Digitale Videoarchivierung: Best practices für Semi-Pro bis Nationalarchiv

Peter Bubestinger

23. Mai 2014

Über mich

Peter Bubestinger

- Studierte Medieninformatik an der TU-Wien
- Praxiserfahrung mit professionellen Archiven seit 2002:
  - ORF (National broadcaster, Austria)
  - VoV (National broadcaster, Vietnam)
  - RTV (National broadcaster, Slovenia)
  - SRTC (National broadcaster, Sudan)
  - Fonoteca Nacional (Mexico)
  - Memnon Archiving Services (Belgium)
  - SRF (Sweden), YLE (Finland), SRR (Romania), ...
- Arbeite mit GNU/Linux Systemen seit 2001
- Angestellter in der Videoabteilung der Österreichischen Mediathek
- Koordinator von FSFE Aktivitäten in Österreich

"Open Source" im Archivbereich:

- Videodigitalisierung fast ausschließlich mit GNU/Linux
- Neuer Massenspeicher: Open Hardware, GNU/Linux
- Zunehmend Interesse, im Archivbereich Freie Software einzusetzen
- Nicht weil gratis, sondern wegen "use, study, share & improve"
- Extrem (kosten-)effiziente Lösungen möglich
- Und: Freie Software + Offene Formate = "Virtually immortal"

Digitales Video: Tech-Details

Das Wichtigste gleich am Anfang!

Die "Dreifaltigkeit":

- Container
- Videocodec
- Audiocodec

Das Wichtigste gleich am Anfang!

Die "Dreifaltigkeit":

- Container
- Videocodec
- Audiocodec

Aussagen wie "Die Videos sind im Flashformat" oder "Die Kamera speichert AVI-Videos" sagen also nur etwas über den Container aus. Und der ist oft noch das "Harmloseste" bei digitalen Videofiles...

fps: Frames Per Second

Übliche Frameraten:

- 24fps: Film
- 25fps: PAL/SECAM (Europa)
- 29.97fps: Eigentlich "30000/1001". Kommt von NTSC (USA)
- 23.98fps: Film auf NTSC
- 38.42fps: WTF? Ja, bei born-digital Videos kann alles vorkommen :)

`http://vanillavideo.com/blog/2012/history-frame-rates-why-speeds-vary`

Zeilenabtastverfahren

Schönes, old-school "Ingenieurs-Sprech"

Im Englischen heißt es "Scanning Method".

Halbbild oder Vollbild?

progressive: Ein Bild (Frame) = ein Vollbild.
Film als Quelle zB ist immer progressive.

interlaced: 2 Halbbilder (Field) pro Frame.
Kommt aus der Ära von CRT-Röhren und Fernsehen.

Doppelte zeitliche Auflösung bei halber vertikaler Auflösung. Rundere Bewegungsabläufe (zB Sportübertragung).
Bsp: PAL Fernsehen hat 25 Frames pro Sekunde (fps), aber interlaced, also 50 Fields pro Sekunde.

Vorsicht: Programme "deinterlacen" zunehmend automatisch.

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Pixelauflösung: “width * height”

Mindestens: 720x576 (PAL-SD)

Achtung: Breitbildformate werden oft “anamorph” aufgenommen!

Bsp: HDV mit 1440x1080, statt 1920x1080

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Pixelauflösung: “width * height”
Mindestens: 720x576 (PAL-SD)
Achtung: Breitbildformate werden oft “anamorph” aufgenommen!
Bsp: HDV mit 1440x1080, statt 1920x1080

GOP: Group Of Pictures. Nicht jedes Bild ist alleine “lebensfähig”.
Größere GOP=kleineres File,
Kleinere GOP=stabileres File

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Pixelauflösung: “width * height”
Mindestens: 720x576 (PAL-SD)
Achtung: Breitbildformate werden oft “anamorph” aufgenommen!
Bsp: HDV mit 1440x1080, statt 1920x1080

GOP: Group Of Pictures. Nicht jedes Bild ist alleine “lebensfähig”.
Größere GOP=kleineres File,
Kleinere GOP=stabileres File

Bitrate: Nur bei lossy-Codecs relevant. Aber dort sehr wichtig!

