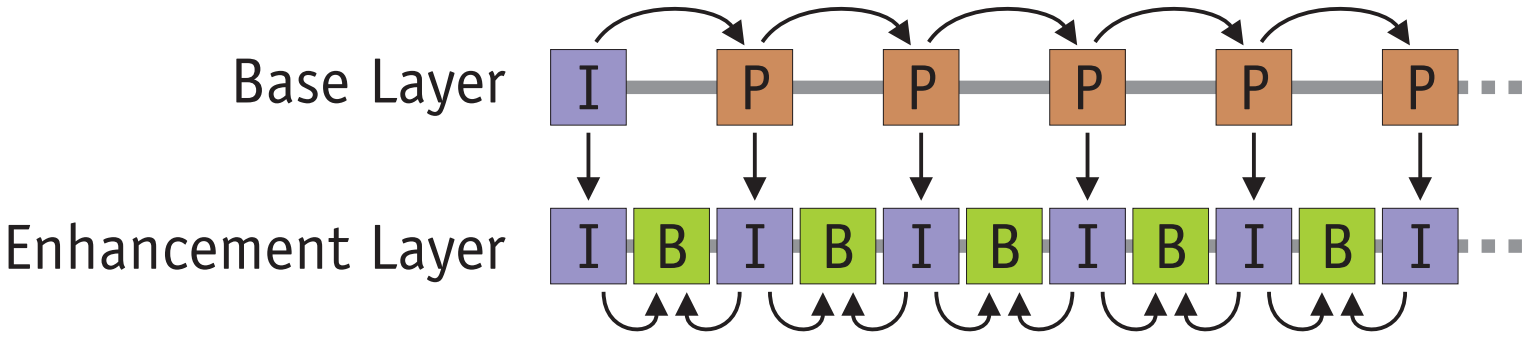
SNR Scalability

- entspricht in etwa dem Precision Progressive Mode von JPEG
- Grund-Ebene enthält nur sehr grobe DCT-Koeffizienten
- Erweiterungsebene(n) enthalten Verfeinerung (zusätzliche Bits)

Scalability

Temporal Scalability

- Erhöhung der Framerate durch Einfügen zusätzlicher Bilder in den zusätzlichen Datenströmen



- Fazit: Scalability ist toll, wird aber in der Praxis nicht verwendet
  - es existieren keine brauchbaren Implementierungen

Profiles & Levels

- Problem:
  - jede Menge Features
  - kein Decoder unterstützt alle
- Lösung:
  - Standardisierung von wohldefinierten Untermengen
- MPEG-1: Constrained Parameter Bitstream (CPB)
- MPEG-2: Profiles und Levels:
  - **Profile** = definierte Untermenge von Features
  - **Level** = Spezifikation technischen Parametern (max. Bildgröße, Bitrate, MB-Durchsatz, ...)

- momentan sind fast alle MPEG-2-Daten im **Main Profile** und **Main Level** (MP@ML) oder **High Level** (MP@HL) codiert

Profiles & Levels

MPEG-2-Profile

Simple = nur I- und P-Frames; kein Dual Prime
Main = I-, P- und B-Frames
SNR = Main + SNR Scalability
Spatial = Main + Spatial Scalability
High = höhere Präzision, besseres Chroma-Subsampling

MPEG-2-Levels

Simple = max. 352x288/25 fps oder 352x250/30 fps, 4 Mbps
Main = max. 720x576/25 fps oder 720x480/30fps, 15 Mbps
High 1440 = max. 1440x1152/30 fps, 60 Mbps
High = max. 1920x1152/30 fps, 80 Mbps

Datenströme

MPEG-2 kennt 4 Datenstromtypen:

ES (Elementary Stream)
- einzelner Video- oder Audio-Bitstrom

PES (Packetized Elementary Stream)
- in Pakete zerteilte ES mit Metadaten, z.B. Timestamps

PS (Program Stream) und **TS (Transport Stream)**
- Multiplex mehrerer Video- und Audio-PES-Ströme
- PS: variable (große) Paketlänge – z.B. VideoCD, DVD
- TS: konstante Paketlänge (188 Byte), zusätzliche Pakete zur Strukturierung (PSI) – z.B. DVB, Blu-ray
- nicht auf MPEG beschränkt, eignen sich auch z.B. für H.264

MPEG-3 ?

- ursprünglich geplant als HDTV-fähiger Nachfolger von MPEG-2, aber:
  - Standardisierung für HDTV stand bevor, so dass die Zeit für die Entwicklung zu knapp wurde
  - HDTV war auch mit MPEG-2 realisierbar

- das Audio-Kompressionsformat »MP3« hat **nichts** mit MPEG-3 zu tun – es handelt sich um **MPEG-1 Audio Layer 3**!

MPEG-4

- 1994 begonnen;
  2000 standardisiert
- zunächst in Software realisiert
- ursprünglich für Low-Bitrate-Anwendungen (Videotelefonie) konzipiert, kann aber für fast alles eingesetzt werden
- verwendet typischerweise viel längere GOPs als MPEG-1/2

DivX ;-) ...

- 1999 entwickelt Microsoft (beteiligt an der MPEG-4-Entwicklung) einen Videocodec namens **MS-MPEG4** auf Grundlage einiger früher MPEG-4-Entwürfe
  - Qualität für damalige Verhältnisse bahnbrechend!

- Nachteil: Der Codec verweigerte die Einbettung in das populäre AVI-Containerformat, um das Microsoft ASF-Format zu pushen

- der Hacker Gej modifiziert den Codec so, dass die Einbettung in AVI wieder möglich ist, und ändert dabei den Namen in **DivX ;-) 3.1**

- Trivia am Rande: DivX (ohne ;-) war einst der Name für eine »selbstzerstörende« DVD ...

... und XviD

- ab 2000 wurde eine legale, **OpenDivX** genannte MPEG-4-ASP-Implementation als Open-Source-Projekt begonnen

- OpenDivX starb bald, wurde aber in zwei getrennten Projekten weitergeführt:
  - **DivX 4.x - 6.x** (kommerzielles Produkt)
  - **XviD** (Open-Source-Projekt)

- seit 2002 existieren auch DivX-kompatible DVD-Player
  ... obwohl die überwiegende Masse an DivX-Material weltweit aus illegalen Kopien urheberrechtlich geschützer Kinofilme besteht ... ☺

MPEG-4 als Framework

- MPEG-4 ist nicht nur ein Videokompressionsverfahren – es ist ein komplettes Multimedia-Framework!
- Bild (=Szene) besteht aus Objekten, die über einen Szenegraphen angeordnet werden
- Szene kann Text, Standbilder (Wavelet-komprimiert) oder auch Videos (**Video Object Planes**, **VOPs**) enthalten
- sogar Interaktivität ist möglich

- VOPs können auch teilweise transparent sein
  - spezielle Anpassungen der DCT (Shape Adaptive DCT) und der Motion Compensation

- in der Praxis nicht genutzt – MPEG-4-Videos bestehen aus einer Szene mit einem einzigen nicht-transparenten Video

H.263-Quantisierung

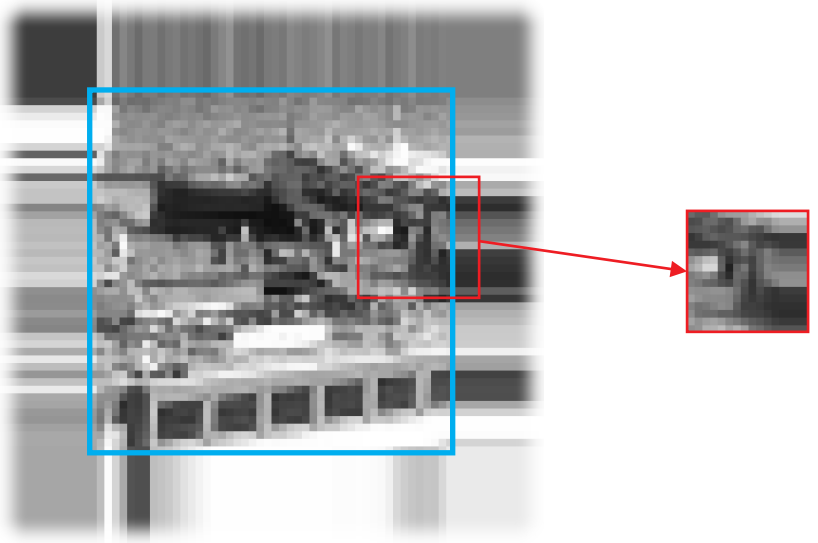
- neben klassischer MPEG-Quantisierung kann H.263-Quantisierung gewählt werden

- alle Koeffizienten werden durch den selben konstanten Wert dividiert

- Vorteil:
  - es werden ein paar Bits eingespart
- Nachteil:
  - das Bild erscheint u.U. etwas unschärfer

Unrestricted MVs

- Bewegungsvektoren können über Bildgrenzen hinaus zeigen
  - die Randpixel des Bildes werden dabei ggf. wiederholt
  - dadurch bessere Prediction bei Kameraschwenks



Quarter-Pel MC

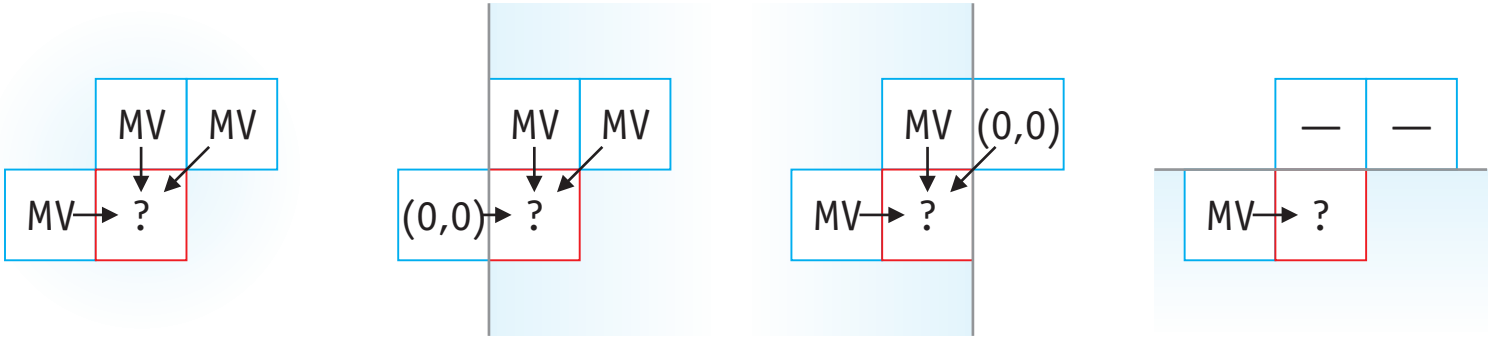
- Bewegungsvektoren können optional auf 1/4 Pixel genau sein (»**QPel**«)
  - ermöglicht präzisere MC für feine Bewegungen

