

RBG - Seminar

MPEG 1,2,4 und 7
Eine Übersicht

Dr. Carsten Gnörlich

Übersicht

1. Der Anfang: MPEG-1
 - generelle Idee der Datenreduktion
2. Filme werden digital: MPEG-2
 - DVD, DVB, professionelle Anwendungen
3. MPEG-4
 - noch mehr Kompression und Interaktivität
4. Ausblick: MPEG-7
 - Metadaten für Audio-visuelle Daten

MPEG-1

- Grundlagen: digitale Filme
- Datenreduktion von Bild- und Tonmaterial
- Multiplexen von Bild- und Tonspuren
- Leistung / Zusammenfassung von MPEG-1

Motivation

Auflösung eines Fernsehbildes

- Deutschland: PAL B/G
- real: ≈ 400 Linien x 625 Zeilen, 50 Halbbilder/Sek.
- digital: 720 x 576 Pixel, 25 Vollbilder/Sek.

bei 3 Byte/Pixel: $3 * 720 * 576 * 25 \approx 30$ MB/Sek.

angepeilte Bandbreite bei MPEG-1

PAL B/G digital: 30 MB/Sek

MPEG-1: Anfang 80er Jahre

- Ziel: Filme auf CD speichern
- verfügbare Datenrate: 150 KB/Sek für 80 min

Ansatz:

- halbe Auflösung: 352x288, 25 Vollbilder (=7 MB/Sek)
- (Video-)Kompression um Faktor 50

Datenreduktion

Datenreduktion = verlustbehaftete Kompression

- Audio-Daten
 - psychoakustische Phänomene:
Weglassen von “unhörbaren” Informationen
 - Kodierung im Frequenzraum
- Video-Daten
 - ähnlich Datenreduktion wie bei JPEG
 - + Ausnutzen der “Trägheit” des Auges
 - + Zeitliche Redundanz in Bildfolgen

Filme bestehen aus mehreren Szenen



“Divide and Conquer”: Szenenweise kodieren



“Divide and Conquer” geht weiter



I “Intra-Frame”

komplettes Bild, vgl. JPEG

P-Frames mit mittlererer Kompression einfügen



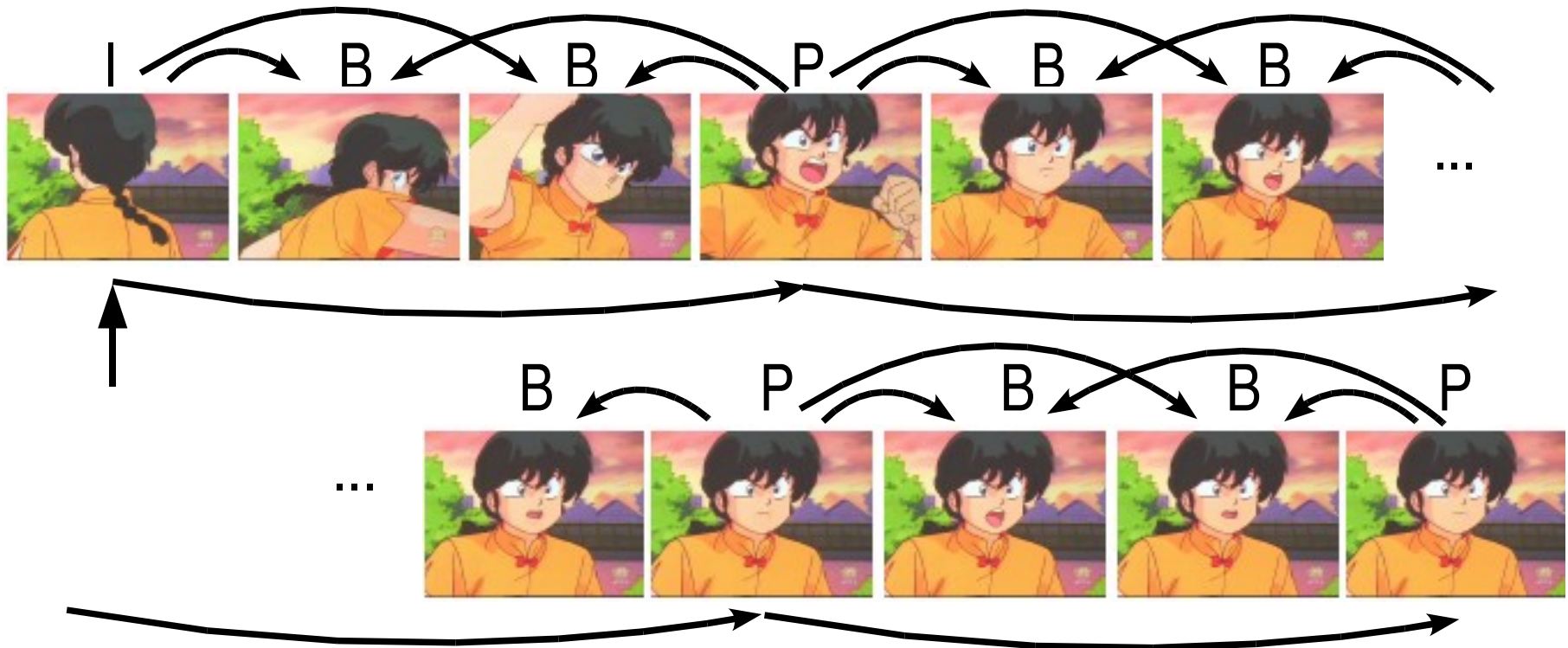
I "Intra-Frame"