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Pixelauflösung: “width * height”
Mindestens: 720x576 (PAL-SD)
Achtung: Breitbildformate werden oft “anamorph” aufgenommen!
Bsp: HDV mit 1440x1080, statt 1920x1080

GOP: Group Of Pictures. Nicht jedes Bild ist alleine “lebensfähig”.
Größere GOP=kleineres File,
Kleinere GOP=stabileres File

Bitrate: Nur bei lossy-Codecs relevant. Aber dort sehr wichtig!

Framerate: Achtung: Manche Kameras/ADCs inserten/droppen/interpolieren Frames

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Farbraum: YUV, RGB oder XYZ? Linear oder logarithmisch?

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Farbraum: YUV, RGB oder XYZ? Linear oder logarithmisch?

Subsampling: Eine Art Kompression, da Farbinformation mit geringerer Pixelauflösung gespeichert wird.
Bsp: *4:2:2* oder *4:2:0*.
4:4:4 bedeutet “Kein Subsampling”.

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Farbraum: YUV, RGB oder XYZ? Linear oder logarithmisch?

Subsampling: Eine Art Kompression, da Farbinformation mit geringerer Pixelauflösung gespeichert wird.
Bsp: *4:2:2* oder *4:2:0*.
4:4:4 bedeutet “Kein Subsampling”.

Bits-Per-Component (bpc): Anzahl der Bits pro Komponente des Farbraums.
Components: Y/U/V, R/G/B, etc.

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Farbraum: YUV, RGB oder XYZ? Linear oder logarithmisch?

Subsampling: Eine Art Kompression, da Farbinformation mit geringerer Pixelauflösung gespeichert wird.
Bsp: *4:2:2* oder *4:2:0*.
4:4:4 bedeutet “Kein Subsampling”.

Bits-Per-Component (bpc): Anzahl der Bits pro Komponente des Farbraums.
Components: Y/U/V, R/G/B, etc.

“Auflösung”
Im weitesten Sinne. . .

Mehrere Faktoren:

Farbraum: YUV, RGB oder XYZ? Linear oder logarithmisch?

Subsampling: Eine Art Kompression, da Farbinformation mit geringerer Pixelauflösung gespeichert wird.
Bsp: *4:2:2* oder *4:2:0*.
4:4:4 bedeutet “Kein Subsampling”.

Bits-Per-Component (bpc): Anzahl der Bits pro Komponente des Farbraums.
Components: Y/U/V, R/G/B, etc.

Standard bei Digitalkameras (auch HD) ist meistens YUV, 4:2:0 Subsampling und 8bpc linear.
Alles Andere ist oft “schlecht kartografiertes Gebiet”.

Container

Containerformate

Dateiendung = Containername

- AVI (.avi), Matroska (.mkv), Quicktime (.mov), MPEG-4 (.mp4,.m4v)
- WebM (.webm), Flash (.flv)
- MPEG (.mpeg, .mpg), VOB (.vob)
- MXF (.mxf), ISO9660/UDF (.iso), DV (.dv)

Für Audio gibt es andere Container: .wav, .aiff, .mp3, .ogg, .mka, etc.

Codecs

Codecs

Was ist ein "Codec"?

Steht für "(en)COder / DECoder".

"Codec" ist das Format in dem der effektive Inhalt/Content (Audio/Video/etc) im Container abgelegt wird.

Im digitalen Videobereich sind die meisten Codecs Kompressionsmethoden - meist verlustbehaftet (*lossy*), können aber auch verlustfrei (*lossless*) oder sogar unkomprimiert sein (*uncompressed*).