- Wenn QPel aktiv ist, werden die 1/2-Pixel-Positionen für die Luminanz nicht mehr bilinear interpoliert, sondern mit einem 8-tap-Filter [-8, 24, -48, 160, 160, -48, 24, -8]
  - 1/4-Pixel werden nach wie vor bilinear aus den umliegenden Halbpixeln interpoliert

- Nachteil: deutlich höherer Rechenaufwand

Motion Vector Prediction

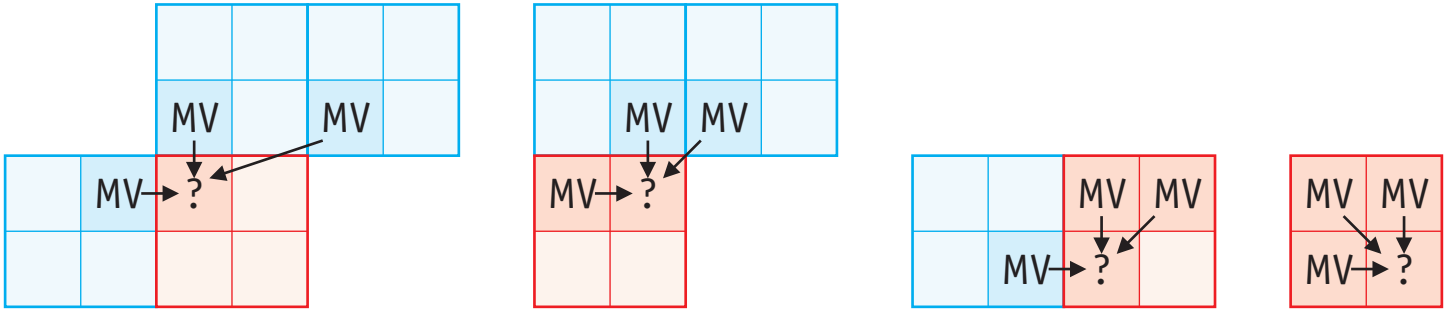
- MV-Prediction wurde gegenüber MPEG-2 deutlich verbessert:
  - aus den MVs der umliegenden drei Makroblöcke wird der Median-Wert für die x- und y-Koordinaten gebildet; die Abweichung wird dann codiert



- Vorteil:
  - MVs werden um einiges kleiner, da der Unterschied zur Vorhersage meist gering ausfällt

4 MVs pro Makroblock

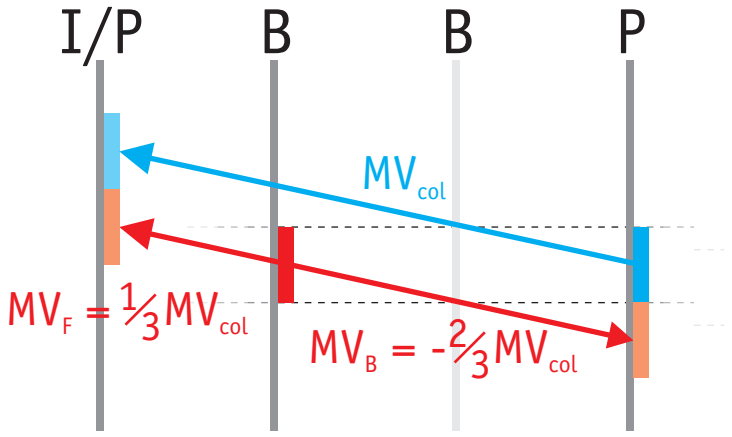
- optional können den vier Luma-Blöcken jedes Makroblocks eigene Bewegungsvektoren zugeordnet werden
- MV Prediction ändert sich entsprechend:



- Vorteil:
  - bessere Isolation räumlich eng begrenzter Bewegungen

Direct Mode Prediction

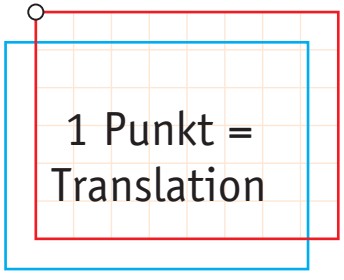
- die Bewegungsvektoren für B-Frames können automatisch aus dem folgenden P-Frame bestimmt werden
- der MV des gleichen Makroblocks im Referenzbild (**co-located MB**) wird dabei entsprechend dem Abstand der Bilder skaliert

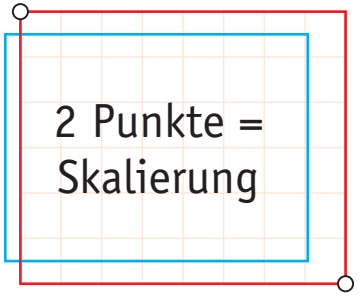


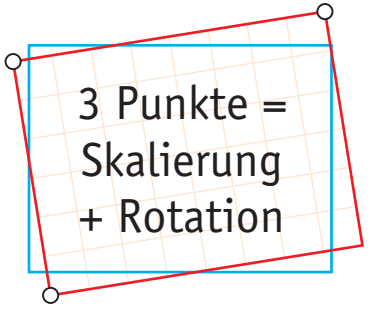
- derart generierte MVs sind meist nahe am Optimum
  - beste Resultate bei langsamen, großflächigen Bewegungen
  - Ergebnis: geringe MV-Differenz, höhere Codiereffizienz

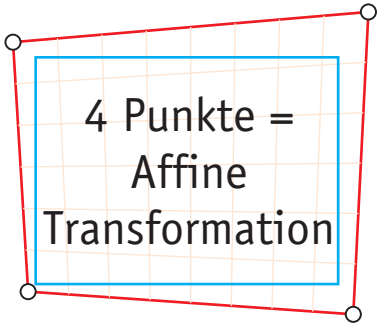
Global Motion Compensation

- optional zur Standard-Bewegungskompensation kann eine globale Bildtransformation angegeben werden
- Parameter dieser Transformation implizit durch Fluchtpunkte (**Warp Points**) gegeben

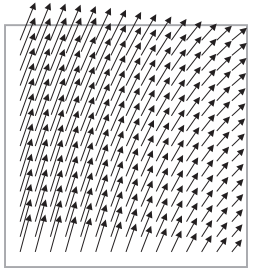








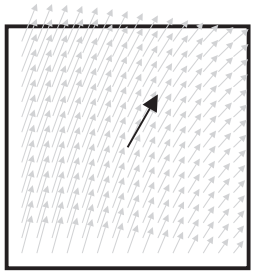
- GMC belegt Bewegungsvektoren vor
  - dabei erhält jedes **Pixel**, nicht nur jeder (Makro)-Block, einen eigenen MV



Global Motion Compensation

- die Entscheidung, ob GMC verwendet werden soll, wird für jeden Makroblock einzeln getroffen

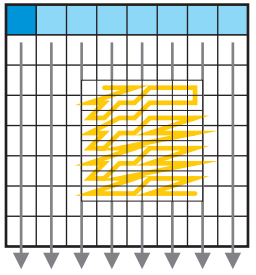
- Problem: ein nicht-GMC-Makroblock neben einem GMC-Makroblock benötigt für die MV Prediction einen MV für den ganzen Block
  - in diesem Fall werden alle Pixel-MVs über den ganzen Block gemittelt



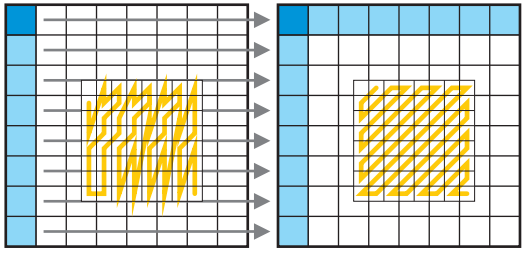
- Vorteil:
  - auch komplexe globale Bewegungen, Zooms usw. können effizient codiert werden
- Nachteil:
  - enorm hoher Berechnungsaufwand

AC Prediction

- in I- und P-Frames können in jedem Block die erste Zeile **oder** die erste Spalte von DCT-Koeffizienten aus dem darüberliegenden oder links benachbarten Block übernommen werden



- die Differenzen der vorhergesagten zu den tatsächlichen Koeffizienten werden schwächer quantisiert