komplettes Bild, vgl. JPEG

P "Predicted Frame"

Unterschiede zum Vorgänger

B-Frames mit hoher Kompression einfügen



I "Intra-Frame"

komplettes Bild, vgl. JPEG

P "Predicted Frame"

Unterschiede zum Vorgänger

B "Bidirectional Frame"

Unterschiede zu beiden Nachbarn

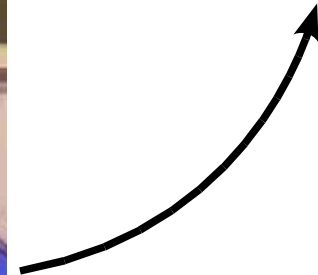
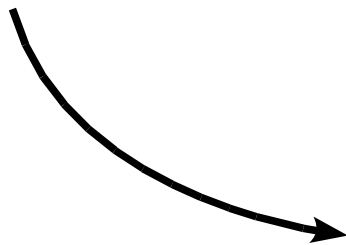


Temporale Kompression

1



3

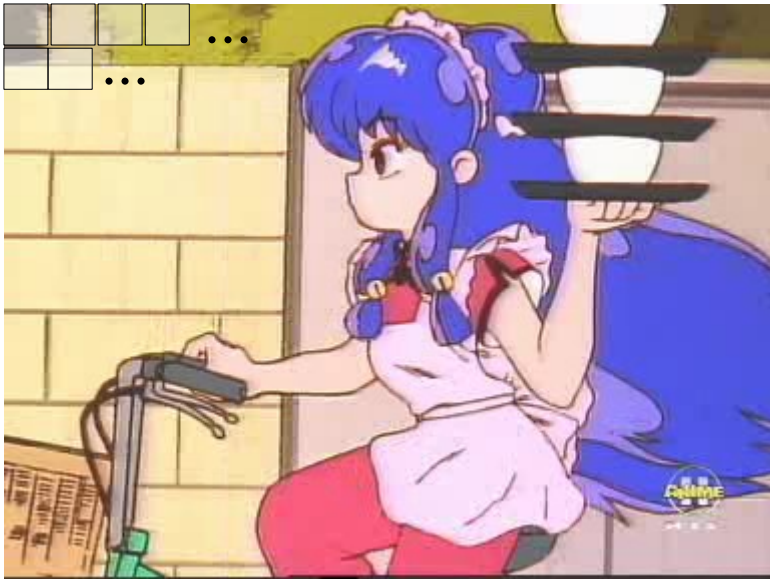


2



Aufteilung in Macroblöcke

1



3



2



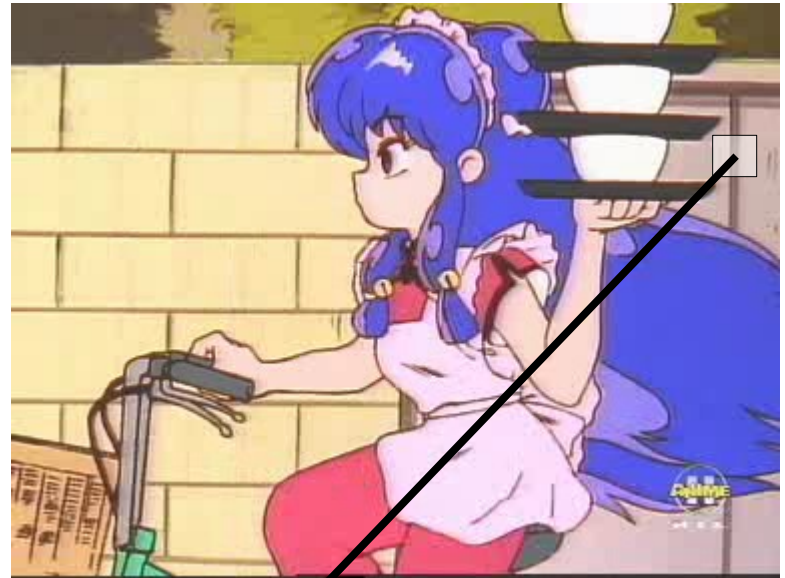