Codecs - Kompressionsarten

lossy vs. lossless vs. uncompressed

Verlustbehaftet (lossy):

- Immer gewisser Informations-/Qualitätsverlust
- Beste Kompression.
- Kann gute Bild-/Tonqualität bieten (je nach Codec und Einstellung)
- Bei jeder Bearbeitung oder Umwandlung in ein anderes lossy-Format entstehen Qualitätsverluste und Kompressionsartefakte (aka “Generation Loss”).
- Bei Video: Standard im Consumer- und Semi-Pro-Bereich
- Decoding schneller als lossless
- Rechenintensiv

Codecs - Kompressionsarten

lossy vs. lossless vs. uncompressed

Verlustfrei (lossless):

- *Kein* Informations-/Qualitätsverlust
- Wesentlich (!) größere Files als bei lossy-Komprimierung.
- Dafür geht bei Bearbeitungen und Umwandlung in andere lossless-Formate *keine* Qualität verloren.
  Egal wie oft migriert wird.
- Bei Video: Seit Kurzem auch für Consumer leistbar machbar
- Decoding meist langsamer als lossless
- Rechenintensiv

Codecs - Kompressionsarten

lossy vs. lossless vs. uncompressed

Uncompressed:

- Ebenfalls verlustfrei.
- Die Daten werden hier gar nicht komprimiert sondern quasi "roh" abgelegt.
- Im Audiobereich ist unkomprimiert Standard (.wav).
- Bei Video sind die Datenmengen derzeit nur sehr kostspielig handzuhaben.
  zB: PAL SD Video (YUV 4:2:2) = ca. 1.86 GiB/Minute
- Wenigste Rechenlast, dafür aber sehr viel Daten-I/O.

Codecs
Größenvergleiche

Video (1 Min. PAL-SD yuv422):

uncompressed : 1186.5 MiB

FFV1 : ca. 400 MiB (lossless/inhaltsabhängig)

JPEG2000-lossless : ungefähr gleich wie FFV1

Audio (1 Min. 44.1 kHz/16bit stereo):

PCM/WAV : ca. 10 MiB

FLAC : ca. 6.7 MiB (lossless/inhaltsabhängig)

MP3/OGG-Vorbis : bei 128kbps ca. 1 MiB

Audio ist vom Platzbedarf im Vergleich zum Video fast vernachlässigbar.

Videocodecs
Überblick

Verlustbehaftet:

MPEG-2: In verschiedenen Varianten. zB Video-CD oder DVD

MPEG-4 (SP): MPEG-4 Simple Profile. Besser bekannt unter "DivX" oder "XviD".

MPEG-4 (AVC): MPEG-4 Advanced Video Coding. Besser bekannt unter "h264" oder "x264".

ProRes: Proprietäres Apple Format. Sehr populär im Editing/Produktionsbereich.

JPEG2000: Hauptanwendungsgebiet: Digital Cinema Package (DCP)

VP8: Teil der WebM Spezifikation. Hauptanwendung: Webvideos.

Videocodecs
Überblick

Verlustfrei:

h264-lossless: h264 kann auch lossless eingestellt werden. h264-lossy ist weit verbreitet, aber lossless ist schlecht/kaum unterstützt.

JPEG2000-lossless: JPEG2000 gibt es ebenfalls lossy *und* lossless.

FFV1: "FFmpeg Video codec 1". Reiner lossless-Codec. Komplett offenes Format.

HuffYUV: Huffman-basiert. Sehr schnell. Komprimiert aber nicht so gut.

Uncompressed: Unkomprimiert ist ebenfalls verlustfrei. Hier gibt es je einen Codec für unterschiedliche Farb- und Subsamplingvarianten.