- entsprechend der Vorhersagerichtung wird eine Auslesesequenz ausgewählt

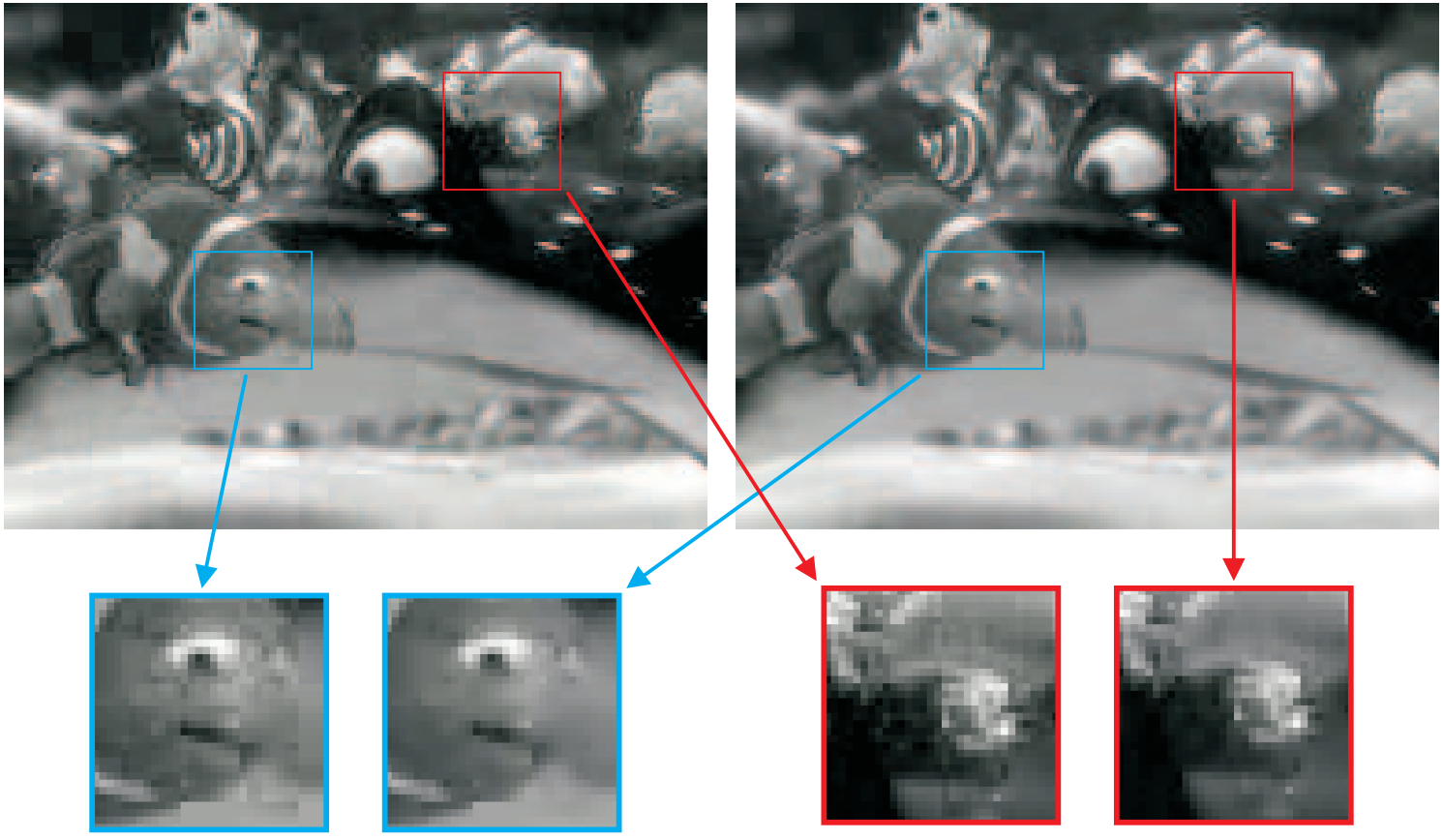
- Vorteil:
  – effiziente Codierung von vertikalen und horizontalen Kanten

Postprocessing

- mit allen vorgestellten Methoden ist MPEG-4 bei TV-üblichen Bildgrößen und Datenraten »nur« ca. doppelt so effizient wie MPEG-2
- enormer Qualitätsschub durch **Postprocessing**
  = Filter, die auf dem decodierten Bild ausgeführt werden
  – theoretisch auch mit MPEG-1 und -2 möglich, aber erst mit MPEG-4 populär geworden
- zwei Komponenten:
  – **Deblocking** = Detektion und Weichzeichnung von Blockartefakten
  – **Deringing** = Detektion und Entfernung von hochfrequenten Artefakten (»Moskitos«)
- die meisten Softwaredecoder unterstützen Postprocessing

Postprocessing

- Ein Beispiel ohne und mit Postprocessing:



Profiles & Levels

- MPEG-4 Visual kennt 19 Profile – die wichtigsten sind:
  Simple = nur P-Frames
  Advanced Simple = Simple + B-Frames, QPel, GMC, Frame Interlaced Pictures
  Simple Scalable = Simple + Spatial Scalability
  Core = beliebig geformte Objekte, Temporal Scal.
  Main = Transparenz, Interlacing, Sprites
- es wird fast nur das (Advanced) Simple Profile verwendet

- jedes Profil hat 7 Levels – die wichtigsten sind:
  Level 0/1 = max. 176x144/30fps, 128 kbps
  Level 2/3 = max. 352x288/30fps, 384/768 kbps
  Level 4 = max. 352x576/30fps, 3 Mbps
  Level 5 = max. 720x576/30fps, 8 Mbps

H.264 / AVC

- vom Joint Video Team (JVT) der ISO zunächst unter dem Namen H.26L entwickelt, 2003 standardisiert

- ebenfalls in MPEG-4-Familie aufgenommen als **MPEG-4 Part 10: Advanced Video Coding** (AVC)

- zuerst in Software realisiert, Hardware erst seit Mitte 2004

- deckt alle Anwendungsbereiche ab
  - Pflichtcodec für Blu-ray Discs
  - für HDTV-Übertragungen verwendet
  - Standardcodec für mobile Applikationen (Smartphones)
  - Standardcodec für Internet-Streaming-Video

Integertransformation

- Problem: 8x8-DCT von MPEG ist ein Floating-Point-Verfahren
  - Implementation aus Performancegründen trotzdem meist in Fixed Point
  - geringe implementationsspezifische Abweichungen zwischen Encoder und Decoder wurden toleriert
  - strenge Vorgaben zur Rechengenauigkeit, die nicht von jeder Implementation eingehalten wurden
    - führt zur Fehlerfortpflanzung durch P-Frames!

- Lösung in H.264: DCT-ähnliche Integertransformationen
- Vorteile:
  - Berechnung in 16-Bit-Integerarithmetik möglich
  - exakte Rekonstruktion – keine Rundungsfehler

4x4-Transformation

- Standard-Transformationsgröße in H.264 ist 4x4

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

Transformationsmatrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

Rücktransformation

- notwendige Skalierungen der Koeffizienten sind in den Quantisierungsschritt integriert
- Vorteil: weniger anfällig für »Moskito«-Artefakte
- für 16x16-Intra-MB werden die DC-Koeffizienten der 4x4 Subblöcke zudem einer ähnlichen Transformation (**Hadamard-Transformation**) unterzogen
- im High Profile auch 8x8-Transformationen möglich
- optional **Scaling Lists** für MPEG-artige Matrixquantisierung

Entropiecodierung

CAVLC (Context-Adaptive Variable-Length Coding)

- effizientere Codierung der Koeffizienten durch Übertragung
  - der Anzahl von Koeffizienten ungleich Null **(TotalCoeff)**
  - der Anzahl von Nullen im Scan **(TotalZeros)**
  - der Anzahl von Koeffizienten am Ende des Scans mit dem absoluten Betrag 1 – von ihnen müssen später nur noch die Vorzeichen codiert werden **(TrailingOnes)**
- Codierung der Koeffizienten in Rückwärtsreihenfolge
  - typischerweise monotoner Anstieg der Beträge
  - Codierung ist entsprechend angepasst
- Auswahl der VLC-Tabelle für TotalCoeff und TrailingOnes ist abhängig von den TotalCoeff-Werten der umliegenden Blöcke (»kontextadaptive« Codierung)

CABAC

CABAC (Context Adaptive Binary Arithmetic Coding)

- optional anstelle von CAVLC
- leistungsfähige kontextsensitive Arithmetische Codierung
- Was ist Arithmetische Codierung (AC)?
  - bei konventioneller Huffman-Entropiecodierung kann ein Wert nur eine ganze Anzahl von Bits beanspruchen
  - bei den meisten Häufigkeitsverteilungen wäre eine feinere Abstufung jedoch wünschenswert – AC leistet genau das
- Binäre Arithmetische Codierung
  - man codiert nur Nullen und Einsen arithmetisch (»**bins**«)
  - die Wahrscheinlichkeit für 0/1 kann beliebig sein, es wird immer ein optimaler Code generiert
- Kontextadaptiv
  - die Wahrscheinlichkeiten werden laufend nachjustiert

Exp-Golomb-Codes

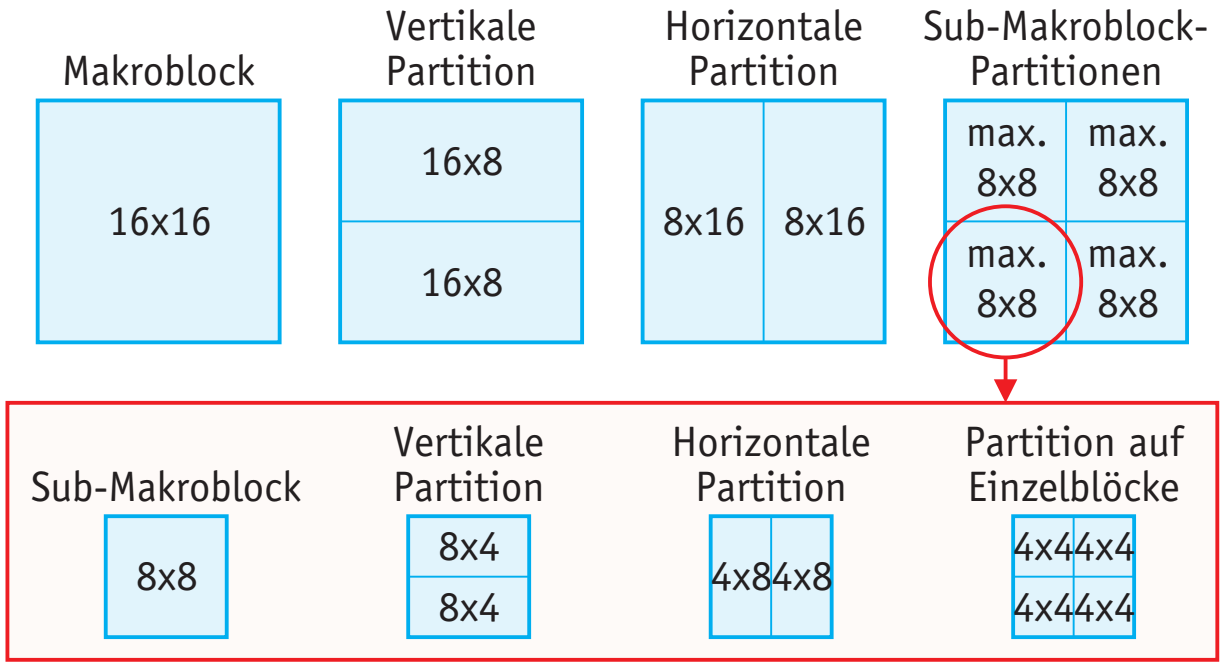
UVLC (Universal Variable-Length Code): **Exp-Golomb-Code**

- verwendet für die meisten Syntaxelemente außer Koeffizienten
- kleine (häufigere) Werte erhalten kürzere Codes
- n binäre '0' + 1 binäre '1' + n Bits Offset = 2^n - 1 + Offset
- Beispiele:

1	= 0
010	= 1
011	= 2
00100	= 3
00000101011	= 42

Macroblock Partitioning

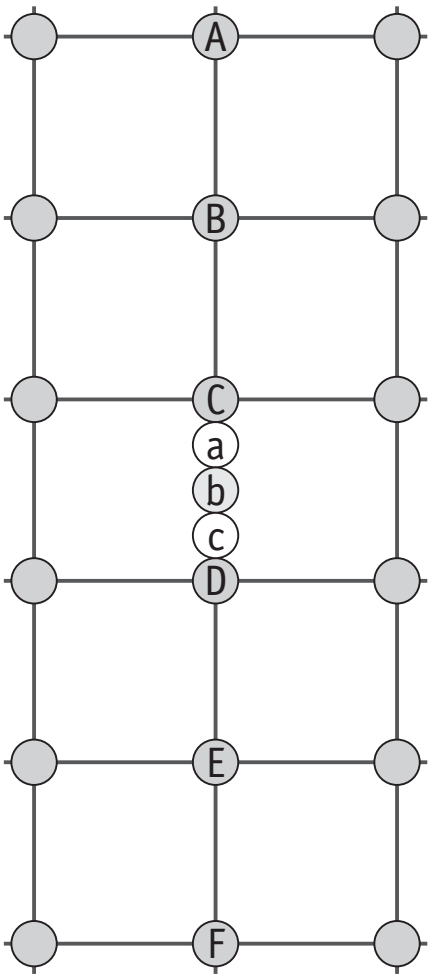
- Makroblöcke können in **Macroblock Partitions** und sogar **Sub-Macroblock Partitions** zerlegt werden



- jede Partition hat eigene MVs
  - präzise Isolation lokaler Bewegungen möglich

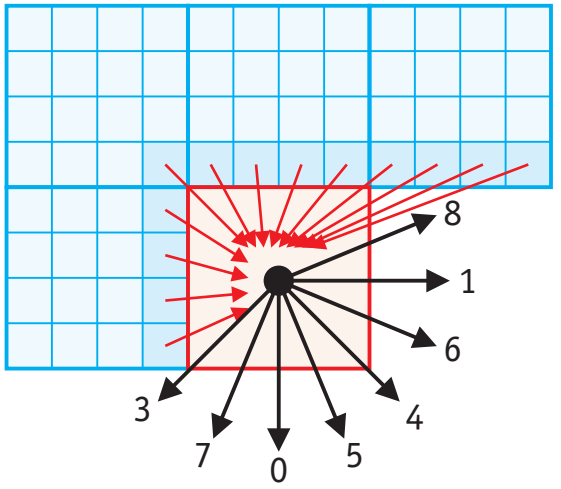
Motion Compensation

- Quarter-Pel-MC ist nun obligatorisch
- Interpolationsfilter für Halb-Pixel-Positionen ist nun ein 6-tap-Filter:
  $b = (A - 5B + 20C + 20D - 5E + F) / 32$
- Viertel-Pixel-Positionen werden noch bilinear ermittelt:
  $a = (C + b) / 2$
  $c = (b + D) / 2$
- Chroma-MC: bilinear, auf 1/8 Pixel genau
- keine Global Motion Compensation



Intra Prediction

- auch innerhalb von I-Frames wird Prediction verwendet
  - Pixelwerte eines Makroblocks oder Transformationsblocks werden aus den umliegenden Pixeln vorhergesagt und nur die Differenz codiert
  - macht AC Prediction überflüssig



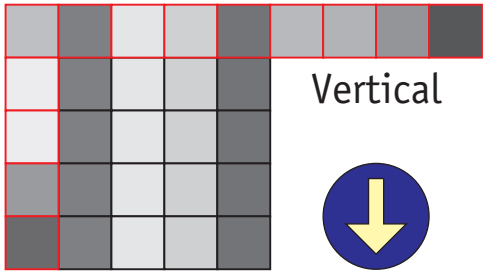
4x4 und 8x8 Intra Prediction

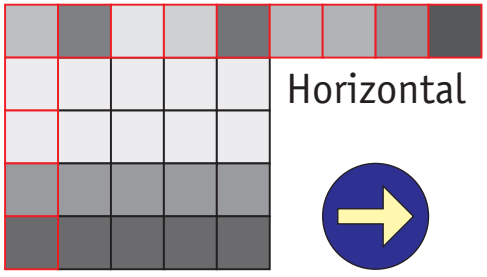
- Vorhersage erfolgt durch Interpolation der Pixel in einer von neun Richtungen

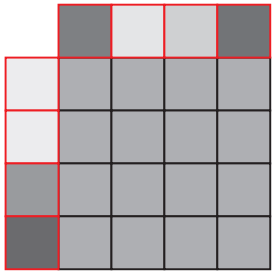
16x16 Intra und **Chroma Prediction**

- dito, aber nur vier Richtungen: horizontal, vertikal, DC und Plane (erzeugt einen Farbverlauf)

Intra_4x4 Prediction Modes

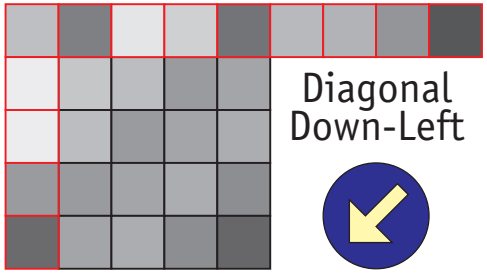


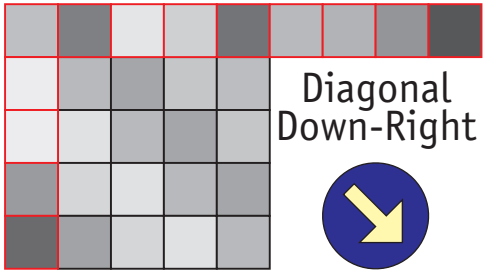


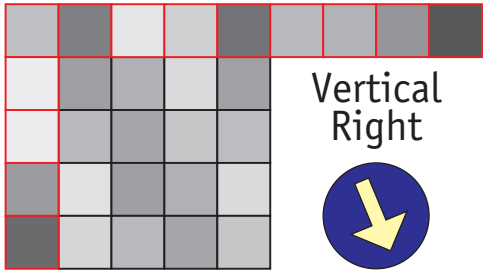


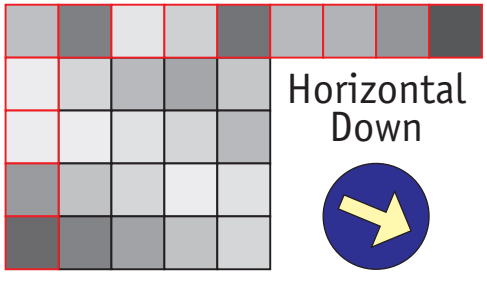
DC

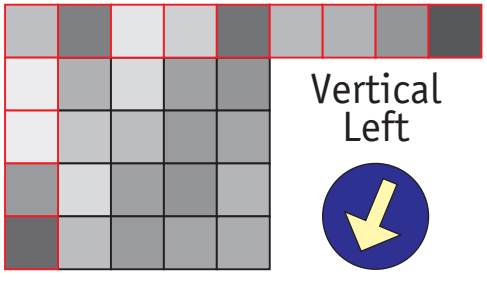
= Durch-schnitt

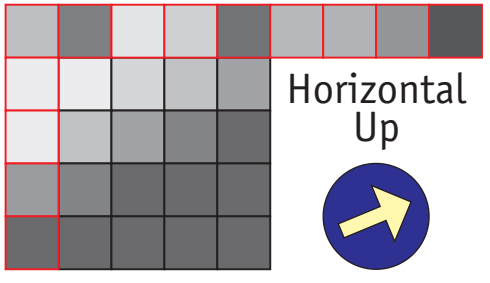






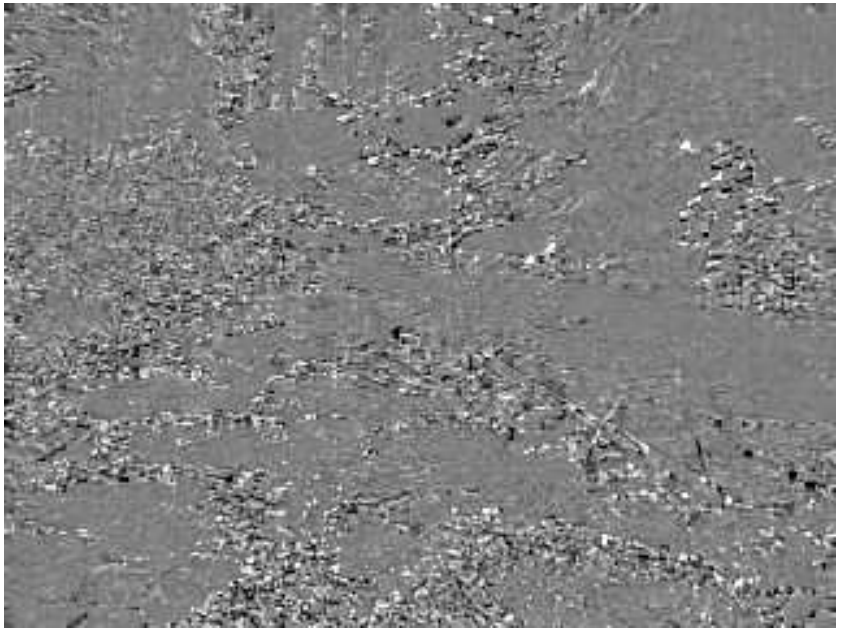






Beispiel für Intra Prediction

Originalbild

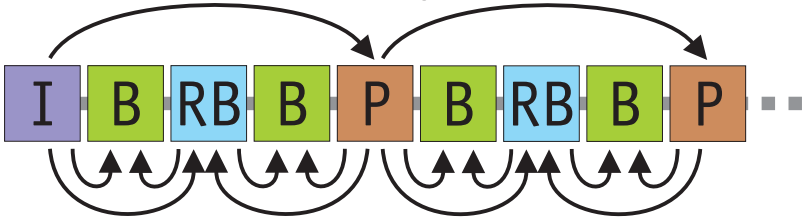


Prädiktionsfehler (»Residual«) nach Intra Prediction

Long-Term Prediction

- es können nicht nur 2, sondern bis zu 16 Frames als Referenz herangezogen werden
- Bilder können als **Long-Term Reference Pictures** im Bildpuffer »festgehalten« werden
- Vorteil:
  - bessere Codierung periodischer Bewegungen
- Nachteil:
  - höherer Speicherbedarf, höhere Komplexität

- außerdem: B-Bilder können für Prediction benutzt werden
  - Reference B Pictures, »B-Pyramide«



Weighted Prediction

Implicit Weighted Prediction

- mischt im Direct Mode auch die Bilddaten entsprechend der Entfernungen zu den Referenzbildern

