

Netzwerktechnologie für Multimedia Anwendungen (NTM)

Kapitel 2

Florian Metzger

florian.metzger@univie.ac.at

David Stezenbach

david.stezenbach@univie.ac.at

Bachelorstudium Informatik
WS 2014/2015

2. Technische Grundlagen

- 2.1 Geräte
- 2.2 Digitalisierung, Kodierung & Komprimierung
- 2.3 Audiokompressionsverfahren
- 2.4 Videokompressionverfahren

2. Technische Grundlagen

- **2.1 Geräte**
- 2.2 Digitalisierung, Kodierung & Komprimierung
- 2.3 Audiokompressionsverfahren
- 2.4 Videokompressionverfahren

Geräte-Kategorien

- Input
- Verarbeitung
- Speicherung
- Transport
- Output

Input

- Mikrophon
- Digicam
- Camcorder
- Tastatur, Maus, Touchpad, Touchscreen, Gamepad
- Bewegungssensoren und Motion Capture, 3D-Scanner
- Weitere Sensoren
 - Geolocation, Barometer, Temperatur, Gyroskope, ...

Videokameras

- Historische Camcorder Nutzungswege:
 - Camcorder → TV
 - Camcorder → Videorecorder → TV
 - Camcorder → PC Capture Card → PC → DVD → TV
 - Camcorder → Firewire/USB/SD-Karte → PC (→ DVD) → TV/Monitor
- Außer im professionellen Bereich heutzutage fast vollständig abgelöst von Smartphones und Action-Cams
 - Direkter Upload zum Rechner oder Video-Sharing Seiten



Video-Verarbeitung

Knoten, die Daten verändern, filtern, verwerfen, usw.

- Rechen- und Speicherintensiv
- Besonders geeignet für GPUs
- Server-Racks
- Tablets und Smartphones (Android, iOS, usw.)
- Special-Purpose Transcoding HW

Datentransport

Geräte für den Datentransport:

- Netzwerk-Karte (Adapter)
- Switch, Hub
- Router
- Modem (DSL, Kabel, 3G/4G, usw.)
- Satelliten Schüssel (inkl. LNB/LNC), Antenne

Speicherung

Geräte zur temporären/dauerhaften Speicherung:

- Festplatte (intern, extern)
- Network Attached Storage (NAS)
- Disk: CD, DVD, Blu-Ray
- Flash-Speicher
 - USB-Stick
 - Speicherkarten
 - SSD
- Bandspeicher

Ausgabe

- Visuell
 - 2D/3D Anzeigen
 - 3D Brillen
- Akustisch
 - Mehrkanalig
- Haptisch
 - Force Feedback
 - Braille
- Olfaktorisch und anderes?

Ausgabegeräte

- Set-top Boxen
 - Meist keine General Purpose Geräte (ASICs)
 - Oft proprietär, mit geschlossener Architektur
 - Von Provider für eine spezifische Anwendung zur Verfügung gestellt
 - Geringe Rechenleistung, Geringer Stromverbrauch
- Konsolen
 - Oft teuer, aber relativ leistungsfähig und hoher Stromverbrauch
 - Multimediazentrale im Heim (Wohnzimmer)
 - Nutzen von Streamingangeboten von mehreren Anbietern
 - Beispiele: PS4, Xbox One, Wii U, OUYA, ...
- Zwischenstufe: Streaming Media Player (z.B. Apple TV, Chromecast)

Notation

Bildaufbau

- **Frame:** Vollbild
 - Progressiv (Vollbilder): 'p'
- **Field:** Halbbild (Nur jede 2. Zeile)
 - Interlaced (Halbbilder): 'i'

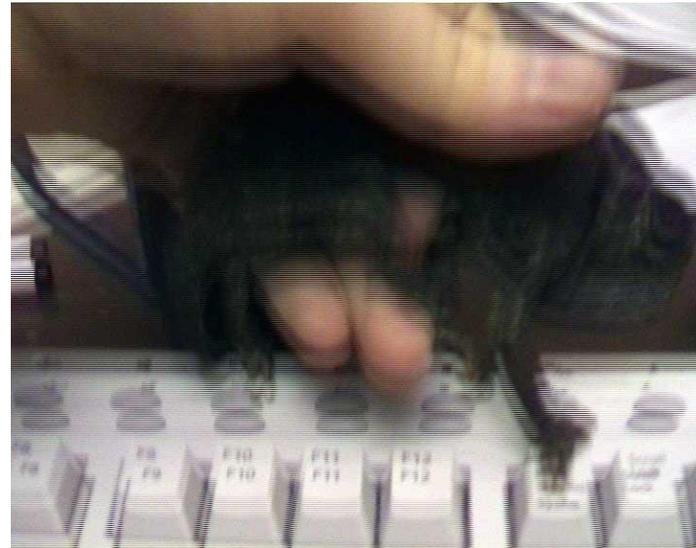
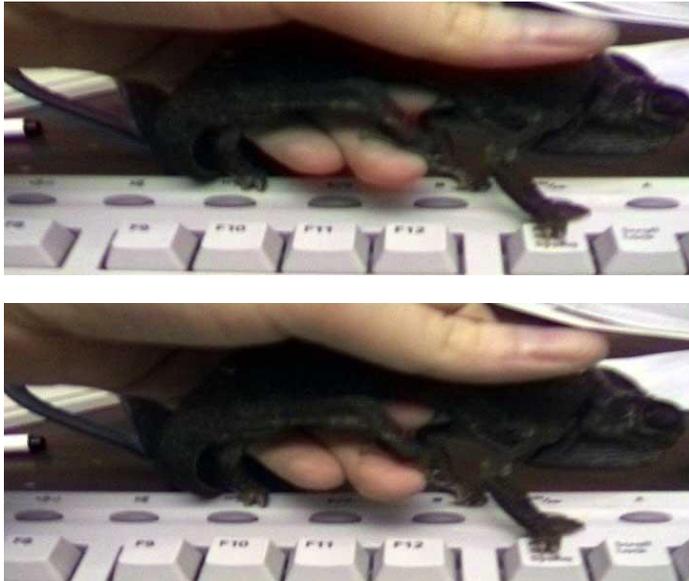
Bildwiederholrate

- Frames per second, fps, Hz

Menschliche Wahrnehmung

- Wahrnehmung von **Flimmern (Licht ein/aus)**
 - Vermeiden durch hohe Bildwiederholraten
- Bedingt durch verwendete Technologie
 - Analog-Kino: Rotierende Blende verdeckt Filmtransport
 - TV (Kathodenstrahlröhre)
 - Elektronenstrahl tastet Fluoreszenzschirm ab
 - Nur sehr kurzes Nachleuchten, dann wieder Dunkel
 - Trägheit des Auges erzeugt Illusion eines Bildes
- Historische Abhilfe: Interlacing (Zeilensprungverfahren)
 - TV: 2 Halbbilder verdoppeln die Bildfrequenz (50Hz) bei gleicher Bandbreite im Vergleich zu 25Hz

Fields vs. Frame



50 Halbbilder pro Sekunde

Darstellung abwechselnd zeilenbasiert → Interlaced

Deinterlacing

Problem

- Analoge Aufnahmen und einige TV-Übertragungen sind interlaced
- LCD, Plasma, Beamer zeigen nur **Vollbilder** an
- Interlacing Artefakte sichtbar (**Kämme**)

Deinterlacing

- Umwandlung von **Halbbildern** in **Vollbilder**
- **Herausrechnen** von Kämmen mit **Bild-Filtern (z.B. Weaving)**

Deinterlacing



Bewegungsauflösung des Auges

- Kino: 24Hz, PAL: 25Hz, NTSC: 29.97Hz
 - Natürlicher **Motion Blur** entsteht durch Aufnahmetechnik
 - Maskiert Großteil des Ruckelns
- Monitore: 60/120/144+Hz
 - Ausgabe muss synchron zur Monitorgeschwindigkeit erfolgen
 - Sonst Synchronisierungsartefakte und Ruckeln (24p Modus für Filme)
- Videogames:
 - interaktiv!
 - Keine implizite Maskierung durch Motion Blur
 - 30 oft nicht genug, 60+

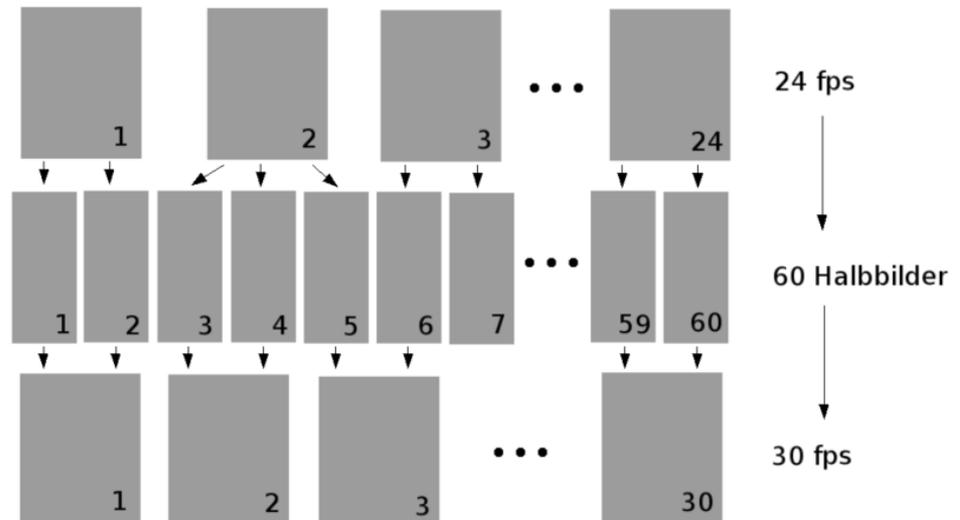
Historische Analog-Formate

- Kino:
 - 24p, 35mm Film (äquivalent zu ca. 4-16 Megapixel)
 - Seitenverhältnis oft 2,40:1 oder 1,85:1
- PAL (digital):
 - **720x576**, interlaced (576i)
 - 50 Halbbilder, 25Hz
 - Seitenverhältnis: 4:3
- NTSC D1:
 - **720x480**,interlaced (480i)
 - 60x1000/1001 Halbbilder pro Sekunde, 29.97Hz
 - Seitenverhältnis: 4:3