Audiocodecs
Überblick

Verlustbehaftet:

- MPEG-1 Audio Layer II (mp2)
- MPEG-1 Audio Layer III (mp3)
- Vorbis
- MPEG-4 Advanced Audio Codec (aac)

Verlustfrei:

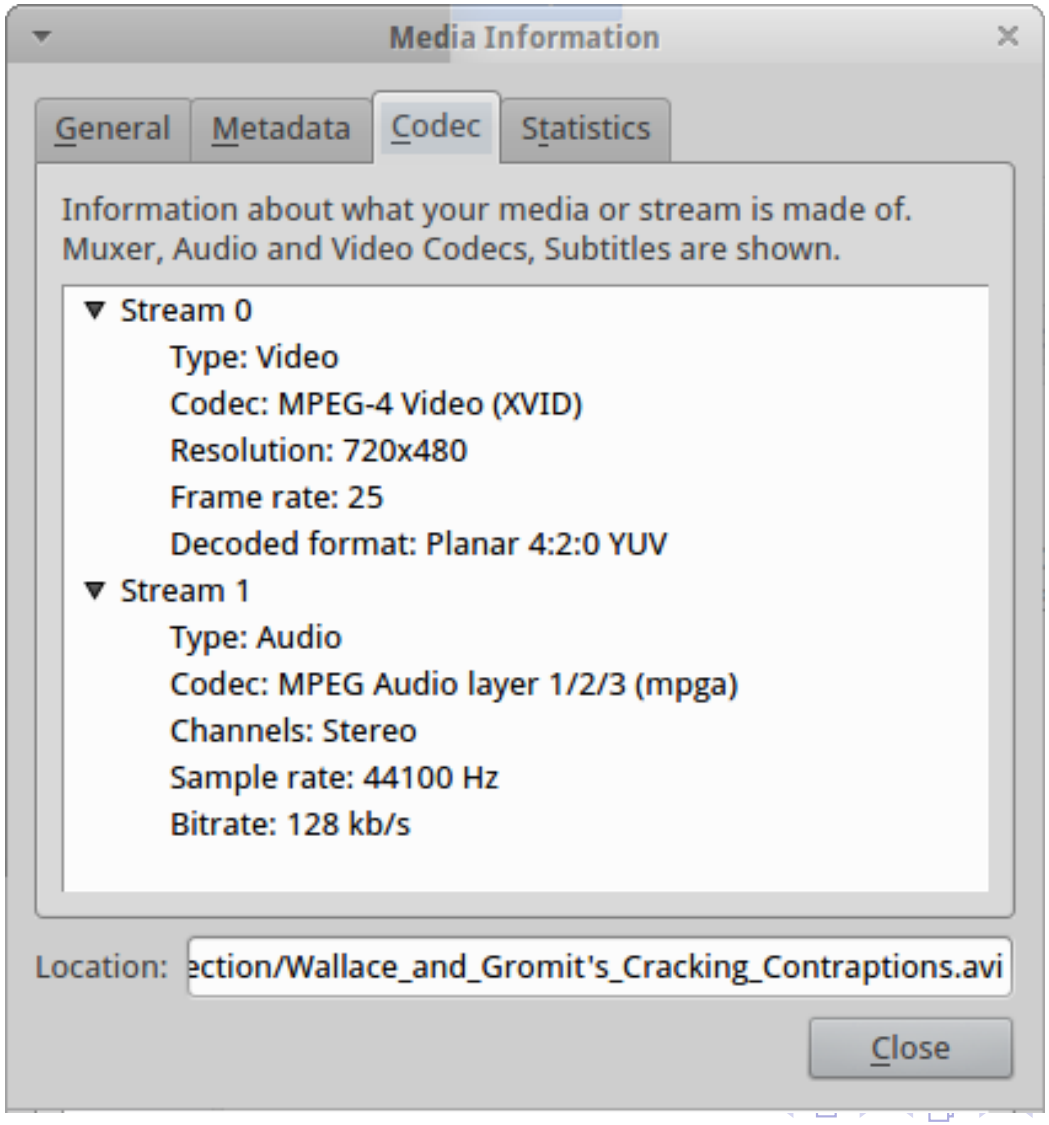
- PCM (.wav, .aiff)
- Free Lossless Audio Codec (FLAC)

Archivieren

Datenformate

Was ist mein Video eigentlich jetzt genau?