Bewegung von Macroblöcken

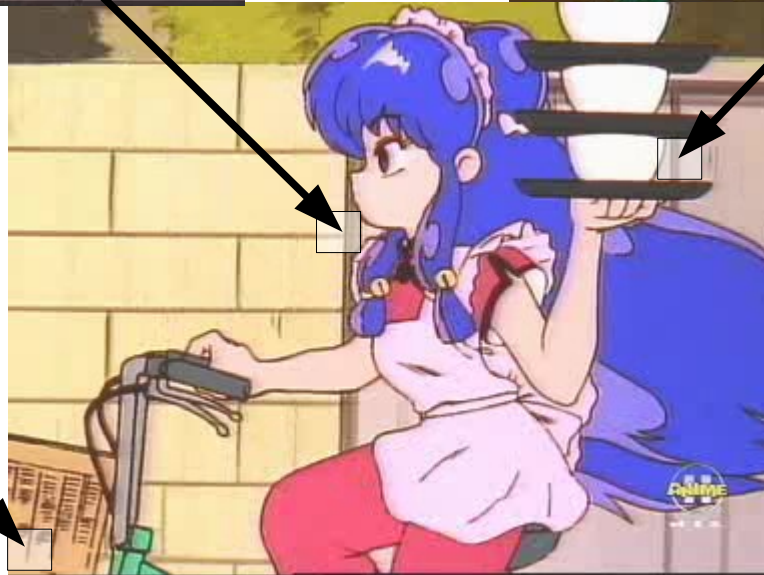
1



3



2



Audio-Kodierung (1)

Psychoakustik: “Herausrechnen” unhörbarer Anteile

- zwei Töne sind nur unterscheidbar,
wenn sie sich bzgl. Tonhöhe (Frequenz) deutlich unterscheiden
- nach einem lauten Geräusch
sind leise Töne für eine kurze Zeit nicht wahrnehmbar

Audio-Kodierung (2)

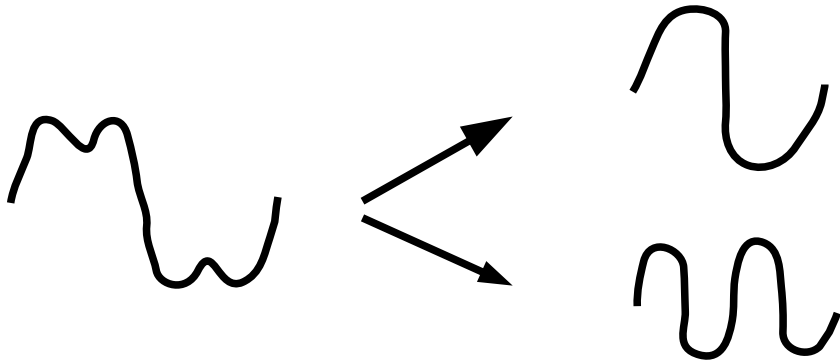
Kanalkoppelung bei Stereo

- beide Kanäle sind nicht unabhängig voneinander
- Differenzbildung zwischen beiden Kanälen nutzen