Formatkonvertierungen

- Kino (24p) in PAL (25Hz)
 - Der Film wird um den Faktor $25/24$ schneller
 - Stimmen, Geräusche werden höher (kaum zu merken)
 - Frequenzhöhe per digitaler Signalverarbeitung anpassbar
- Kino (24p) in NTSC (29.97Hz)

- **3:2 Pulldown**



Von Analog zu Digital

Analogfernsehen und Radio sind mittlerweile fast vollständig durch digitale Nachfolger ersetzt:

- CRT → LCD, Plasma, Beamer, OLED
- VHS → LaserDisc, DVD, Blu-Ray
- PAL, NTSC → DVB
- Radio (FM, AM) → DAB, DAB+

Digitale physische Medien werden nun teilweise auch schon durch Streamingdienste ersetzt

Neuere Bildschirmtechnologien

LCD, Plasma, OLED

- Lange Nachglühzeiten der Dioden
- Macht hohe Bildwechsel-/wiederholraten problematisch
- Bildanzeige immer progressiv
- Alle Pixel **gleichzeitig** dargestellt
- Oft eingeschränkte Blickwinkelstabilität
- Verschiedene Konstruktionsweisen (Panels)
 - Bestimmen Farbtreue, Blickwinkelstabilität, Geschwindigkeit, ...

Digitale Videostandards

Benötigen

- Wandlung von analog zu digital
- Kodierung, Speicherung, Wiedergabe von großen Datenmengen
- Transport zum Konsumenten
- Evtl. auch Rückwärtskompatibilität
 - DVD soll z.B. mit PAL und NTSC CRT Fernsehern funktionieren
 - Digitale Nachfolger haben teilweise ähnliche Probleme geerbt
 - Auflösung
 - Umgang mit Interlacing

Gängige Auflösungen TV/Video

Name	Norm	Auflösung	Seitenverhältnis
480i/p	NTSC	720x480	4:3
576i/p	PAL	720x576	4:3
XGA	XGA	1024x768	4:3
720p	HD	1280x720	16:9
1080i/p	Full HD	1920x1080	16:9
„4K“	UHD	3840x2160	16:9
„8K“	FUHD	7680x4320	16:9

- Andere Auflösungen:
 - 1600x900, 1480x1152 (16CIF), 704x576 (4CIF), 352x288 (CIF), 176x144 (QCIF), 128x96 (SQCIF), usw.
 - Viele 16:10, 5:4 Varianten
 - Kinofilme oft noch breiter (z.B. 2,4:1)

2. Technische Grundlagen

- 2.1 Geräte
- **2.2 Digitalisierung, Kodierung & Komprimierung**
- 2.3 Audiokompressionsverfahren
- 2.4 Videokompressionverfahren

Digitalisierung

Sinneswahrnehmungen sind **analog**:

- Lautstärke
- Licht
- Druck
- Temperatur

Analoge Medien

Schallplatte, Audio-Kassette/Bandaufzeichnung, VHS/S-VHS,
PAL, NTSC

Beeinträchtigung der Medien durch:

- Verschmutzung, Zerkratzen
- Übertragungsstörungen (→ Schneefall, Geisterbilder)
- Kopier-Fehler

→ **Digitalisierung** von analogen Signalen
(A/D Wandlung + Kodierung)

Digitalisierung

A/D Wandlung:

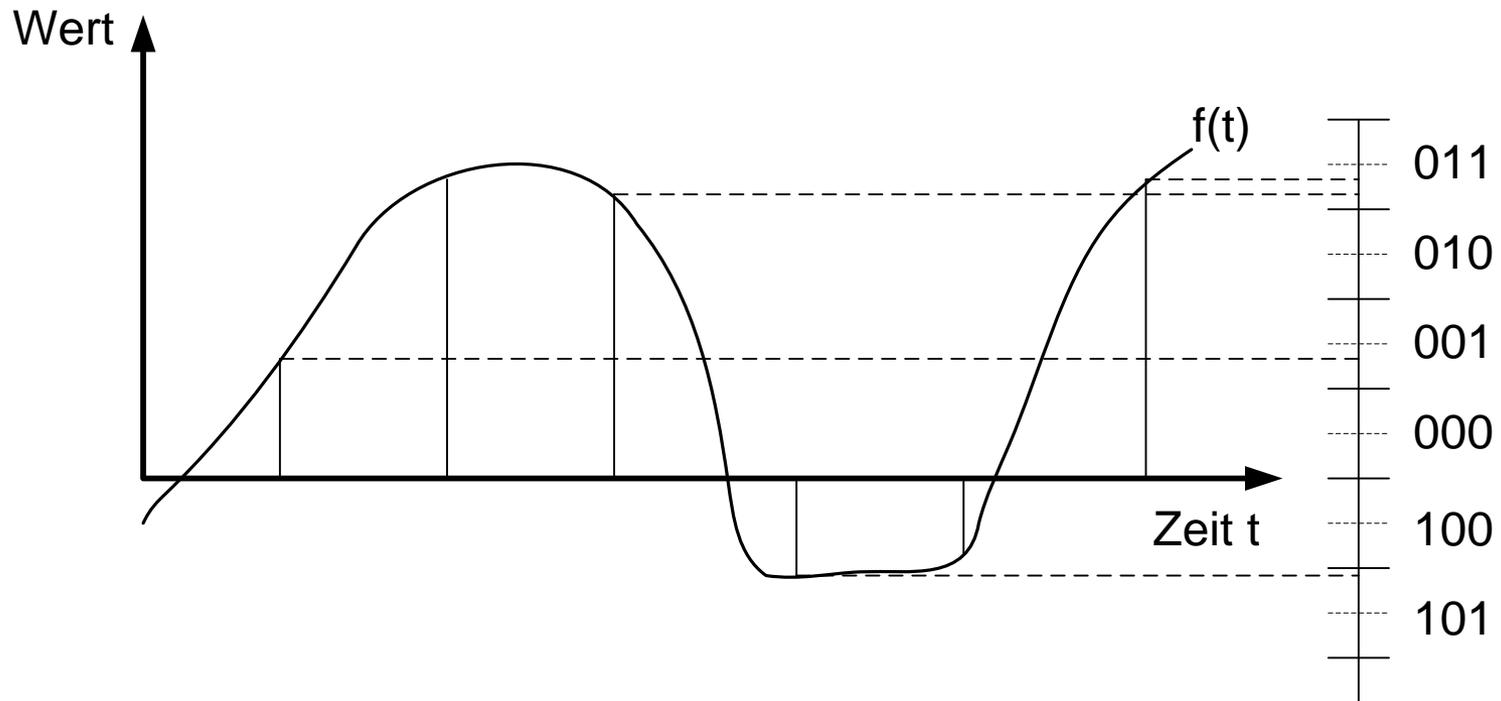
- Quantisierung
 - Analoges Signal wird abgetastet → analoger Wert wird auf digitalen Wert gerundet
 - Verlustbehafteter Prozess
- Potentiell sehr große Datenmengen

Kodierung, Komprimierung:

- Werte werden als Zahlen abgespeichert
- Redundanzen werden potentiell weggelassen

A/D Wandlung

Lineares PCM



Wie oft soll man abtasten?

Nyquist-Shannon-Abtasttheorem

- Signal ist aufgebaut aus Schwingungen (Frequenzen)
- Reproduktion einer Frequenz f
 - Abtastung mit einer Frequenz die infinitesimal höher ist als $2f$ (d.h. $2f + \varepsilon$) und davor analoge „Reinigung“ des Signals von Frequenzen über f (mittels Tiefpass-Filter), um Aliasing-Effekte zu vermeiden!

Beispiel:

- Menschliches Gehör kann bis ca. 20 kHz reproduzieren
 - Musik sollte man mit mind. 40 kHz sampeln! (CD: 44,1 kHz)

Quantisierungsfehler

A/D Wandlung

$$D(t) = S(t) + N(t)$$

- $S(t)$...das eigentliche Signal
- $D(t)$...Quantisiertes Signal
- $N(t)$...**Noise**, Fehler, Störung, Rauschen

Quantisierungsfehler – SNR

Signal to Noise Ratio (SNR) beim Kodieren mit n Bits

Bei gleichförmiger Verteilung des Signals über das Quantisierungsraster gilt:

$$SNR(dB) = 10 \log_{10} \frac{P_{signal}}{P_{noise}} = 20 \log_{10} (2^n) \approx 6,02n \text{ dB}$$

8 Bits...48 dB

16 Bits...96 dB

Kodierung, Komprimierung

Beispiel: Audio CD

- Abtastung mit 44,1 kHz
- 16 bit/Sample → 704 kbit/s
- 2 Kanäle (Stereo) → ca. 1,4 Mbit/s Datenrate

Aufgabe:

Reduktion der Datenmenge durch effiziente Kodierung

Datenkompressionsansätze

- Verlustfrei (lossless)
 - Redundanzen und Korrelationen werden ausgenutzt
 - Exakte Reproduktion der Originaldaten möglich
 - Text muss immer verlustfrei sein
- Verlustbehaftet (lossy)
 - Originaldaten können (im Gegensatz zu verlustfreien Verfahren) nicht mehr ganz originaltreu rekonstruiert werden
 - Der Fehler **sollte** aber für Menschen kaum wahrnehmbar sein
 - Psychovisuelle /-akustische Modelle
 - Mehr darüber später bei der Bildkodierung!