Explicit Weighted Prediction

- jedem Referenzbild wird ein Gewicht (**weight**) und ein **Offset** zugeordnet
  - Prediction = Referenzpixel * Weight + Offset
  - ermöglicht beliebige Mischungen mehrerer Referenzbilder
  - ermöglicht Helligkeits- und Kontrastanpassungen
  - nützlich z.B. für Fade-Effekte (Fade-Out, Fade-In, Crossfade)

In-Loop Deblocking Filter

- H.264 verläßt sich nicht nur auf externes Postprocessing
- Deblocking-Filter ist in den En- und Decoder integriert:
  In-Loop Deblocking Filter (gelegentlich auch »Loop Filter«)
- Frames werden direkt nach der Decodierung gefiltert
  - Interframe-Vorhersagen beziehen sich auf bereits gefilterte Frames!
- Ergebnis: deutlich gesteigerte Bildqualität

Beispiel mit und ohne Deblocking Filter nach 62 P-Frames

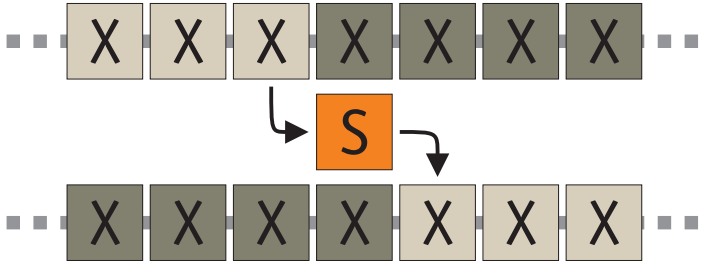
Slices

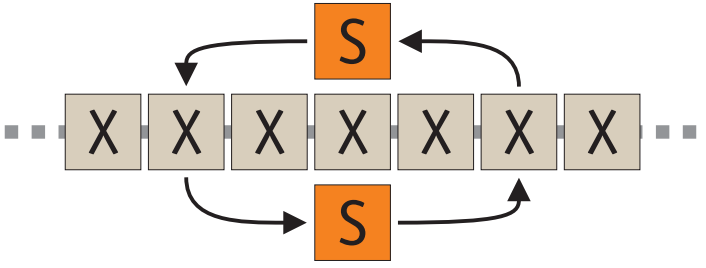
- Slices eines Bildes können unterschiedlichen Typs sein
  - man spricht von I/P/B-**Slices**, nicht -Frames
  - meist nur gleichartige Slices in einem Bild, aber Mischen verschiedener Slice-Typen ist prinzipiell möglich
- Slices werden in **NAL-Units** (**Network Abstraction Layer**) transportiert

- Optional: **Data Partitioning**
  - Daten eines Slices werden auf 3 NAL-Units aufgeteilt
- Optional: **Arbitrary Slice Order (ASO)**
  - Slices eines Bildes werden in beliebiger Reihefolge codiert
- Optional: **Flexible Macroblock Order (FMO)**
  - Makroblöcke eines Slices in beliebiger Reihenfolge

Switching Slices

- Switching Slices (SI- oder SP-Slices) ermöglichen Übergänge zwischen zwei Datenströmen A und B, ohne auf den nächsten I-Frame warten zu müssen
- SI/SP-Slices können aus Datenstrom A rekonstruiert werden und sind gleichzeitig gültige Referenzen für Datenstrom B
- kann auch zur Realisierung von Bildsuchlauf o.ä. verwendet werden
- dennoch: exotisches Feature, kaum genutzt





Interlaced Encoding

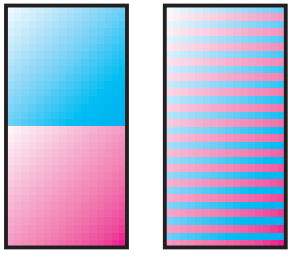
PAFF: Picture Adaptive Frame/Field Coding

- Unterscheidung in Frame und Field Pictures auf Frame-Ebene (analog MPEG-2)

MBAFF: Macroblock Adaptive Frame/Field Coding

- entspricht etwa den Frame Interlace Pictures von MPEG-2
- Besonderheit: paarweise Codierung von Makroblöcken
  - es werden zwei Makroblockzeilen gemeinsam codiert

Frame-Makroblockpaar
zwei Frame-Makroblöcke übereinander



Field-Makroblockpaar
Makroblöcke fieldweise ineinander »verwoben«

Lossless Encoding

- H.264 bietet zwei Möglichkeiten für verlustfreie Videocodierung

unkomprimierte Makroblöcke

- eigener Makroblocktyp **I_PCM**
- die 384 Samples des Makroblocks werden unkomprimiert abgespeichert

verlustfreie Entropiecodierung

- nur im höchsten Profil (High 4:4:4) verfügbar
- Transformationsschritt wird komplett ausgelassen (»Transform Bypass Mode«)
- Differenz zur Prediction wird direkt der Entropiecodierung unterzogen

Profiles

- H.264 definiert ca. 13 Profile – die wesentlichen:
  - **Baseline** = keine B-Slices, keine Weighted Prediction, kein CABAC, keine SP/SI-Slices, aber ASO/FMO
  - **Extended** = Baseline + B/SI/SP-Slices, Data Partitioning
  - **Main** = Baseline + B-Slices, Weighted Prediction, CABAC, Interlacing, kein ASO/FMO
  - **High** = Main + 8x8 + Scaling Lists
  - **High 10** = High + 10 Bit pro Sample Farbtiefe
  - **High 4:2:2** = High 10 + Chroma nur horizontal reduziert
  - **High 4:4:4** = High 4:2:2 + Chroma in voller Auflösung
- Baseline Profile sehr verbreitet, aber ASO/FMO nicht
  - 2009 Einführung des **Constrained Baseline Profile**
  - praktisch jeglicher Content im BP ist in Wirklichkeit CBP

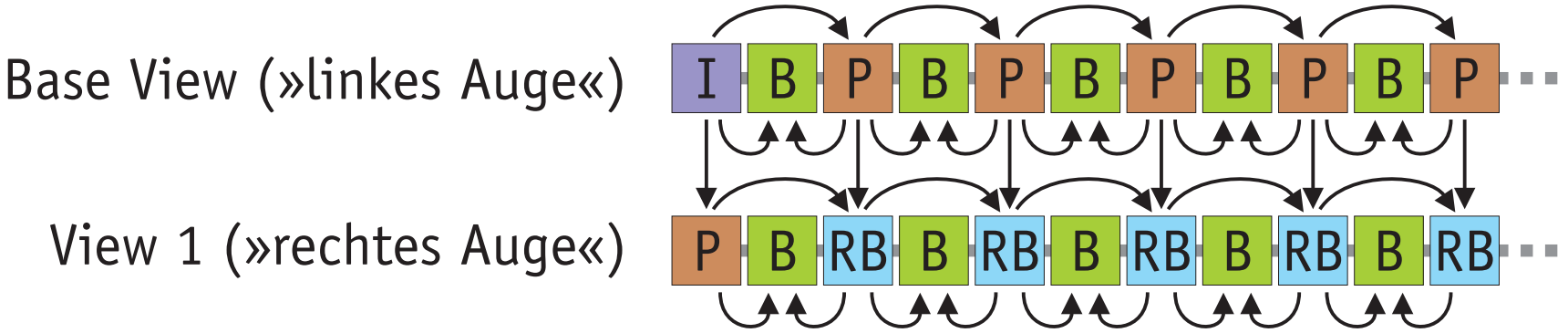
Levels

- insgesamt 17 Levels, die wichtigsten davon:

Level 1	176x144/15 fps, 64 kbps
Level 1.1	176x144/30 fps, 192 kbps
Level 1.3	352x288/30 fps, 768 kbps
Level 2.0	352x288/30 fps, 2 Mbps
Level 3	720x576/25 fps, 720x480/30 fps, 10 Mbps
Level 3.1	1280x720/30 fps, 720x576/60 fps, 14 Mbps
Level 3.2	1280x720/60 fps, 20 Mbps
Level 4	1920x1080/30 fps, 20 Mbps
Level 4.1	1920x1080/30 fps, 50 Mbps (Blu-ray)
Level 4.2	1920x1080/60 fps, 50 Mbps
Level 5.1	4096x2304/24 fps, 240 Mbps
Level 5.2	4096x2304/60 fps, 240 Mbps

SVC und MVC

- H.264 SVC = **Scalable Video Coding**
  - Scalable-Modi ähnlich MPEG-2
- H.264 MVC = **Multi-View Video Coding** (2009)
  - spezieller Modus für 3D-Content, benutzt bei Blu-ray Discs
  - codiert bis zu 1024 Ansichten (Views) der gleichen Szene
  - nutzt Ähnlichkeiten der Ansichten zur besseren Codierung
  - abwärtskompatibel: Basisansicht = normaler H.264-Strom
    - kann von jedem H.264-Decoder decodiert werden
  - zusätzliche Ansichten für Nicht-MVC-Decoder »unsichtbar«



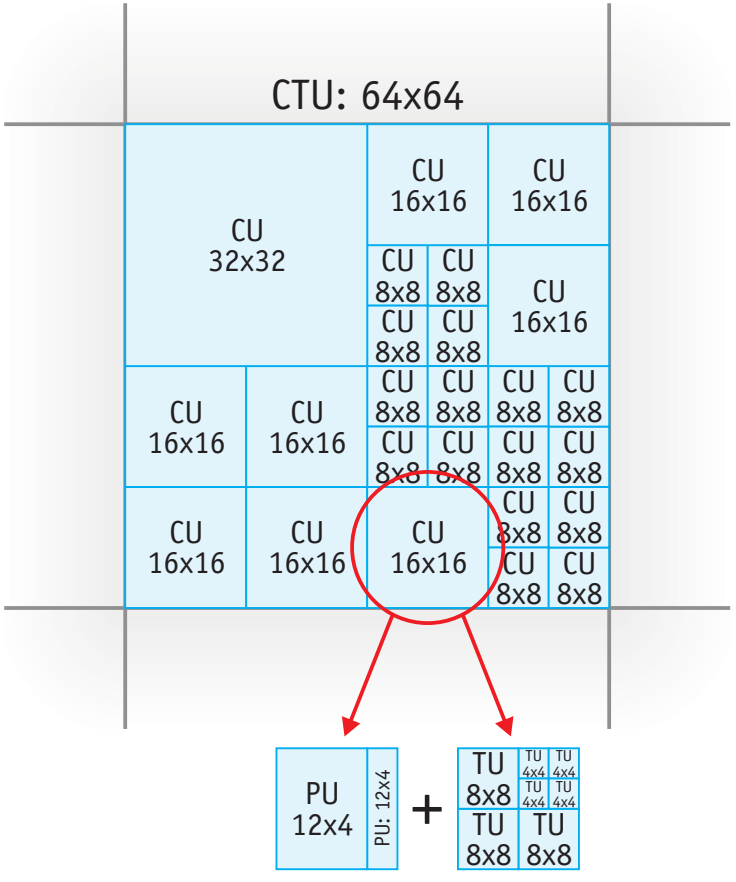
H.265 / HEVC

- vom »Joint Collaborative Team for Video Coding« (JCT-VC) der ISO entwickelt, 2013 fertiggestellt
- Arbeitstitel: **High Efficiency Video Coding** (HEVC)
- von der ISO aufgenommen als ISO 23008-2 **MPEG-H Part 2**
- von der ITU-T aufgenomme als **H.265**