Audiokodierung (3)

Ergebnis von Fourier:

Komplexe Töne = Addition einfacher Schwingungen



- ▶ Töne in einzelne Komponenten (Frequenzen) aufteilen
- ▶ entstehende Frequenzbänder separat kodieren

Audioformate in MPEG-1

3 Audio-Layer wurden definiert

mp1: abgespeckte Version von mp2

→ Philips DCC

mp2: Das Standardverfahren mit 128kbit

→ Liveübertragung im Radio per ISDN

mp3: wer kennt es nicht ;-)

→ als Tonspur zu Video eher unüblich

MPEG-Container: Bild- und Tonspuren

CD/DVD: wahlfreier Zugriff sehr langsam

- Bild- und Tonspuren bestehen aus Datenpaketen
 - ▶ "Multiplexen" beider Spuren in eine Datei:

PTS Video PTS Video PTS Video PTS Video

PTS Audio PTS Audio PTS Audio PTS Audio



PTS Audio PTS Video PTS Audio PTS Audio PTS Video

PTS = presentation time stamp

Zusammenfassung MPEG-1

Anwendungen:

- VCD
- Philips CD-i

Gute Idee, aber noch nicht praktikabel:

- Filme hatten ungefähr VHS-Qualität
+ bessere Farbtreue, weniger Rauschen
- viele Kompressionsartefakte
- 70min Laufzeit nicht Spielfilm-Kompatibel

MPEG-2

MPEG-2: Erweiterung/Verbesserung von MPEG-1

- typische Anwendungen
- Erweiterungen gegenüber MPEG-1

MPEG-2: Anwendungen

[Erinnerung: MPEG-1 - 150 KB/sek]

- DVD / Spielfilme (1350 KB/sek)
- DVB / digitales Fernsehen (250 – 600 KB/sek über Satellit)
- Studioanwendungen (bis zu 1875 bzw. 6250 KB/sek)

Gemeinsamkeiten mit MPEG-1

Prinzipiell gleicher Aufbau:

- gleiche Datenreduktions-Verfahren
- gleiches Prinzip beim Multiplexen

MPEG-Container wurden erweitert

Container = Dateiformat im weiteren Sinne

Transportstrom

- Fernsehübertragung
- + Videotext, EPG, Fehlerkorrektur, ...

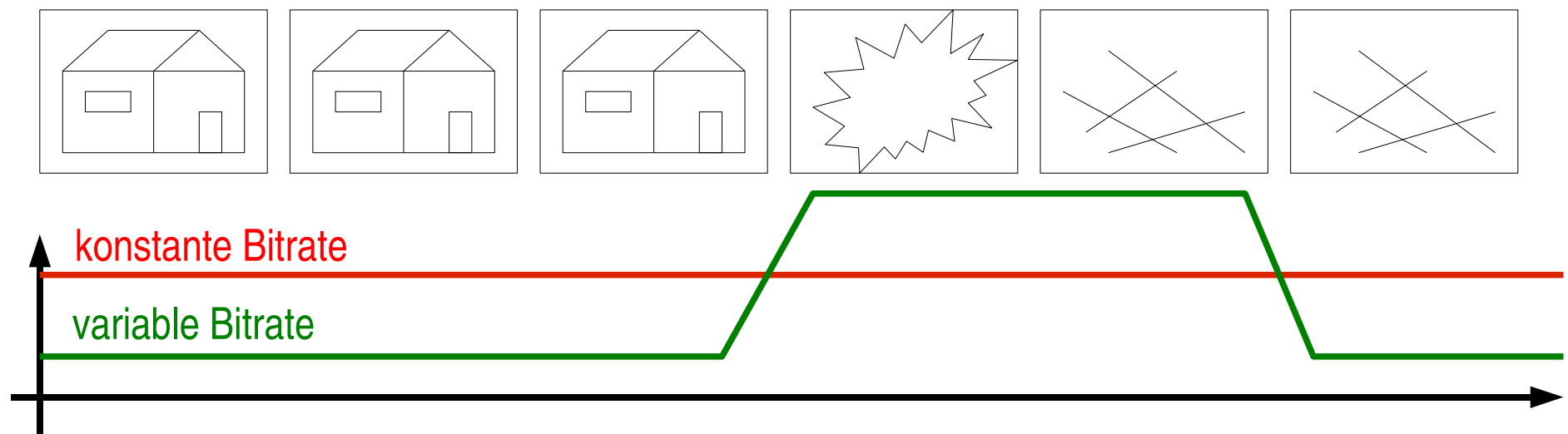
Programmstrom

- Filme auf Datenträger
- Vor-/Zurückspulen, Szenen ansteuern
- + mehrere Tonspuren, Untertitel, ...