Verlustfreie Kodierung - Lauflängen

Symbolfolge: A,A,A,B,C,D,E,E,E,E,E,E,E,E,E,A

Manche Symbole werden oft wiederholt

Kodiert wird: 3A,B,C,D,10E,A

Einfache binäre Kodierung

- Gegeben: Symbole (beobachtete Werte)
A, B, C, D
- Wahrscheinlichkeiten des Beobachtens:
 $P(A) = 1/2$, $P(B) = 1/4$, $P(C) = P(D) = 1/8$
- Kodierung mit 2 Bits/Symbol: A=00, B=01, C=10, D=11

Durchschnittliche Anzahl der gespeicherten Bits:

$$\frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} \right) \cdot 2 = 1 \cdot 2 = 2$$

Entropie-Kodierung - Huffman

- Entropie:
 - Maß für den mittleren Informationsgehalt pro Zeichen

$$H = - \sum_{z \in Z} p_z \log_2 p_z$$

- Unterschiedliche Wortlänge
- Optimal & Präfixfrei
 - Kürzeste symbolbasierte Kodierung bei bekannten Auftretswahrscheinlichkeiten

Huffman:

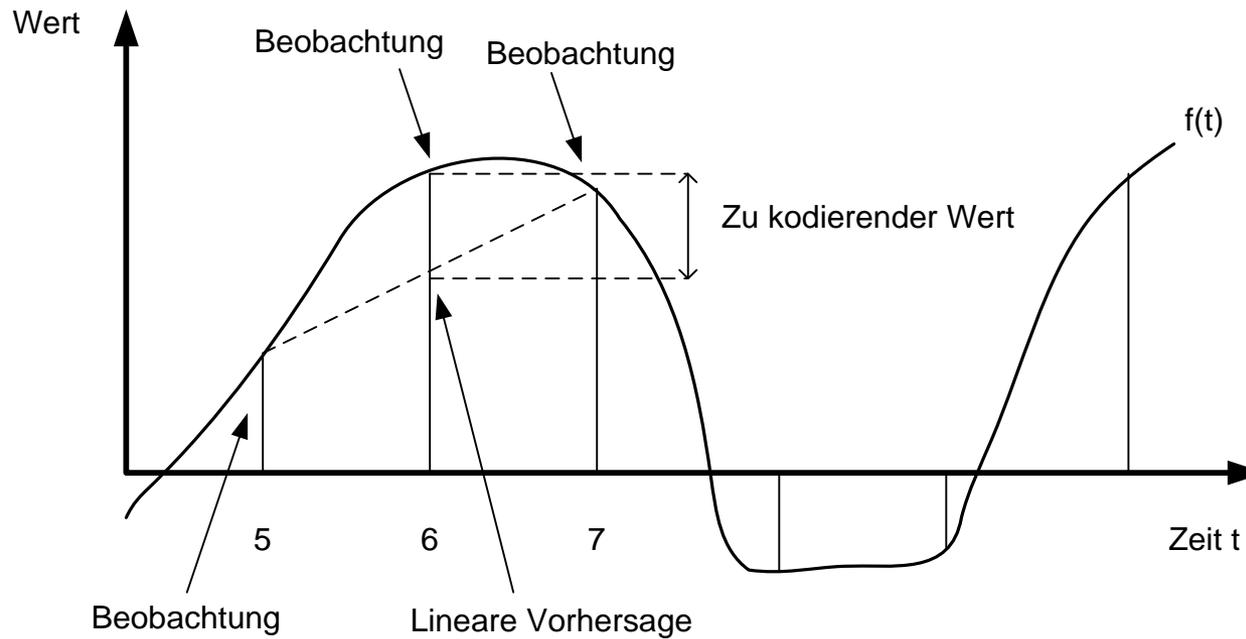
➤ Tafelbeispiel

Ausnutzen von Korrelationen

Nachteil von z.B. Huffman:

- Je mehr Symbole (Werte) es gibt, umso mehr Bits braucht man
 - **Eine Reduktion der Menge** der beobachteten Werte wäre wünschenswert
- Differenzielles Kodieren
 - Der nächste Wert wird aus den vergangenen Werten mittels **Extrapolation** (Modell) geschätzt
 - Kodiert wird nur die **Differenz zur Vorhersage**
 - Je besser die Vorhersage (das Modell), umso weniger beobachtete Werte hat man
 - Wichtig: Modelltyp muss zu den Daten passen!

Ausnutzen von Korrelationen



Ausnutzen von Korrelationen

Beobachtete Werte: $X(t)$, $X(t+dt)$

Modellannahme: $X(t)$ ist fast so groß wie $X(t+dt)$

→ $X(t)$ ist der **Prediktor** für $X(t+dt)$

Es gilt: $X(t+dt) = X(t) + R(t+dt)$, mit $R(t+dt)$ möglichst klein

$R(t+dt)$...**Residuum**, Fehler

$R(t+dt) = X(t+dt) - X(t)$ wird kodiert

Lineare Zeitreihen

- Autoregressives (AR) Modell
- Gegeben sind Beobachtungen X_1, X_2, \dots, X_n
- Modellannahme:

$$X_n = a_1 X_{n-1} + a_2 X_{n-2} + a_3 X_{n-3} + \dots + a_k X_{n-k} + \varepsilon_n$$

Zu modellieren: a_1, \dots, a_k

Kodieren: $\varepsilon_n = X_n - (a_1 X_{n-1} + a_2 X_{n-2} + a_3 X_{n-3} + \dots + a_k X_{n-k})$

Beobachtung

- Um X_n dekodieren zu können benötigt man

$$X_{n-k}, X_{n-k+1}, \dots, X_{n-1}$$

- Angenommen X_{n-1} wurde nicht übertragen
 - Kann Werte, die auf X_{n-1} beruhen, nicht mehr korrekt dekodieren
 - Fehler

2. Technische Grundlagen

- 2.1 Geräte
- 2.2 Digitalisierung, Kodierung & Komprimierung
- **2.3 Datenkompressionsverfahren**
- 2.4 Videokompressionverfahren

FLAC

- Free Lossless Audio Codec
- Reduktion um ca. 50%

Signal wird durch ein Modell angenähert

- **Polynom** oder **lineares Zeitreihenmodell**
- Residuen = Differenz Modell – echte Werte
- Spezielle **Huffman** Kodierung (Rice) für **Residuen**

ITU-T G.711 Audio Codec

- „Verlustbehaftet“
 - Verlustfreie Kompression, aber Kompondierung
- Globaler **Telefonstandard**
- **Weiteste Verbreitung, keine Patentgebühren**
- Low latency encoding (0.125 ms) -> Echtzeit
- Kompondierung (~Dynamikkompression) von 14 bit (μ -Law) bzw 13 bit (A-Law) Eingabesamples auf 8 bit zur Übertragung

ITU-T G.711 Audio Codec

- Abtastrate 8 kHz
- **Blöcke** bei Packet Switching: **10 ms**, 20 ms, 30 ms
- Verarbeitung der Samples
 - Kodierung mit Tabelle
 - 1 Bit Vorzeichen (Sign)
 - 3 Bits Größenordnung (Exponent)
 - 4 Bits Genauigkeit (Mantisse)
 - **Kodierung in 8 bits/Sample**
- Datenrate: 64 Kbit/s
- **Pro Paket: 2 Blöcke, 160 Bytes, 20 ms Sprache**
- 50 Pakete/s

ITU-T G.729 Audio Codec

- Bitraten: 6.4 Kbit/s, **8 Kbit/s** oder 11.8 Kbit/s
- Teilweise Patentbehafte
- Geeignet für schmalbandige Verbindungen
- **Blöcke** bei Packet Switching: **10 ms**, 20 ms, 30 ms
- Abtastrate 8 kHz (300 – 3400 Hz **Sprache**)
- **Sprachpausen werden nicht kodiert**
- Codebook Excited Linear Predictor (**CELP**)
 - Basiert auf einem linearen Zeitreihenmodell

ITU-T G.729 Audio Codec

Netzwerkübertragung

- 2 Blöcke / übertragenes Paket
 - **Jedes übertragene Paket repräsentiert 20 ms Sprache**
- Kodierungslatenz: ca. 25 ms
- 50 Pakete/s
- Gleichabständig

MPEG-1/2 – Audio Layer III (MP3)

- Kompression erfolgt in **Frequenzdomäne**
- Transform Encoder:
 - Zeitsignal wird zuerst in Frequenzen zerlegt
- Perceptual Codec:
 - Frequenzen die nicht gehört werden können, werden nicht kodiert
- Samplingraten: 8 - 48 kHz
- Bis zu 5.1 Kanäle möglich (2.0 üblich)
- Bitraten: 8 - 320 Kbit/s, VBR Modus
- Stückelung: 384, 576 oder 1152 Abtastwerte pro **Frame**
- Hohe Effizienz, aber hohe Kodierungslatenz (~200 ms)