- etwa doppelte Effizienz gegenüber H.264
- in Hinblick auf UHDTV (4K, 8K) entwickelt, jedoch auflösungsunabhängig benutzbar
- Gesamtkomplexität deutlich höher als bei H.264
  - einige Komponenten aber sogar vereinfacht

Coding Tree

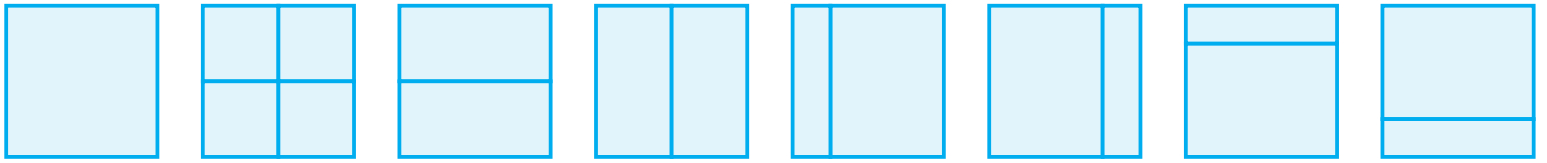
- keine 16x16-Makroblöcke mehr
- stattdessen **CTU**s (Coding Tree Units)
  - 16x16, 32x32 oder 64x64
- CTUs baumartig unterteilt in **CU**s (Coding Units)
  - 64x64 ... 8x8
- CUs unterteilt in
  - **PU**s (Prediction Units)
  - **TU**s (Transform Units)
  - PU-Unterteilung ist weitgehend unabhängig von TU-Unterteilung



PU- und TU-Splits

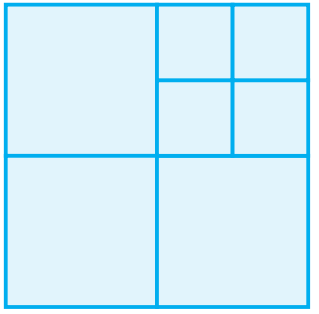
PU-Unterteilungen

- eine Unterteilung pro CU in 1, 2 oder 4 PUs
- minimale PU-Größe: 4x8 oder 8x4 (8x8 bei bidir. Prädiktion)
- »asymmetrische« Unterteilungen (¼:¾) möglich

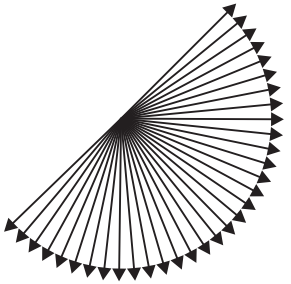


TU-Unterteilungen

- hierarchische Unterteilung, ähnlich CTU-zu-CU-Unterteilungen
- minimale TU-Größe: 4x4
- bei Intra darf eine TU höchstens so groß sein wie die entsprechende PU



Intra Prediction



- 35 Modi: 33 direktionale + Planar + DC
  - generischer Algorithmus für direktionale Modi
  - Planar-Modus komplexer als bei H.264
- gleiche Modi für alle PU-Größen von 4x4 bis 64x64
- Einbeziehung aller Nachbarpixel (H.264: unten links ignoriert)

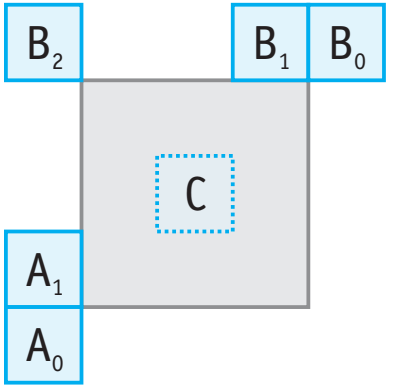
- Prediction des IP-Modus mit 3 MPMs (Most Probable Modes)
  - typischer Fall: 2 MPMs = Nachbarn, 3. MPM = Plane oder DC
  - wenn alle Nachbarn gleich und direktional:
    MPMs = Modus der Nachbarn und umliegende Modi
  - Encoder signalisiert, welchen MPM oder welchen der anderen 32 Modi er wählt
- Chroma-Modus aus Luma-Modus abgeleitet
  - 5 feste Mapping-Tabellen, Encoder wählt Tabelle aus

MV Prediction

- Zwei Modi für Motion Vector Prediction, wählbar auf CU-Ebene
  1. **AMVP** (Advanced Motion Vector Prediction):
     - Berechnung von 2 MV-Kandidaten
     - explizite Signalisierung des gewünschten Kandidaten
     - Codierung eines MV-Deltas
  2. **Merge Mode**
     - Berechnung von maximal 5 MV-Kandidaten
     - explizite Signalisierung des gewünschten Kandidaten
     - gewählter Kandidat wird direkt benutzt, kein MV-Delta
- Merge Mode kommt auch für »skipped« CUs zum Einsatz
  - d.h. auch für skipped CUs werden Daten übertragen

AMVP

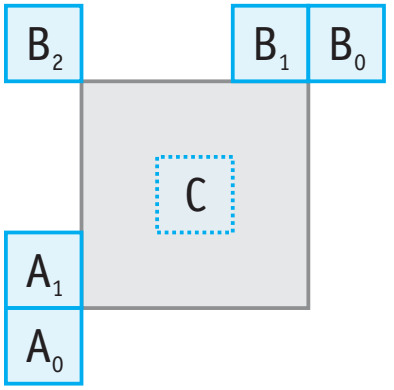
- Konstruktion einer Liste von Kandidaten, bestehend aus
  - einem MV der linken Nachbarn (A)
  - einem MV der oberen Nachbarn (B)
  - dem MV des Blocks im Vorgängerbild (C)
  - Nullvektoren
- nicht vorhandene Vektoren werden nicht in die Liste eingefügt
  - z.B. Bildrand, andere Slice, Intra-codiert
- Liste wird auf 2 Einträge gekürzt



- Generierung der Kandidaten A und B:
  - $A = A_0$ / $B = B_0$, falls verfügbar
  - falls nicht: $A = A_1$ / $B = B_1$, falls verfügbar
  - falls nicht: $B = B_2$, falls verfügbar

Merge Mode

- Konstruktion einer Liste wie folgt:
  - A_1, B_1, B_0, A_0, B_2, C
  - doppelte Kandidaten werden aussortiert (nur teilweise, um Vergleiche zu sparen)
  - in B-Slices: Hinzufügen von »Combined Bi-predictive Candidates«
    - wenn sich in der Liste sowohl Forward- als auch Backward-Vektoren befinden, werden diese zu bidirektionalen Vektoren zusammengesetzt



  - Hinzufügen von Nullvektoren für alle vorkommenden Referenzbilder
  - Reduktion der Liste auf 5 Elemente (oder weniger, wenn der Encoder so konfiguriert wurde)

Transformation

- Haupt-DCT als 32x32-Matrix definiert
- kleinere Transformationen benutzen Teilmatrizen davon