Erweiterungen der Codecs (1)

Video-Kodierung

- höhere Auflösungen (typ.: 720 x 576) und Datenraten
- Halbbilder, anamorphe 16:9-Optimierung
- variable Bitraten



Erweiterungen der Codecs (2)

Audio-Kodierung

- 5.1-Mehrkanalkodierung (rückwärtskompatibel zu mp1...mp3)
- 5.1 Mehrkanalkodierung AAC (nicht rückwärtskompatibel)
- darauf aufbauend: Dolby Digital, DTS (digital theater sound)

Zusammenfassung MPEG-2

Evolution statt Revolution

- dank größerer Datenraten gute Qualität
- zur Zeit das führende Format (DVD, DVB)

MPEG-4

Nachfolgeformat von MPEG-2

- beabsichtigte Anwendungen
- Änderungen gegenüber MPEG-2
- Microsofts Fehltritt
- Aktuelle Implementierungen
- Welchen Codec sollte man nehmen?

Ziele von MPEG-4

Ursprünglich:

- Video auf Geräten mit wenig Bandbreite / Rechenleistung (z.B. Handies, Webcams, Video-Telefone)

Letztendlich:

- Universelles Multimedia-Format
Vorbild: Apple Quicktime

Unterschiede zu MPEG-2

MPEG-2: 2 Container-Formate+ Video/Audio-Codec

MPEG-4:

- Verschiedene Container-Formate
- H.263 Standard für Video-Kompression
- alte (mp3,AAC) und neue Audio-Codecs (u.a. eine lossless-Variante)

- VRML-ähnliche Szenenbeschreibungssprache
- Digital Rights Management

Was bringt H.263?

Speziell für geringe Bandbreiten angepaßt

- prinzipiell gleiches Vorgehen wie bei den Vorgängern
- starke zeitliche Kompression (Folgen gleicher Bilder)
- sehr rechenintensive Vor- und Nachfilterung

Qualität

- das pure dekomprimierte Bild sieht fürchterlich aus
- die Nachfilterung beseitigt die meisten Artefakte
- ▶ 2h - Spielfilm auf eine CD komprimieren “geht so”

Szenenbeschreibungssprache (1)

Aufteilung der Szenen in

natürliche Objekte

- Video-Objekte (z.B. Personen)
- Audio-Objekt (z.B. Stimme des Sprechers)

→ Objekte können unabhängig kodiert / positioniert werden

künstliche Objekte (VRML)

- künstliche Grafiken und Geräusche einbinden

→ Texte, 2D-Grafik, 3D-Polygon-Grafik mit Texturen

= Knöpfe, Navigationen, animierte Cartoon-Figuren als Sprecher

Szenenbeschreibungssprache (2)

Anwendungen

- Markup zum Editieren von Szenen und Komponenten
- Komplexe Szenen aus Objekten komponierbar
 - Benutzer kann Szene interaktiv drehen und manipulieren
 - funktioniert ähnlich wie bei VRML-Browsern
- Nachgekartet: MPEG-J - Java-API im Player

Tolle Sache?

(Übersetztes) Zitat von

<http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>

F: “Wann wird MPEG-4 MPEG-2 ersetzen”?

A: “Bestimmt nicht in naher Zukunft, denn es wurden Milliarden Dollars in MPEG-2 investiert.

Das MPEG-Komitee sind die letzten, die daran interessiert sind, daß diese Investitionen wertlos werden.”

→ Philips CD-i in 3D?

Was bleibt?

MPEG-4 ist heute hauptsächlich auf den H.263-Codec reduziert

... und das kam so....