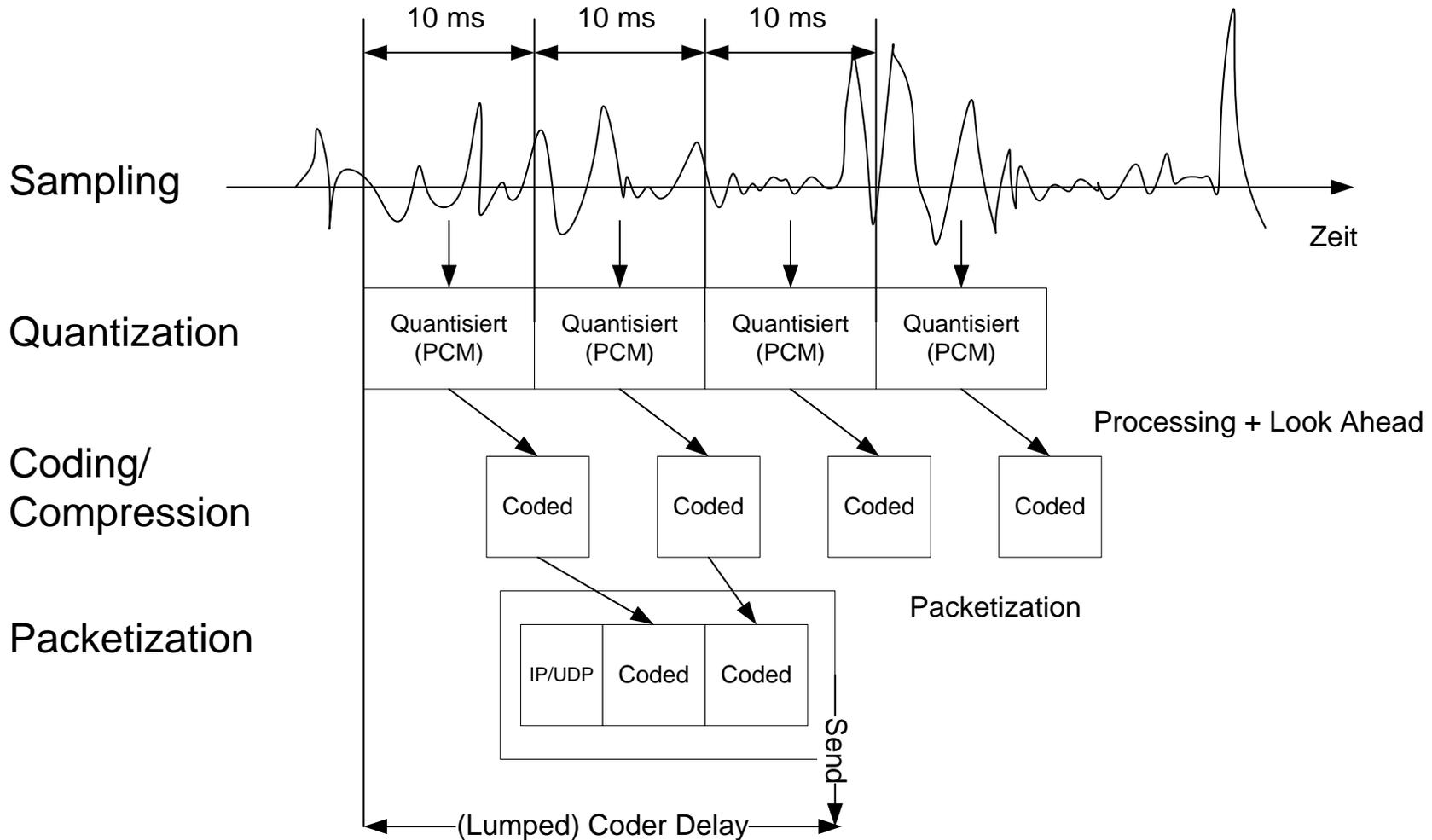
MPEG-4 AAC

- Advanced Audio Coding
- 8 – 96 kHz, Sampling
- Bis zu 48 Kanäle
- Beliebige Bitraten, Frame Längen
- Dateiformat streaminggeeignet
- Vergleichbare Audioqualität zu MP3 bei gleichen Bitraten
- Große Industrieunterstützung, daher kommerziell weit verbreitet

Opus

- Offener Standard RFC 6716
- Kombination aus SILK und CELT
- Sehr flexibel
 - 6 bis 510 Kbit/s Bitraten
 - 8 bis 48 kHz Abtastraten
 - 5 bzw. 26,5 ms Kodierungslatenz
- Kann viele andere Audiokompressionsverfahren ersetzen
 - z.B. MP3, G.729
 - Bessere Qualität bei vergleichbaren Bitraten

Kodierungslatenz (*Algorithmic Delay*)



Audio Echtzeitübertragung

- Audiodaten werden
 - komprimiert
 - in Pakete gesteckt
- Pakete werden über das Netzwerk übertragen
- **Oft mit fixer Bandbreite (= regelmäßiger Übertragung)**

Üblich:

- **20-30 ms Sprache/Paket** \Leftrightarrow 33,3 – **50 Pakete/s**

Verlustfreie Bildkompression

GIF

- Seit 2004 patentfrei und frei nutzbar
- Nur 8bit Farbpalette aus 24bit Farbraum nutzbar
- LZW-Kompression (Lempel-Ziv-Welch)
- Animated GIFs

PNG (RFC 2083)

- Bessere Kompression (DEFLATE) und Farbraum (32bit)
- Nicht Animated

Bildformate – JPEG

- Joint Photographic Experts Group
- Normen: ITU T.81, ISO 10918-1
- Kompressionsrate bis zu 1:100

Rohdaten

- Bild mit Auflösung $X*Y$ Pixel
- Jeder Pixel hat drei Werte (Rot, Grün, Blau)
- → Drei Bild-Matrizen (R, G, B)

Bildformate – JPEG

Komprimierungsschritte

1. Umrechnen RGB auf YC_bC_r / YUV Farbräume
2. Auflösung der Farbkomponenten C_b und C_r wird halbiert (verlustbehaftet)
3. 8x8 Blöcke, **diskrete Kosinustransformation (DCT)**
4. Quantisierung der Frequenzen → verlustbehaftet
5. Umsortierung, Differenz DC zum linken Nachbarn
6. Lauflängenkodierung, Huffman, Arithmetisch

JPEG – Farbraum

RGB auf YUV / YC_bC_r

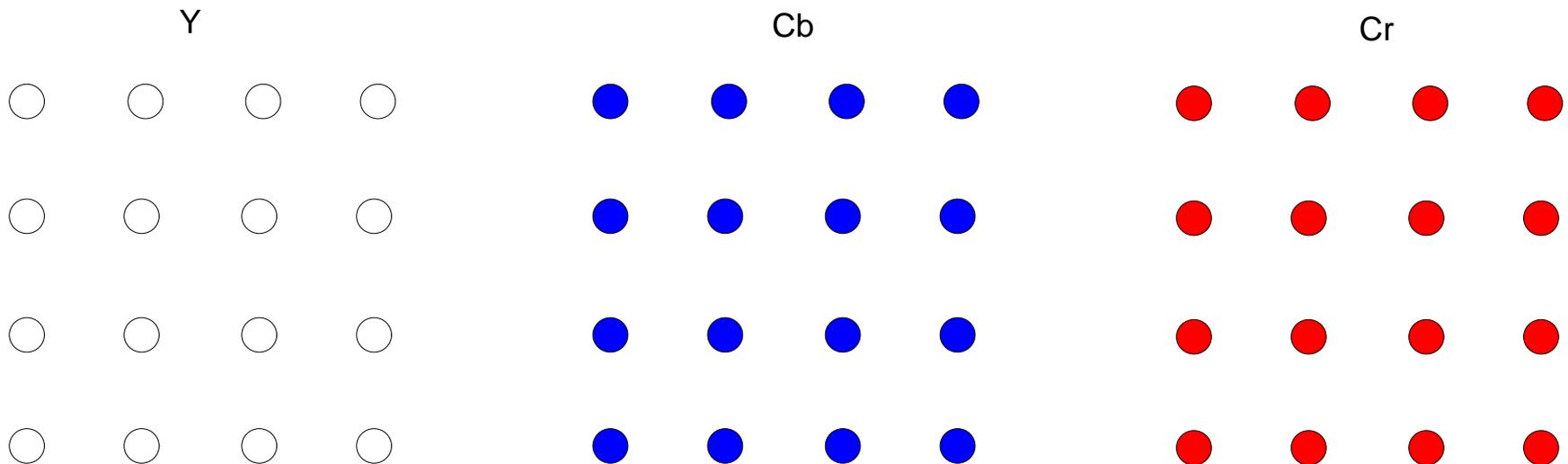
- Jeder Pixel hat 3 Werte für Rot, Grün, Blau
- Y...Helligkeit (Luminance)
- U...**Abweichung** Y zu Blau, $C_b = U+128$ (Chrominance)
- V...**Abweichung** Y zu Rot, $C_r = V+128$

Ergebnis: Drei Bildmatrizen (Y, C_b , C_r)