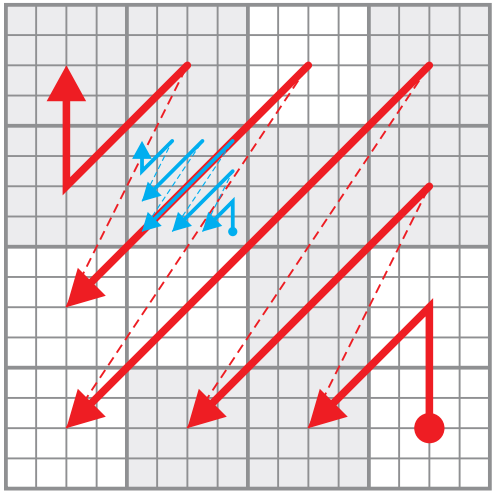
64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
90	90	88	85	82	78	73	67	61	54	46	38	31	22	13	4	-4	-13	-22	-31	-38	-46	-54	-61	-67	-73	-78	-82	-85	-88	-90	-90
90	87	80	70	57	43	25	9	-9	-25	-43	-57	-70	-80	-87	-90	-90	-87	-80	-70	-57	-43	-25	-9	9	25	43	57	70	80	87	90
90	82	67	46	22	-4	-31	-54	-73	-85	-90	-88	-78	-61	-38	-13	13	38	61	78	88	90	85	73	54	31	4	-22	-46	-67	-82	-90
89	75	50	18	-18	-50	-75	-89	-89	-75	-50	-18	18	50	75	89	89	75	50	18	-18	-50	-75	-89	-89	-75	-50	-18	18	50	75	89
88	67	31	-13	-54	-82	-90	-78	-46	-4	38	73	90	85	61	22	-22	-61	-85	-90	-73	-38	4	46	78	90	82	54	13	-31	-67	-88
87	57	9	-43	-80	-90	-70	-25	25	70	90	80	43	-9	-57	-87	-87	-57	-9	43	80	90	70	25	-25	-70	-90	-80	-43	9	57	87
85	46	-13	-67	-90	-73	-22	38	82	88	54	-4	-61	-90	-78	-31	31	78	90	61	4	-54	-88	-82	-38	22	73	90	67	13	-46	-85
83	36	-36	-83	-83	-36	36	83	83	36	-36	-83	-83	-36	36	83	83	36	-36	-83	-83	-36	36	83	83	36	-36	-83	-83	-36	36	83
82	22	-54	-90	-61	13	78	85	31	-46	-90	-67	4	73	88	38	-38	-88	-73	-4	67	90	46	-31	-85	-78	-13	61	90	54	-22	-82
80	9	-70	-87	-25	57	90	43	-43	-90	-57	25	87	70	-9	-80	-80	-9	70	87	25	-57	-90	-43	43	90	57	-25	-87	-70	9	80
78	-4	-82	-73	13	85	67	-22	-88	-61	31	90	54	-38	-90	-46	46	90	38	-54	-90	-31	61	88	22	-67	-85	-13	73	82	4	-78
75	-18	-89	-50	50	89	18	-75	-75	18	89	50	-50	-89	-18	75	75	-18	-89	-50	50	89	18	-75	-75	18	89	50	-50	-89	-18	75
73	-31	-90	-22	78	67	-38	-90	-13	82	61	-46	-88	-4	85	54	-54	-85	4	88	46	-61	-82	13	90	38	-67	-78	22	90	31	-73
70	-43	-87	9	90	25	-80	-57	57	80	-25	-90	-9	87	43	-70	-70	43	87	-9	-90	-25	80	57	-57	-80	25	90	9	-87	-43	70
67	-54	-78	38	85	-22	-90	4	90	13	-88	-31	82	46	-73	-61	61	73	-46	-82	31	88	-13	-90	-4	90	22	-85	-38	78	54	-67
64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64	64	-64	-64	64
61	-73	-46	82	31	-88	-13	90	-4	-90	22	85	-38	-78	54	67	-67	-54	78	38	-85	-22	90	4	-90	13	88	-31	-82	46	73	-61
57	-80	-25	90	-9	-87	43	70	-70	-43	87	9	-90	25	80	-57	-57	80	25	-90	9	87	-43	-70	70	43	-87	-9	90	-25	-80	57
54	-85	-4	88	-46	-61	82	13	-90	38	67	-78	-22	90	-31	-73	73	31	-90	22	78	-67	-38	90	-13	-82	61	46	-88	4	85	-54
50	-89	18	75	-75	-18	89	-50	-50	89	-18	-75	75	18	-89	50	50	-89	18	75	-75	-18	89	-50	-50	89	-18	-75	75	18	-89	50
46	-90	38	54	-90	31	61	-88	22	67	-85	13	73	-82	4	78	-78	-4	82	-73	-13	85	-67	-22	88	-61	-31	90	-54	-38	90	-46
43	-90	57	25	-87	70	9	-80	80	-9	-70	87	-25	-57	90	-43	-43	90	-57	-25	87	-70	-9	80	-80	9	70	-87	25	57	-90	43
38	-88	73	-4	-67	90	-46	-31	85	-78	13	61	-90	54	22	-82	82	-22	-54	90	-61	-13	78	-85	31	46	-90	67	-4	-73	88	-38
36	-83	83	-36	-36	83	-83	36	36	-83	83	-36	-36	83	-83	36	36	-83	83	-36	-36	83	-83	36	36	-83	83	-36	-36	83	-83	36
31	-78	90	-61	4	54	-88	82	-38	-22	73	-90	67	-13	-46	85	-85	46	13	-67	90	-73	22	38	-82	88	-54	-4	61	-90	78	-31
25	-70	90	-80	43	9	-57	87	-87	57	-9	-43	80	-90	70	-25	-25	70	-90	80	-43	-9	57	-87	87	-57	9	43	-80	90	-70	25
22	-61	85	-90	73	-38	-4	46	-78	90	-82	54	-13	-31	67	-88	88	-67	31	13	-54	82	-90	78	-46	4	38	-73	90	-85	61	-22
18	-50	75	-89	89	-75	50	-18	-18	50	-75	89	-89	75	-50	18	18	-50	75	-89	89	-75	50	-18	-18	50	-75	89	-89	75	-50	18
13	-38	61	-78	88	-90	85	-73	54	-31	4	22	-46	67	-82	90	-90	82	-67	46	-22	-4	31	-54	73	-85	90	-88	78	-61	38	-13
9	-25	43	-57	70	-80	87	-90	90	-87	80	-70	57	-43	25	-9	-9	25	-43	57	-70	80	-87	90	-90	87	-80	70	-57	43	-25	9
4	-13	22	-31	38	-46	54	-61	67	-73	78	-82	85	-88	90	-90	90	-90	88	-85	82	-78	73	-67	61	-54	46	-38	31	-22	13	-4

- zusätzlich 4x4-DST (Diskrete Sinustransformation) bei bestimmten Intra-Prediction-Modi

29	55	74	84
74	74	0	-74
84	-29	-74	55
55	-84	74	-29

Koeffizientencodierung

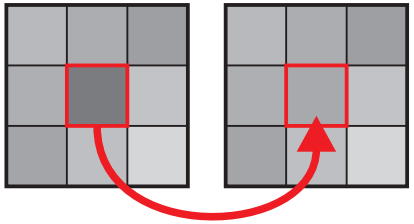
- Codierung der Koeffizienten stets in 4x4-Subblöcken
  - höhere Effizienz, wenn sehr viele Nullen codiert werden: ganze Blöcke können ausgeschlossen werden
- Sonderbehandlung nicht nur von Koeffizienten mit den Werten +/– 1, sondern auch +/– 2
- Scan-Reihenfolge: meistens »Subdiagonal« (Zick-Zack-ähnlich)
  - bei Intra 4x4 und 8x8 auch Horizontal und Vertikal möglich
- optional **Sign Data Hiding**
  - Vorzeichen des letzten Koeffizienten eines Subblocks wird nicht übertragen
  - indirekte Ableitung aus der Summe der Koeffizienten
  - Encoder muss ggf. Koeffizienten modifizieren



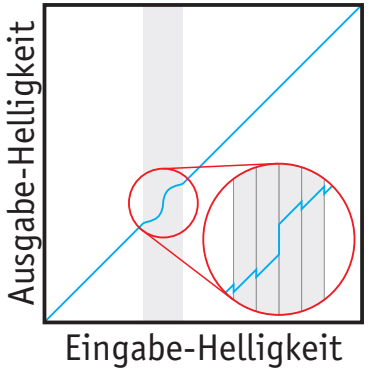
Deblocking und SAO

- Deblocking-Filter ähnlich H.264, aber leicht vereinfacht

- zweites, zusätzliches Filter: **SAO** (Sample Adaptive Offset)
  - notwendig, weil größere Transformationen Rundungsfehler und Ringing-Artefakte mit sich bringen
- angewendet auf alle Pixel, zwei Modi:
  - Modifikation aufgrund der Umgebungspixel (Edge Mode)
  - Modifikation aufgrund der Pixelwerte selbst (Band Mode)



Beispiel Edge Type SAO:
Pixel deutlich dunkler als Umgebung, wird aufgehellt



Band Type SAO:
Lokale Änderung der Helligkeitswerte in einem kleinen Bereich des Histogramms (4 von 32 »Bändern«)

- Parameter auf CTU-Ebene signalisiert

Verschiedenes

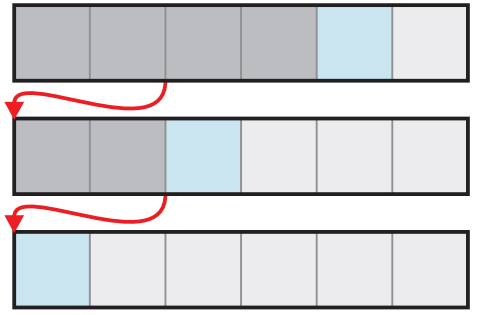
- Referenzbildverwaltung gegenüber H.264 deutlich vereinfacht
  - explizite Signalisierung aller Referenzbilder in jedem Frame
- Luma-MVs: Quarter-Pel, Interpolation mit zwei Filtern:
  - 7-tap für Half-Pel-Positionen
  - 8-tap symmetrisch für Quarter-Pel-Positionen
- Chroma-MVs: 1/8-Pel, Interpolation mit vier 4-tap-Filtern
- kein CAVLC mehr, (leicht modifiziertes) CABAC ist Pflicht
- Dependent Slices: Slices müssen nicht mehr vollständigen Header enthalten, um Overhead zu minimieren
- keinerlei Interlaced-Support mehr (kein PAFF, kein MBAFF)

Tiles und Wavefronts

- verschiedene Verfahren, um Parallelisierung zu erleichtern
- Bild kann optional in rechteckige Bereiche (**Tiles**) eingeteilt werden
  - Tile = rechteckige Gruppe aus CTUs
  - keine Prediction über Tile-Grenzen (aber: Deblocking ist möglich

1	2	7	10	11	12
3	4	8	13	14	15
5	6	9	16	17	18
19	20	21	22	23	24

- Alternativ **Wavefronts** zur parallelen En-/Decodierung von CTU-Zeilen
  - »Wavefront Parallel Processing« (WPP)
  - Zeilen setzen um 2 CTUs versetzt ein



- Slices über Tile-/Wavefront-Grenzen hinweg sind möglich
  - aber: Tabelle im Slice-Header ermöglicht direktes »Einspringen« in bestimmte Tiles/Wavefronts

Profile, Levels und Tiers

- derzit nur 3 Profile: **Main Profile**, Main 10 Profile (10 Bit pro Pixel), Main Still Picture Profile
- 13 Levels, Benennung an H.264 angelehnt
- einige Levels zusätzlich in je zwei **Tiers** unterteilt
  - Main Tier und High Tier, nach Bitrate und Speichergröße
- wichtigste Levels (Bitrate jeweils für beide Tiers):

Level 3	720x576/30 fps, 6 Mbps
Level 3.1	720x576/60 fps, 1280x720/30 fps, 10 Mbps
Level 4	1280x720/60 fps, 1920x1080/30 fps, 12/30 Mbps
Level 4.1	1920x1080/60 fps, 20/50 Mbps
Level 5	4096x2160/30 fps, 25/100 Mbps
Level 5.1	4096x2160/60 fps, 40/160 Mbps
Level 6	8192x4320/30fps, 60/240 Mbps
Level 6.1	8192x4320/60fps, 120/480 Mbps

Sonstige Verfahren

- die MPEG-Standards sind nicht die einzigen Videokompressionsverfahren

- andere Standards basieren jedoch auf den selben grundlegenden Konzepten

- Unterschiede in
  - Bitstromaufbau und -Syntax
  - Art und Weise der Prediction
  - verfügbaren **Coding Tools** (Features)

H.26x

- Videocodecfamilie der ITU-T Study Group 16 (ursprünglich »Video Coding Experts Group«, VCEG)
  - enge Zusammenarbeit (und personelle Überschneidung) mit MPEG
- H.261: Vorläufer von MPEG-1
  - ohne B-Frames, MVs nur auf volle Pixel genau
- H.262 ist identisch zu MPEG-2 (gemeinsame Entwicklung)
- H.263: Vorläufer von MPEG-4 Simple Profile
  - PB-Doppelframes statt B-Frames
  - H.263+: AC-Prediction, In-Loop Deblocking Filter
- H.264: Bestandteil von MPEG-4
- H.265: wird wahrscheinlich auch in MPEG-4 integriert