Microsofts Fehltritt

Microsoft MPEG-4 v3 Codec

- ursprünglich nur für Streaming-Format ASF gedacht
- wurde mit einer Beta des MediaPlayers mitgeliefert

kleiner Hack entfernte Ausgabebeschränkung im ASF-Format

→ Codec beliebig verwendbar

→ jede Anwendung, die .avi erzeugt kann ihn verwenden

→ Beginn von DivX (MPEG-4 im AVI-Container!)

“Kompletter Spielfilm auf eine CD”

Zeitlinie

DivX 3.11: “gehackter Microsoft-Codec”

DivX 4: erste Neuimplementierung

- zuerst OpenSource, dann zurückgezogen und patentiert

DivX 4/5: MPEG-4 im AVI-Container

- werden in einigen Hardware-Playern unterstützt

Aktuell: DivX 6.2 (Windows, Linux, Mac OS X)

- eigenes Container-Format .divx

Weitere MPEG-4 - Ableger

XviD

- Fork des kurzzeitig freien DivX 4
- OpenSource, aber durch Patente eingeschränkt
- ziemlich aktive Community

FFmpeg (OpenSource, ähnliche Patent-Problematik)

3ivx (kommerziell, echter mp4-Container: streamfähig)

HDX4 (Jomingo, kommerziell)

Nero Digital Video (H.263, H.264, ...)

Gehts auch ohne Patente?

- Ogg – Container-Format
- Vorbis – Audio-Codec
- Theora/Tarkin – Video-Codex (DCT / Wavelets)

- freie Software
- Ziel: H.263 (“MPEG-4”) und RealVideo ersetzen

(www.vorbis.com, www.theora.org)

Stand der freien Implementierungen

Vorbis

- wird ernstzunehmende Konkurrenz zu .mp3
- für Sound-Komprimierung in Spielen sehr beliebt
- es gibt bereits vorbis-Player in Hardware

Theora

- Alpha release, Format ist fest
- vielversprechend, aber bisher nur Software-Player

Tarkin

- proof of concept, zugunsten von Theora zurückgestellt (schade!)

Welchen Codec sollte man nehmen (1)

Ziel: Filmmaterial für 30 Jahre aufheben

gebrannte CD/DVD überleben keine 30 Jahre

- regelmäßig prüfen und umkopieren
- Speichern mit Redundanz: PAR2, ICE ECC, dvdisaster

Filme in 30 Jahren noch abspielbar?

- MPEG-2: ziemlich sicher (vgl: PAL ex. seit 1967)
- MPEG-4: AVI-Container, halb-legal: wahrscheinlich nicht
- Ogg: freie Software, damit theoretisch ja

Welchen Codec sollte man nehmen (2)?

Kodierung MPEG2 → MPEG4 / Ogg lohnt nicht:

- Kodierungsfehler von MPEG2 werden nicht weggerechnet, sondern besonders sorgfältig neu kodiert
 - ▶ starker Qualitätsverlust
 - ▶ geringer Speicherplatzbedarf in Zeiten von (HD)DVD/BluRay kein Argument mehr

MPEG-4 / Ogg für eigene Produktionen:

- wenn Hardware-Player verzichtbar ist
- bei analogem oder DV-Ausgangsmaterial



MPEG-7

kurze Übersicht

Metadaten, Metadaten, Metadaten!

Zurück zu den Wurzeln

► nur Video- und Audio-Inhalte

Aber alles mit Metadaten unterlegt!

- ein Standard zur Beschreibung audio-visueller Daten
- XML-Markup
- Verlustlose Transformation XML ↔ binäres Format

Typen von Meta-Daten

Äußere Erscheinung

- Video: Farben, Texturen, Formen
- Audio: Wellenformen, Frequenzspektrum

Inhaltliche Merkmale

- Objekte in Szenen (Personen, Gegenst., Sprache, Instrumente)
- Thema der Szenen (spielende Hunde, Autorennen, ...)

Weitere Metadaten

- Erstellungszeitpunkt und -Ort
- DRM und FSK

Beispielanwendungen

MPEG-7 soll *der* Standard für Metadaten werden

- Suchmaschinen und Datenbanken für audio-visuelle Inhalte
- Verschlagwortung von audio-visuellen Inhalten
- automatische Merkmalsextraktion
- intelligente Endbenutzer-Geräte
("Zeig mir nochmal die Szene, wo ...")

Dankeschön

Vielen Dank fürs Zuhören !