JPEG – Subsampling

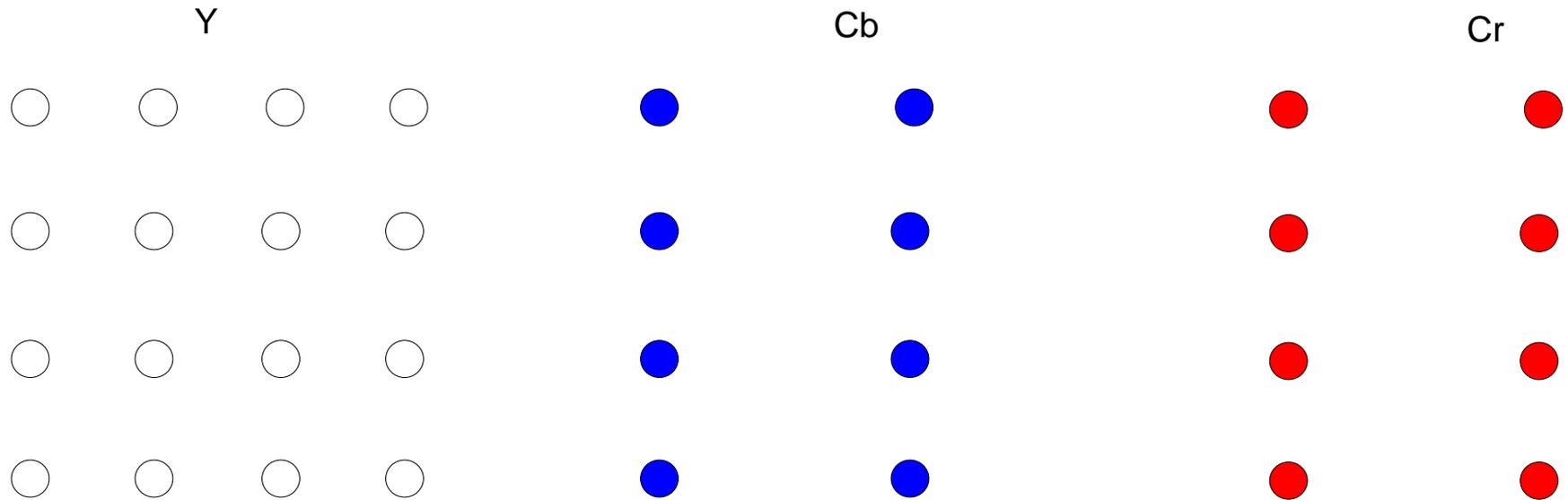
- **Auflösung** der Farbkomponenten C_b und C_r wird halbiert (verlustbehaftet)
- Unterabtastung (Downsampling, Chroma-Subsampling)
 - Grund: Menschliches Auge empfindlicher für Helligkeit als Farbe
 - 4:4:4... kein Subsampling
 - 4:2:2... C_b , C_r **horizontal** mit halber Auflösung
 - 4:2:0... (ein Pixel C_b und C_r) / (4 Pixel Y)

JPEG – Subsampling



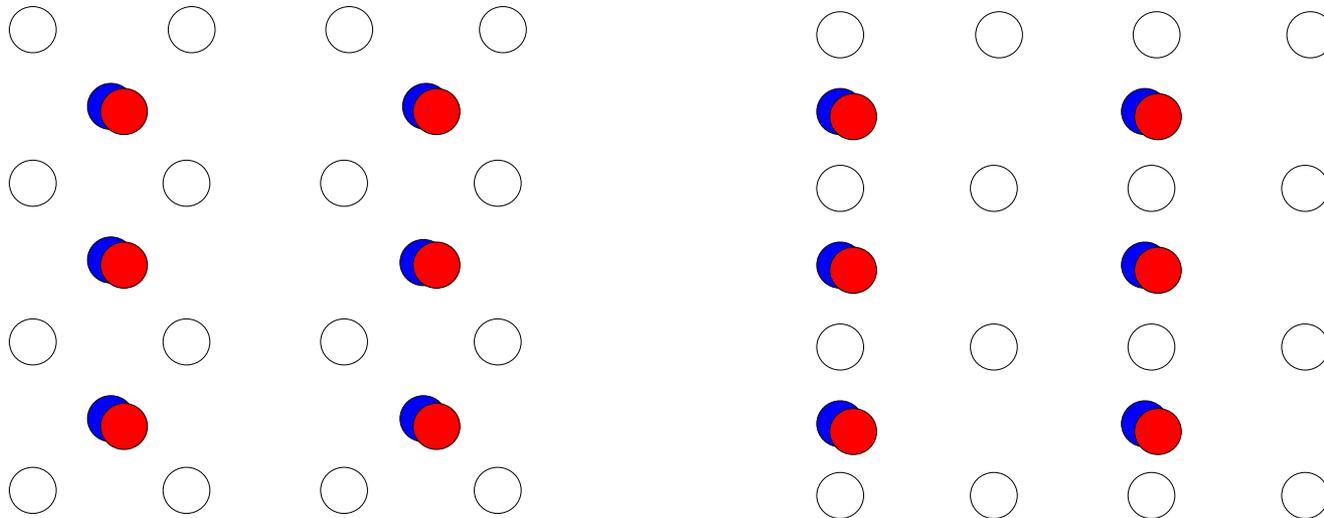
Subsampling 4:4:4

JPEG – Subsampling



Subsampling 4:2:2

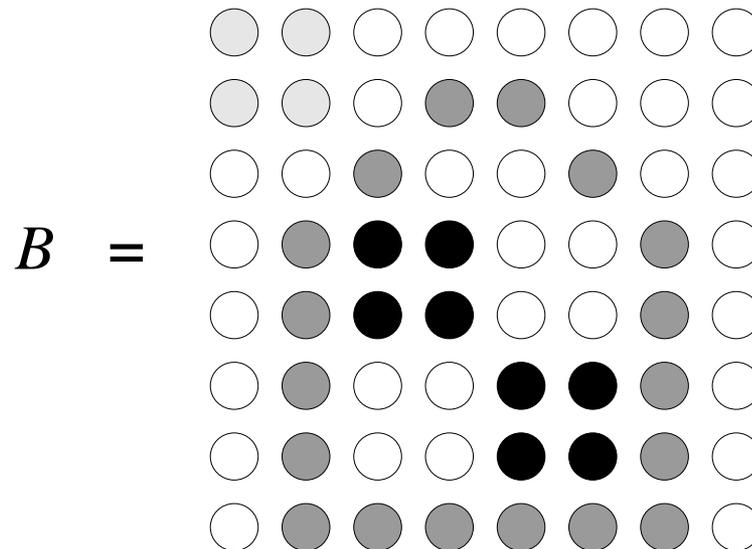
JPEG – Subsampling



Subsampling 4:2:0

JPEG – 8x8 Blöcke

- Matrizen für Y , C_b und C_r in 8x8 Blöcke unterteilt
- Bildwerte: $f_{x,y}$



JPEG – Diskrete Kosinustransformation

Diskrete Kosinustransformation (DCT)

- 64 **Bild-Werte** $f_{x,y}$ des 8x8 Blocks B werden übergeführt in 64 **Frequenzanteile** $F_{u,v}$
- Typ II DCT

$$F_{u,v} = \sum_{x=0}^7 \sum_{y=0}^7 f_{x,y} \cdot M_{u,v}(x, y) \quad 0 \leq u \leq 7, 0 \leq v \leq 7$$

$$F_{u,v} = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 f_{x,y} \cdot \cos\left(\frac{\pi u (2x+1)}{16}\right) \cdot \cos\left(\frac{\pi v (2y+1)}{16}\right)$$

JPEG – Diskrete Kosinustransformation

- $F_{0,0}$... **DC**-Komponente
 - DC gibt den mittleren Wert des Blocks an
 - DC-Komponenten benachbarter Blöcke sind oft ähnlich groß
- Alle anderen $F_{u,v}$... **AC**-Komponenten
 - beinhalten die höherfrequente Bilddetails
- **Kompression durch Weglassen/Vergrößern von (hohen) AC-Komponenten**
- Frequenzen vs. Bildschärfe:
 - Mehr Frequenzanteile bedeutet mehr Bilddetails

JPEG – Quantisierung

- Hohe Bildfrequenzen (AC-Komponenten) weglassen, oder Wertebereich verkleinern → effizientere Kodierung
- “Graceful Degradation of Picture Quality”

Gegeben Quantisierungsmatrix $Q_{u,v}$

$$\tilde{F}_{u,v} = \left\lfloor \frac{F_{u,v}}{Q_{u,v}} \right\rfloor$$

JPEG – Quantisierung

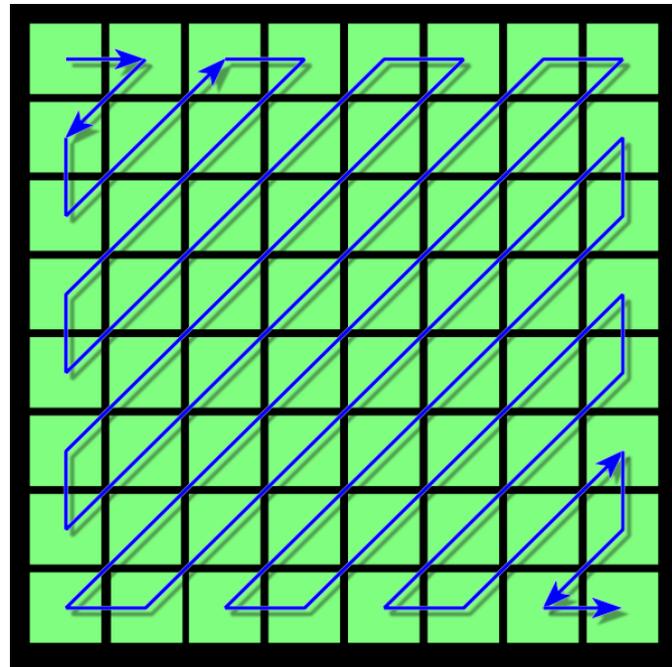
$$F_{u,v} = \begin{matrix} & & & \begin{matrix} u \\ \longrightarrow \end{matrix} & & & & & \\ \begin{matrix} \downarrow v \\ \end{matrix} & \begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} & & & & & & & & \end{matrix}$$

$$Q_{u,v} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\tilde{F}_{u,v} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

JPEG – Quantisierung

- Zigzag Reihenfolge
- Kodieren mit Huffman / Lauflängen



JPEG – Zusammenfassung

- YC_bC_r
- Subsampling für C_b und C_r
- DCT auf 8x8 Blöcken
- Quantisierung
- DC-Differenz
- Lauflängen/Huffman in Zigzag