Windows Media Video

- Microsoft ist maßgeblich an der Entwicklung von MPEG-4 und H.264 beteiligt
- parallel dazu Entwicklung eigener Videocodecs

Windows Media Video 7 (WMV1 / »WMV7«)

- ähnlich MPEG-4 Advanced Simple Profile

Windows Media Video 8 (WMV2 / »WMV8«)

- In-Loop Deblocking Filter
- variable Transformationsgrößen: 8x8, 8x4, 4x8
- XINTRA8-Bilder: 8x8 Intra Prediction (»**Spatial Prediction**«)
  - 10 Modi, komplexere Aufbereitung als bei H.264

Windows Media Video 9

Windows Media Video 9 (WMV3 / »WMV9«)
Simple and Main Profile (SP/MP)

- variable Transformationsgrößen
  - 8x8, 8x4, 4x8, neu: 4x4
  - spezifiziert als Integertransformationen
- Overlapped Transform
  - modifizierte Transformation für Intra-Blöcke
  - erfordert zusätzlichen Schritt bei der Rekonstruktion: **Overlap Smoothing**
- Intensity Compensation
  - verändert Helligkeit und Kontrast des Referenzbilds
  - zur besseren Codierung von Fade-Effekten

Windows Media Video 9

- Range Reduction
  - Bilder können mit halbem Kontrast codiert werden
  - Rückumwandlung geschieht erst bei der Anzeige
- Multi-Resolution Coding
  - GOPs können wahlweise in horizontaler, vertikaler oder beiden Richtung in halber Auflösung gespeichert werden
- Bitplane Coding
  - bestimmte Makroblocktyp-Flags werden für das ganze Bild zusammengefasst und effizient abgespeichert
- XINTRA8-Bilder fehlen wieder
  - funktioniert nicht gut zusammen mit Overlapped Transform
- Bewegungskompensation auf 1/4 Pixel genau

VC-1

- WMV9 erweitert um **Advanced Profile** (AP)
  - wesentliche Neuerung: Interlaced Coding nach MPEG-2-Art
  - Range Reduction und Multi-Resolution verallgemeinert
    - beliebige Helligkeits- und Kontrastverschiebungen bei der Ausgabe möglich
    - Bildauflösung kann GOP-weise beliebig geändert werden

- standardisiert als **SMPTE 421M**
  - bekannter unter dem Namen **VC-1**

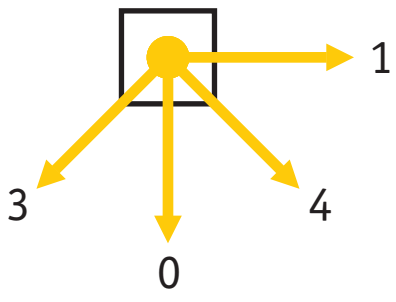
- einer der obligatorischen Codecs für die Blu-ray Disc

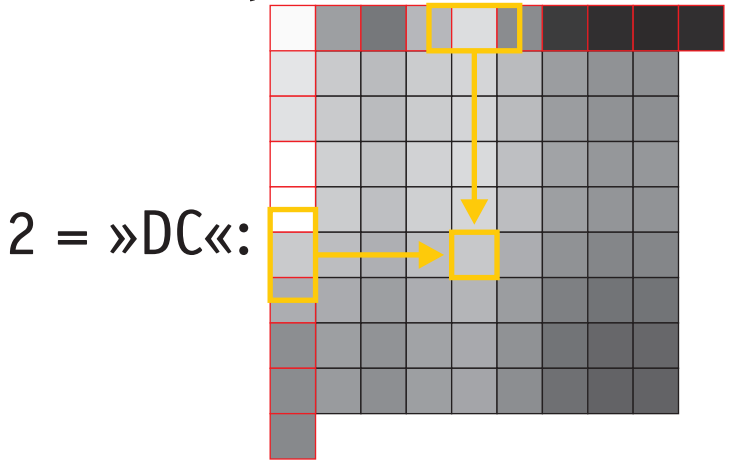
AVS

- chinesischer Videokompressionsstandard
  - Projekt initiiert von der chinesischen Regierung
  - Ziel: Unabhängigkeit von ausländischen Patenten
- erstes entwickeltes Profil: **Jizhun Profile**
  - H.264-ähnlich (»verschlanktes« H.264)
  - 8x8-Integertransformation
  - Partitionsgrößen 16x16, 16x8, 8x16, 8x8
  - Bewegungskompensation auf 1/4 Pixel genau
    - 4-tap-FIR-Filter für Halb- und Viertelpixelpositionen
  - Exp-Golomb-Codes für alles, auch Koeffizienten
  - max. 2 Referenzbilder
  - PAFF, aber kein MBAFF bzw. Frame Interlaced Pictures
- neueres Profil: **Zengqiang Profile** mit CABAC, MBAFF und konfigurierbarer Koeffizienten-Reihenfolge

AVS

- Intra Prediction: Chroma wie H.264, Luma 5 Modi
  - komplexerer DC-Modus: tiefpassgefilterte Prädiktion aus jeweils oberen und linken Randpixeln





  - im Zengqiang Profile alle 9 IP-Modi aus H.264
- in B-Bildern **Symmetric Mode**:
  - Forward-Vektor wird codiert
  - Backward-Vektor wird aus Forward-Vektor und Bilddistanz berechnet

RealVideo

- Codec-Familie der Firma **RealNetworks**
  - Zielanwendung: Videoconferencing, später General Purpose
  - erste RealVideo-Versionen waren H.263-Derivate
  - später proprietäre Eigenentwicklungen: **RealVideo 8-10**
- Motion Compensation wie MPEG-4
  - RealVideo 8: 1/3 Pixel, RealVideo 9: 1/4 Pixel Genauigkeit
  - MVs dürfen nicht außerhalb des Bildes zeigen
  - in RealVideo 9 zusätzlich 16x8- und 8x16-Partitionen
- Intra Prediction und 4x4 Integer Transform ähnlich H.264
- In-Loop Deblocking
- Bildauflösung darf sich frameweise ändern
  - Skalierung von Referenzbildern notwendig
- RealVideo 9: optional Interlaced Coding
- RealVideo 10: ???

On2 VP

- Codec-Familie der Firma **On2 Technologies**
- versucht, allen Patentansprüchen aus dem Weg zu gehen
  - daher z.B. keine B-Frames
- VP3: ähnlich MPEG-4 + In-Loop Deblocking Filter
  - als Open Source freigegeben
  - dient als Grundlage für **Ogg Theora**
  - zusätzliches Referenzbild (»Golden Frame«)
  - Blöcke und Makroblöcke nicht in Raster Scan Order, sondern entlang einer Hilbert-Kurve codiert
  - intensiver Gebrauch von Data Partitioning
- VP5: Arithmetische Codierung als Entropiecodierung
- VP6: QPel
  - lizenziert von Macromedia (jetzt Adobe) für Flash 8
- VP7: 4x4-Transformationen, Intra Prediction, Intensity Comp.?

On2/Google VP8

- 2010 Übernahme von On2 durch Google, VP8 wird Open Source
- Ziel: VP8 als Standard-Codec für Internet-Video (**WebM**)
  - und als JPEG-Nachfolger: **WebP** = VP8 Intra-Frames
- Codierung wieder in Raster Scan Order
- intensiver Gebrauch von Data Partitioning
- Arithmetische Codierung, nicht-adaptiv
- Intra Prediction nahezu identisch mit H.264
- keine B-Bilder, aber Long-Term Prediction mit 3 Referenzbildern: Previous, Golden, AltRef
- Partitionsgrößen: 16x16, 16x8, 8x16, 8x8, 4x4 (kein 4x8/8x4)
- 1/4-Pixel MVs, 6-tap-Filter (auch für 1/4-Pixel-Positionen)
- 4x4-Integertransformation; kein 8x8
- makroblockweise Parametersteuerung durch **Segmentierung**
- In-Loop Deblocking Filter

On2/Google VP9

- VP9: Weiterentwicklung von VP8, im Mai 2013 fertiggestellt
- bis zu 8 Referenzbilder, davon 3 in jedem Frame nutzbar
- 32x32 und 64x64 Pixel große »Superblöcke«, außerdem Tiling
- DCT-Größen von 4x4 bis 32x32, zusätzlich DST
- erweiterte Segmentierung mit temporaler Prediction
- Intra Prediction aus allen umliegenden Inter-Blöcken
- Compound Prediction: Mischung aus mehreren Predictions
  - auch Inter+Intra mit gewichteter Mischung möglich: Pixel nahe der Kante werden mehr nach Intra gewichtet
- MV Prediction bewertet Vektorkandidaten danach, wie gut sie die umliegenden Pixel approximieren
- MC-Interpolationsfilter in 3 verschiedenen Schärfevarianten
- Auflösungsänderung auf Frame-Ebene

Xiph.org Ogg

- Xiph.org entwickelt ebenfalls (patent-)freie Codecs
  - hauptsächlich bekannt durch die Audiocodecs **Vorbis**, **FLAC**, **Speex** und **Opus**
- **Theora:** einfacher MPEG-4-ähnlicher Videocodec auf Basis von On2 VP3
- **Daala:** derzeit in der Konzept-/Experimentierphase
  - Ziel: H.265-ähnliche Effizienz
  - hierarchische Blockstruktur wie in H.265
  - Overlapped Transform, integer-basiert, 100% reversibel
  - Frequency Domain Intra Prediction
    - nicht auf Basis umliegender Pixel, sondern Koeffizienten
    - Blockgrößenanpassung durch Time/Frequency Switching
  - nicht-binärer arithmetische Entropiecodierung

Die Zukunft?

- **konventionelle blockbasierte Verfahren**
  - dürften irgendwann ausgereizt sein ...
  - Verbesserungen der Codiereffizienz in den letzten Jahren nur im Prozentbereich, größere Sprünge nur durch Kombination vieler kleiner Features
- Paradigmenwechsel zu **Wavelets**?
  - Codierung von Standbildern relativ effizient
    - ... aber in der gleichen Liga wie blockbasierte Verfahren
  - aber: wie kann MC realisiert werden?
    - Overlapped Block Motion Compensation (OBMC)
    - 3D-Wavelets: 2D Raum + 1D Zeit
    - pixelweise MC durch Wavelet-komprimierte Motion Maps
  - kein Wavelet-Verfahren konnte sich bisher durchsetzen

Das war's!

Vielen Dank für Ihre Aufmerksamkeit!

Noch Fragen?

Mail an:
martin.fiedler@gmx.net

Folien im Internet:
http://keyj.emphy.de/files/projects/videocomp.pdf