Tools: zB VLC, MediaInfo, ffprobe



Wunschliste:

- Verlustfrei (Bild/Ton)
- Farbraum und Subsampling exakt erhalten ("pix_fmt")
- Metadaten erhalten
- Auf jeden Fall ein offenes Format. Im besten Fall standardisiert. Auch bei Standards: Vorsicht vor proprietären, geschlossenen Implementierungen (Hardware, Software, Kamera).
- Erhaltbarkeit/Zugänglichkeit
- Sourcecode archivieren ("git clone") = Abspielgerät + Bauplan mit-archivieren
- Je simpler, desto besser

Datenformat(e) für die Langzeitarchivierung

Was kann ich in 'x' Jahren wieder öffnen?

Entscheidungen:

- Ein File für alles, oder ein "Ordnerpaket"?
- Formatobsoleszenz?
- Abhängigkeiten (Hardware, Software, Lizenzen)?
- Storagekosten?

Datenformat(e) für die Langzeitarchivierung

Was kann ich in 'x' Jahren wieder öffnen?

Entscheidungen:

- Ein File für alles, oder ein "Ordnerpaket"?
- Formatobsoleszenz?
- Abhängigkeiten (Hardware, Software, Lizenzen)?
- Storagekosten?

Pragmatische Lösung für Langzeitarchivierung: FFV1/PCM in AVI/MOV/MKV.
Im Privatbereich zwar machbar, aber wahrscheinlich doch (noch) zu teuer wegen Platzbedarf.

Speichermedien

Oder: “Warum CDs/DVDs/BluRay keine gute Wahl sind. . .”

Optische Medien:

- Gebrannte optische Datenträger: Schlechte physische Haltbarkeit. Sehr temperatur- und lichtempfindlich.
- Als Videoträger: Nur lossy Codecs und immer Subsampling

Speichermedien

Oder: “Warum CDs/DVDs/BluRay keine gute Wahl sind. . .”

Optische Medien:

- Gebrannte optische Datenträger: Schlechte physische Haltbarkeit. Sehr temperatur- und lichtempfindlich.
- Als Videoträger: Nur lossy Codecs und immer Subsampling

Festplatten:

- Offline Lagerung: Haltbarkeit beschränkt
- Schwachstellen: Bewegliche Teile, Elektronik
- Vorteil: Gutes Preis/Platz-Verhältnis, Daten schnell verfügbar

Speichermedien

Oder: “Warum CDs/DVDs/BluRay keine gute Wahl sind. . .”

Optische Medien:

- Gebrannte optische Datenträger: Schlechte physische Haltbarkeit. Sehr temperatur- und lichtempfindlich.
- Als Videoträger: Nur lossy Codecs und immer Subsampling

Festplatten:

- Offline Lagerung: Haltbarkeit beschränkt
- Schwachstellen: Bewegliche Teile, Elektronik
- Vorteil: Gutes Preis/Platz-Verhältnis, Daten schnell verfügbar

Bandlaufwerk (zB LTO):

- Langsamer als Festplatten
- Vorteil: Medium lang haltbar
- Herausforderung: Passendes Laufwerk + Software + Bandfilesystem

Re-Multiplexing (remuxen)
Den Container “umpacken” ohne neu zu enkodieren

Wozu remuxen?

- Bei born-digital immer zu empfehlen: Fehler früh erkennen und/oder vermeiden.
- Kein Qualitätsverlust. Auch auch bei lossy-Codecs.
- Wenn geht, Originalfile aufheben.
- Vorsicht: Metadaten können dabei leider leicht verloren gehen.

Re-Multiplexing (remuxen)
Den Container “umpacken” ohne neu zu enkodieren

Wozu remuxen?

- Bei born-digital immer zu empfehlen: Fehler früh erkennen und/oder vermeiden.
- Kein Qualitätsverlust. Auch auch bei lossy-Codecs.
- Wenn geht, Originalfile aufheben.
- Vorsicht: Metadaten können dabei leider leicht verloren gehen.

Beispiele:

```
$ ffmpeg -i video.avi -vcodec copy -acodec copy output.avi
$ ffmpeg -i video.avi -c copy output.avi
```

Prüfsummen zur Integritätssicherung

Überblick

Algorithmus: Üblich ist MD5. Ausreichend für Integritätschecks und schneller als zB SHA.