JPEG – Beispiel



79 KB



4 KB

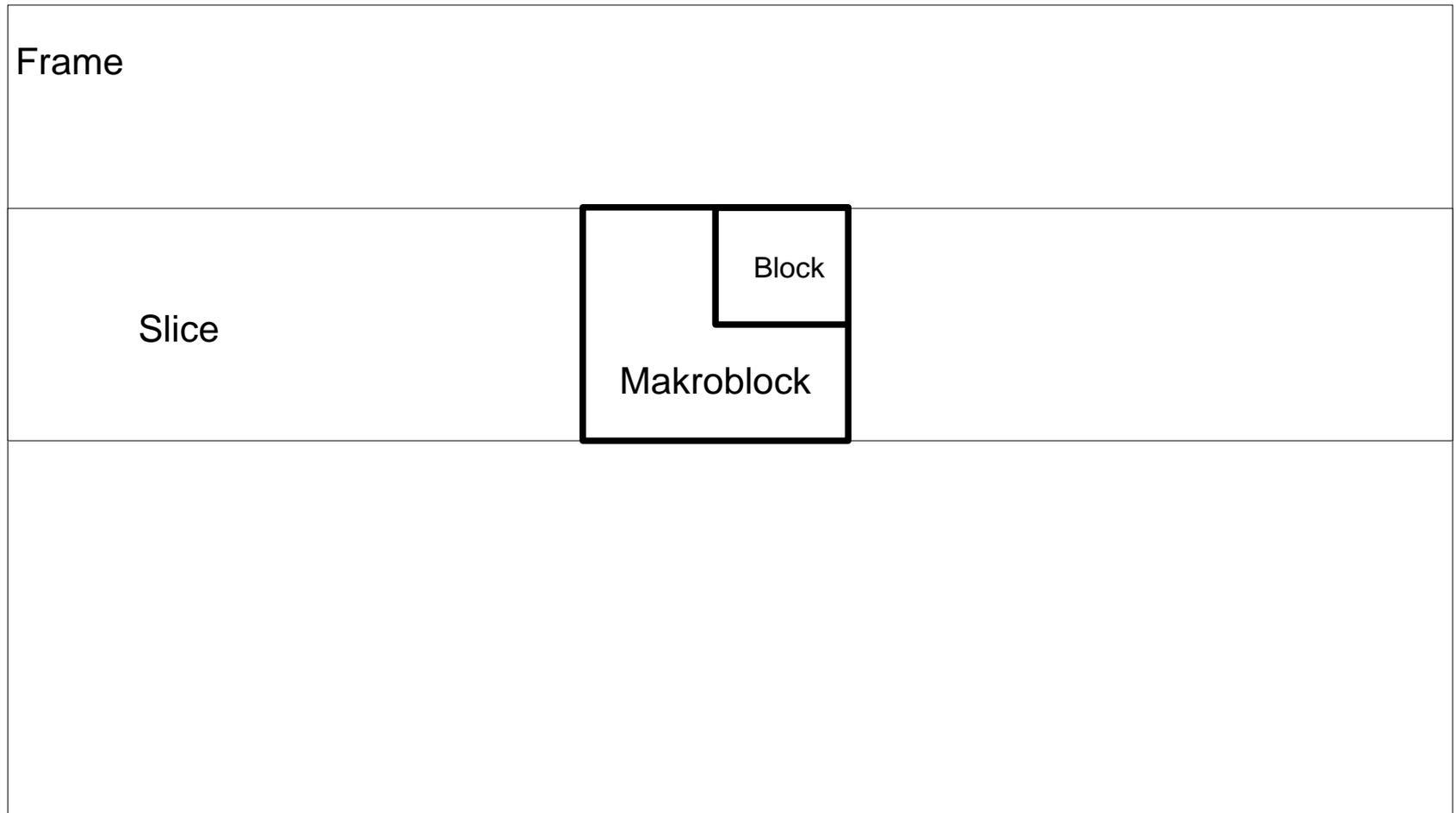
2. Technische Grundlagen

- 2.1 Geräte
- 2.2 Digitalisierung & Kodierung
- 2.3 Audiokompressionsverfahren
- **2.4 Videokompressionverfahren**

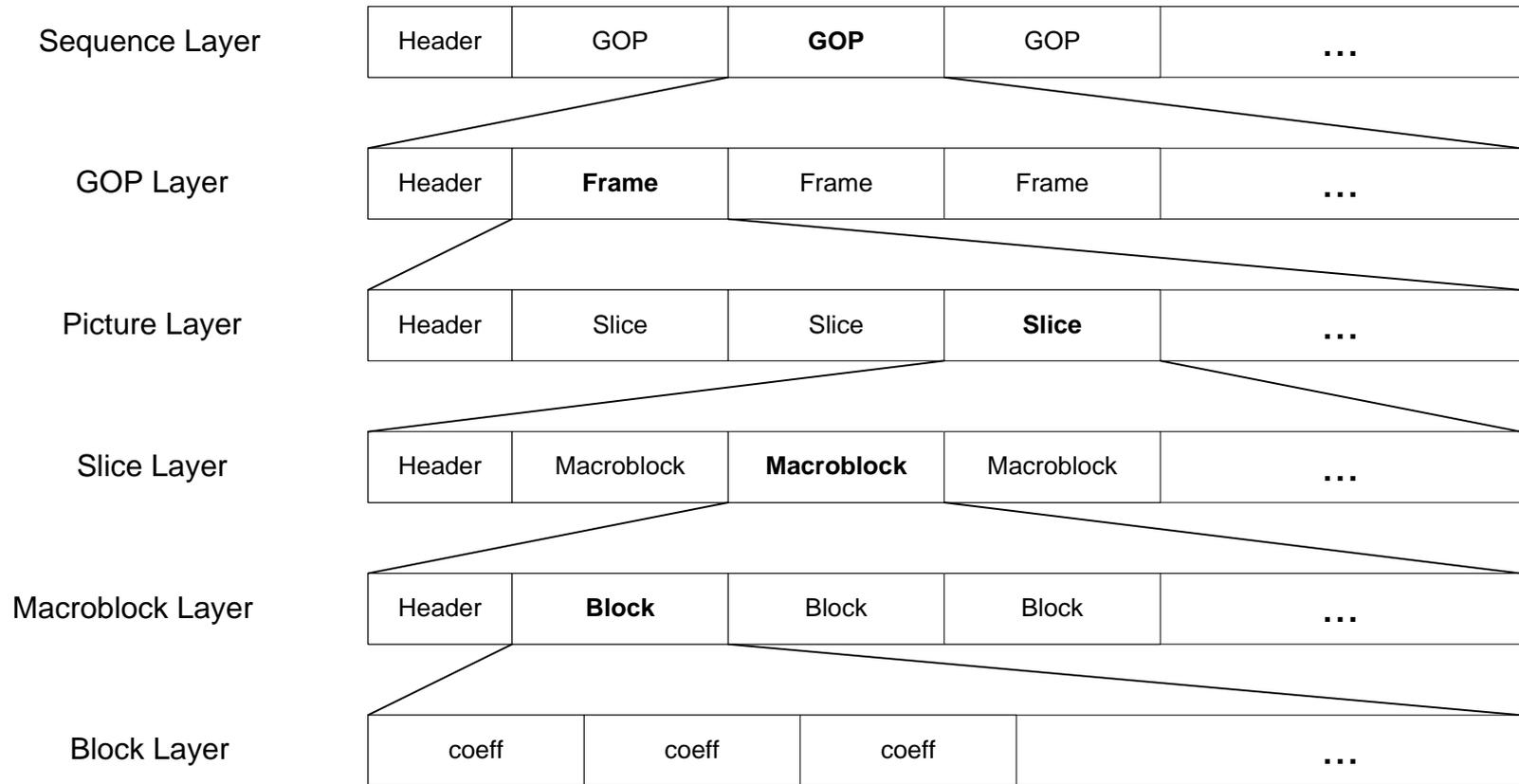
Aufbau

- Ein *Video* besteht aus N *Groups of Pictures (GOPs)*
- Eine GOP besteht aus N *Frames*
- (Ein Frame besteht aus N *Slices* (Bildausschnitt))
- Ein Slice besteht aus N *Makroblöcken* (16x16 Pixel bei MPEG2)
 - Coding Tree Units (CTU) bei HEVC (bis 64x64)
- Ein Makroblock ist unterteilt in N *Blöcke*
 - Fix 8x8 Pixel in MPEG2
 - z.B. 4x4 oder 8x8 in AVC, unterschiedlich in jedem Makroblock

Aufbau

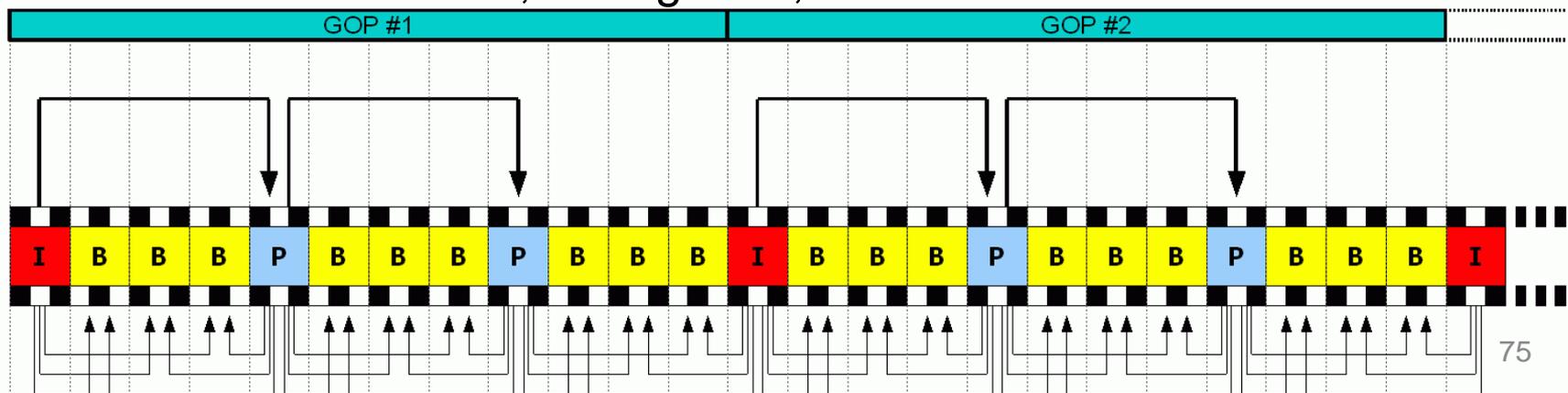


Aufbau



Group of Pictures (GOP)

- Ein Video besteht aus einer Abfolge von GOPs
- Von I-Frame bis zum nächsten I-Frame (exklusive)
- Unterschiedliche GOP-Längen je nach Verfahren
 - Bessere Kompressionsrate bei längeren GOPs aber geringere Interaktivität (Einsprungpunkte) und Robustheit
 - MPEG2: z.B. 15 oder 18
 - MPEG4: variabel, unbegrenzt, z.B. I-Frame bei Szenenwechsel



GOP – Frametypen

- **I-Frame** (Intra)
 - Intrakodiert → keine Abhängigkeiten → Einsprungpunkte
 - Keyframes, ähnlich zu einem JPEG-Bild
- **P-Frame** (Predictive)
 - Temporal abhängig zum vorherigen I/P-Frame kodiert
- **B-Frame** (Bipredictive)
 - Abhängig zu vorherigem und folgendem I/P-Frame
 - Kann nicht als Prädiktionsbasis für andere Frames verwendet werden (vor MPEG-4)
 - MPEG-4: Hierarchische B-Frames

GOP – Frameverlust

Bei **unzuverlässigen** Übertragungen über ein Netzwerk können Frames/Frameteile verloren gehen:

- I-Frame geht verloren: Die **gesamte GOP** ist bis zum nächsten I-Frame beschädigt
- P-Frame geht verloren: Die **restliche GOP** ist beeinträchtigt (verschiedene Effekte möglich, je nach Dekodierung)
- B-Frame geht verloren: Nur ein gezeigtes Bild ist beeinträchtigt

Videokodierung

- Räumliche (**spatial**) und zeitliche (**temporal**) Redundanzen vorhanden
 - Intraframe- und Interframekodierung nötig
- Übliche Vorgehensweise:
 1. Motion Compensation
 2. 2D-Transformation (DCT o.ä.)
 3. Quantisierung
 4. Verlustfreie Kompression (Huffman o.ä.)

Motion Compensation



Input

\ominus



Reference frame

=

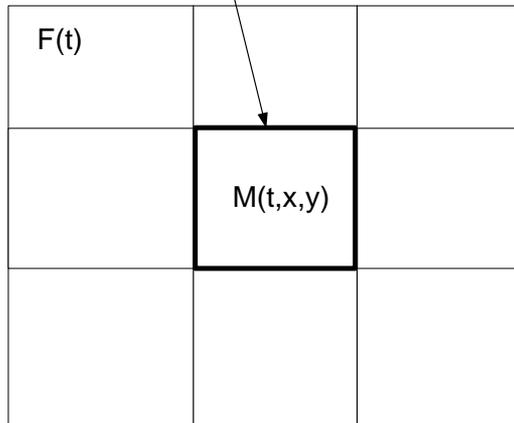


Residual

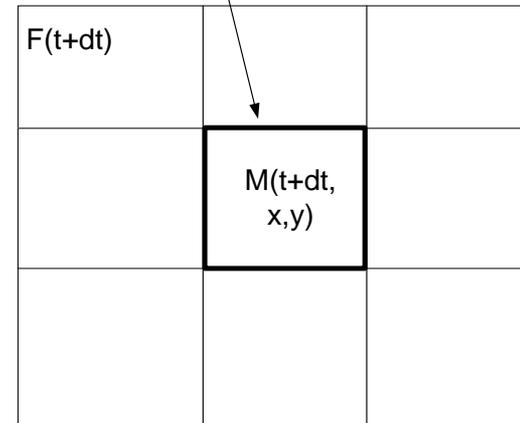
- Bewegung von Bildteilen oder der Kamera
- Zu jedem Block $\mathbf{M}(t+dt, \mathbf{x}, \mathbf{y})$
 - suche benachbarten, um (dx, dy) verschobenen Block $\mathbf{M}(t, \mathbf{x}+dx, \mathbf{y}+dy)$ im vorherigen Bild,
 - damit $\mathbf{M}(t+dt, \mathbf{x}, \mathbf{y}) - \mathbf{M}(t, \mathbf{x}+dx, \mathbf{y}+dy)$ minimal
 - Reduktion des Residuum im Vergleich zu simpler Differenz

Motion Compensation

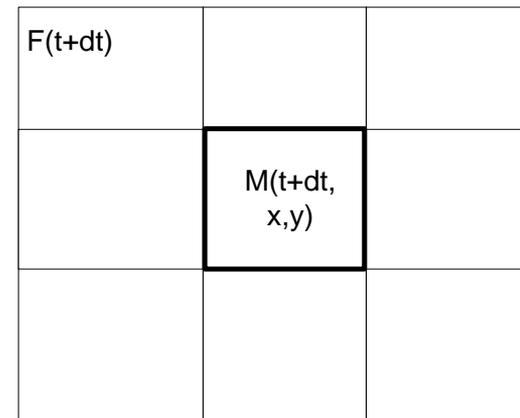
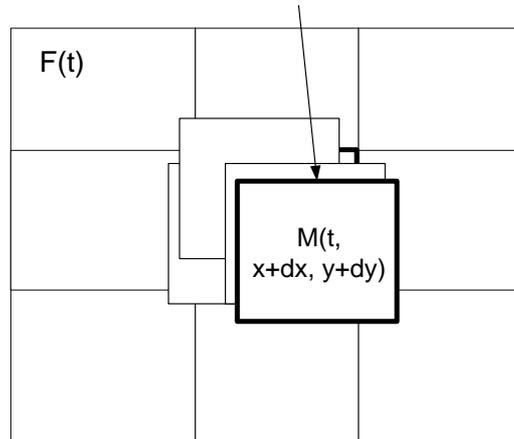
Ist $M(t,x,y)$ die beste Vorhersage für $M(t+dt, x,y)$?



Wo kommt dieser Makroblock her?



Nicht $M(t,x,y)$, sondern um (dx,dy) verschoben



Motion Compensation

- Kodiere $M(t+dt, x, y) - M(t, x+dx, y+dy)$ und zusätzlich Bewegungsvektor (dx, dy)
 - In Summe ist der Gewinn oft größer als diese zusätzlichen Daten
- **Alle möglichen Nachbarn sollten untersucht werden**
→ **Rechenaufwand enorm**
- Oft werden nicht alle möglichen (dx, dy) untersucht sondern zuerst grober Raster (alle in +/- 16 Pixel, oder „Diamond“)
- Dann verfeinerte Suche
 - Halbpixel oder Viertelpixel-Verschiebung

Videokodierungseffizienz

- Effizienz hängt vor allem von der zeitlichen Korrelation ab
 - Je weniger Korrelationen
 - desto schlechter die Komprimierung
 - **Bei vorgegebener Bitrate** → desto schlechter die Bildqualität
 - Kompressionsartefakte (z.B. „Blocking“, „Ringing“)
- Encoder setzen oft I-Frames bei Szenenwechsel

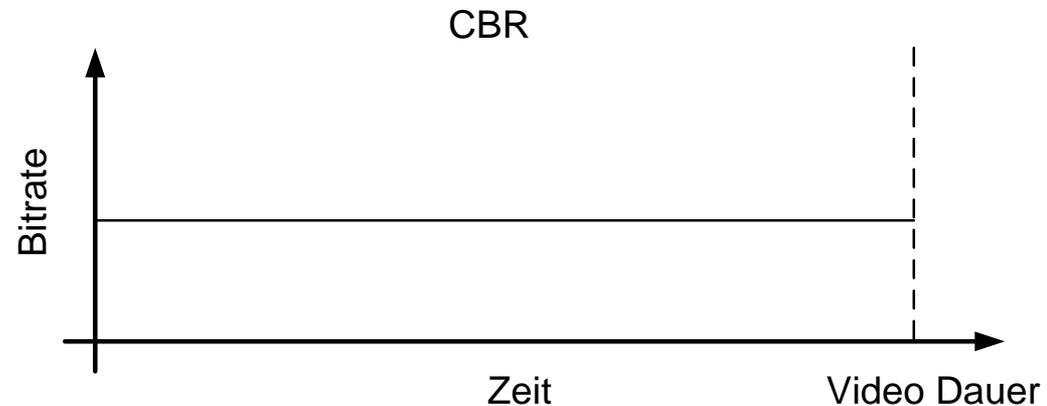
Kodierparameter

- Video Parameter
 - Bildformat
 - Hz
 - Auflösung
- Bitrate (Quantisierung, Dateigröße)
 - Balance zwischen Video Parametern und Zielmedium (z.B. Netzwerk, CD, DVD, Blue-Ray)