Dateiprüfsummen: Eine Prüfsumme für ein File.

Segmentprüfsummen: Mehrere Prüfsummen pro File. Eine pro Segment von x Bytes.

Content-Prüfsummen: Prüfsumme(n) über den Inhalt, ohne Metadaten oder Container.

Prüfsummen zur Integritätssicherung
Dateiprüfsummen

Eigenschaften / Anwendungsfälle:

- Leicht zu erstellen und überprüfen
- Ändern sich, sobald zB Metadaten im Container aktualisiert werden
- File-Integritätscheck bei Speichermigration
- File-Integritätscheck bei Übergabe/Transport

Prüfsummen zur Integritätssicherung
Dateiprüfsummen

Eigenschaften / Anwendungsfälle:

- Leicht zu erstellen und überprüfen
- Ändern sich, sobald zB Metadaten im Container aktualisiert werden
- File-Integritätscheck bei Speichermigration
- File-Integritätscheck bei Übergabe/Transport

Beispiel:

```
$ md5sum *.avi > MD5SUMS
```

Prüfsummen zur Integritätssicherung
Content-Prüfsummen (framemd5)

Eigenschaften / Anwendungsfälle:

- Rechenaufwendiger, weil Content erst “entpackt” werden muss
- framemd5 Video: Prüfsummen werden von den unkomprimierten Bildern gemacht
- framemd5 Audio: Prüfsumme über gewisse Anzahl von Samples
- Integritätscheck bei Format- und Codecmigration, sowie Containerupdates
- Bei worst-case Szenarien: verlustfreie Wiederherstellung der Information möglich

Prüfsummen zur Integritätssicherung
Content-Prüfsummen (framemd5)

Eigenschaften / Anwendungsfälle:

- Rechenaufwendiger, weil Content erst “entpackt” werden muss
- framemd5 Video: Prüfsummen werden von den unkomprimierten Bildern gemacht
- framemd5 Audio: Prüfsumme über gewisse Anzahl von Samples
- Integritätscheck bei Format- und Codecmigration, sowie Containerupdates
- Bei worst-case Szenarien: verlustfreie Wiederherstellung der Information möglich

Beispiel:

```
$ ffmpeg -i video.avi -an -f framemd5 video_avi.framemd5
```

Digitale Inventur

Konzept / Idee:

- Regelmäßige Überprüfung des Ist-Standes der Daten
- Datenfehler frühzeitig erkennen
- Strukturintegrität von Ordner-basierten Paketen erhalten/dokumentieren
- Schrittweise die Schwere von Fehlern erkennen
- Verwendbar um validierte Backup-Kopien zu erstellen

Digitale Inventur

In der Praxis:

- Eine "Unique ID" (aka Archivsignatur) ist empfehlenswert. zB "vx-00815"
- Prüfsummen für alle Files in einem Ordner erstellen (MD5SUMS Datei)
- Bei Check: Prüfsummenfile für aktuelle Daten erstellen, dann mit original MD5SUMS-Datei "diffen"
- Notifications (Mail?) und Logfiles
- "CV-File": Der Lebenslauf einer Signatur

Prüfsummen + Digitale Inventur

Benutzerfreundlicher(er) Tip...

Library of Congress' "BagIt":

- Entworfen um File- und Strukturintegrität bei Dateitransfer zu checken
- Geht Für jede Art von Files/Ordnern
- Rein Textfile-basiert

`http://en.wikipedia.org/wiki/BagIt`

Tool: "Bagger"

- Commandline + GUI (Java)
- Cross-Platform: Linux, Mac, Win

`http://sourceforge.net/projects/loc-xferutils/files/loc-bagger/`

Fragen?

Thank you very much for your attention!

Some rights reserved...

Contact:

- Free Software Foundation Europe: http://fsfeurope.org/
- Peter Bubestinger: bubestinger@fsfeurope.org