Konstante vs. variable Bitrate

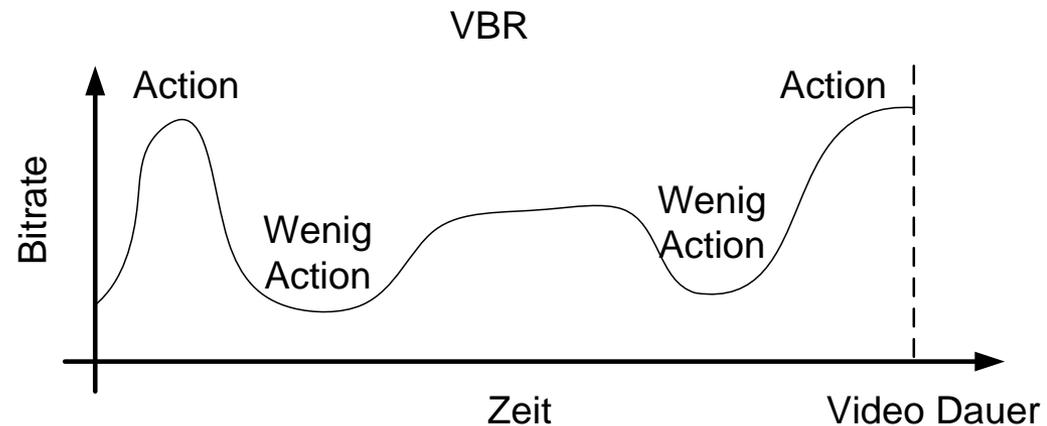
- CBR

- Zielbitrate wird genau eingehalten



- VBR

- Zielbitrate ist nur im Durchschnitt zu erreichen
- Mittlere Bitrate zeitweise über/unterschritten



MPEG

- **ISO/IEC Motion Picture Experts Group**
- Standards (Auszug):
 - **MPEG-1**: VHS Qualität mit ca. 1.4 Mbit/s (CD)
 - **MPEG-2**: TV-Qualität mit 4-10, max. 15 Mbit/s
 - **MPEG-4**
 - MPEG-4 part 2 (Simple Profile, Advanced Simple Profile)
 - MPEG-4 part 10 (Advanced Video Coding, AVC)
 - **MPEG-H Part 2**: High Efficiency Video Coding

H.26X

- **ITU Video Coding Experts Group**
- Standards (Auszug):
 - H.261
 - H.262 (= MPEG-2)
 - H.263, H.263+, H.263++
 - (H.26L)
 - H.264 (= MPEG-4 AVC)
 - H.265 (HEVC)
- ISO/IEC MPEG + ITU VCEG → Joint Video Team (JVT)

MPEG-2

- Video- und Audiokomprimierung
- Systems Layer für Multiplexing und Netzwerkübertragung
 - MPEG Transport Stream transportiert mehrere „Programme“ gleichzeitig, z.B. bei DVB-S 6-10 Programme, 38 Mbit/s gesamt
- Standard definiert nur Decoder und Bitstream Syntax
- Viele Kodier-Parameter frei wählbar
 - Parameter-Gruppen werden in *Profiles* und *Levels* geordnet
 - z.B. DVD/DVB: MP@ML; max. 15Mbit/s, max. 720x576, max. 30Hz

MPEG-4

Parts:

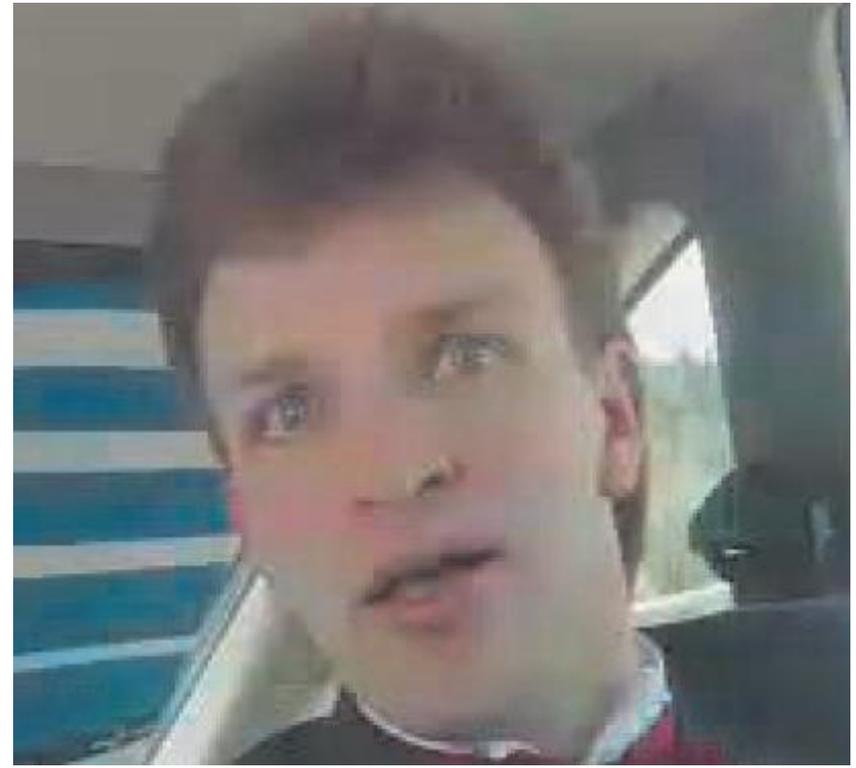
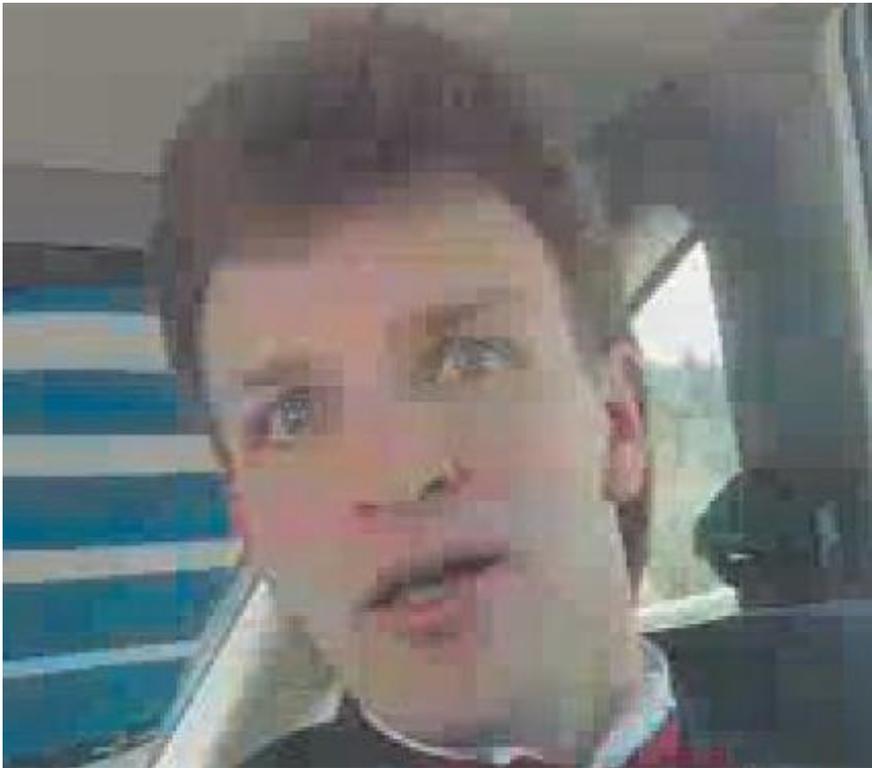
1. Systems
2. **Visual** (ASP)
3. **Audio** (AAC)
4. Conformance
5. Reference Software
6. Delivery multimedia Int. framework
7. Optimized reference software
8. Carriage over IP networks
9. Reference hardware
10. **Advanced Video Coding** (AVC, H.264)
11. Scene description engine
12. Base file format
13. DRM
14. **MP4 file format**
15. AVC file format
16. Animation framework
17. Streaming text format
18. Font compression and streaming
19. Synthesized texture stream
20. Application scene representation
21. MPEG-J (Java)
22. Open font format
23. Symbolic music representation
24. Audio and systems interaction
25. 3D graphics compression model
26. Audio conformance
27. 3D graphics conformance
28. Composite font representation
29. Web video coding
30. Timed text and other visual overlays
31. Video coding for browsers

MPEG-4 Videokodierung

- ASP (Part 2, Advanced Simple Profile)
 - Global Motion Compensation; sehr rechenintensiv
- AVC (Part 10, Advanced Video Coding)
 - Variable Blockgrößen, flexiblere GOP
 - Bis zu 16 vergangene/folgende Frames als Prädiktor (inkl. B-Frames)
 - Integer Transform statt DCT
 - Nur Ganzzahloperationen, keine Fließkommagenauigkeiten
 - Weniger Block Noise, usw.
 - Profile und Level gliedern Features und Kodierraten, z.B.
 - BD, DVB: HiP (High Profile)
 - Streaming: BP oder XP (Baseline, Extended)

MPEG-4 AVC – Deblocking Filter

- Blockkanten werden verschmiert



MPEG-4 Effizienz (bei gleicher Bitrate)

MPEG-2



MPEG-4
ASP



MPEG-4 AVC



SVC – Skalierbarkeit

- MPEG-4 AVC Erweiterung für verlustbehaftetes Streaming
- Sehr komplex, kaum verbreitet
- Spatial
 - Video mit Base Layer und n Enhancement Layers
 - Enhancement Layer differentiell zur Basis
 - Verbessern inkrementell Bildauflösung oder Bildkodiertrate
- Temporal
 - Base Layer hat nur geringe Bildrate
 - Zusätzliche Frames über Enhancement Layer

