On Audio-Visual File Formats

Reto Kromer • AV Preservation by reto.ch

On the Materiality of Audio-Visual Heritage
Elías Querejeta Zine Eskola
Donostia (San Sebastián), Spain
15–18 October 2024

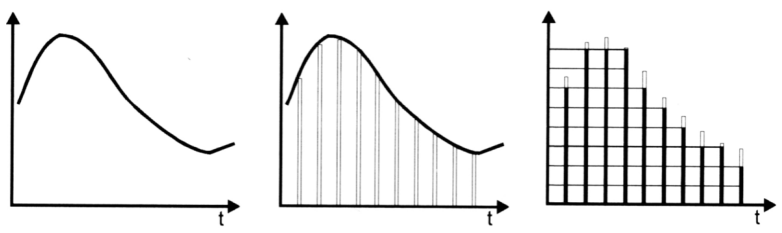
Summary

- digital audio and digital video
- container, codec, raw data
- different formats for different purposes
- audio-visual data transformations
- data maintenance

Digital Audio

Digital Audio

- sampling
- quantisation
- compression



digitisation = sampling + quantisation

Sampling

- 44.1 kHz
- 48 kHz
- 96 kHz
- 192 kHz
- 500 kHz

Quantisation

- 16 bit (2^{16} = 65 536)
- 24 bit (2^{24} = 16 777 216)
- 32 bit (2^{32} = 4 294 967 296)

Digital Video

Digital Video

- resolution
- bit depth
- linear, power, logarithmic
- colour model
- chroma subsampling and compression
- illuminant

Resolution

- SD 480i / SD 576i
- HD 720p / HD 1080i
- 2K / HD 1080p
- 4K / UHD-1
- 8K / UHD-2

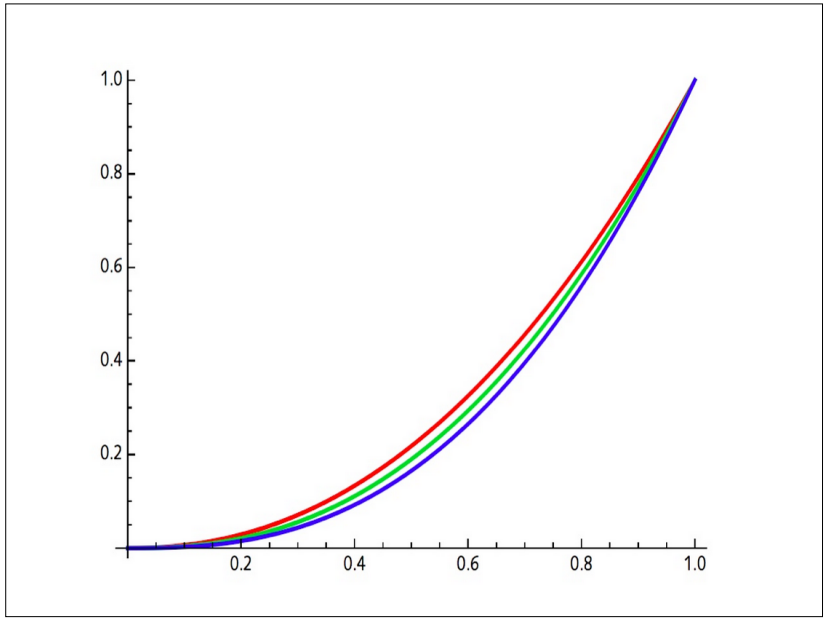
Bit Depth

- 8 bit ($2^8 = 256$)
- 10 bit ($2^{10} = 1\,024$)
- 12 bit ($2^{12} = 4\,096$)
- 16 bit ($2^{16} = 65\,536$)
- 24 bit ($2^{24} = 16\,777\,216$)

Linear, Power, Logarithmic

"medium grey"

- linear scale: 18 %
- power function: 50 %
- logarithmic scale: 50 %

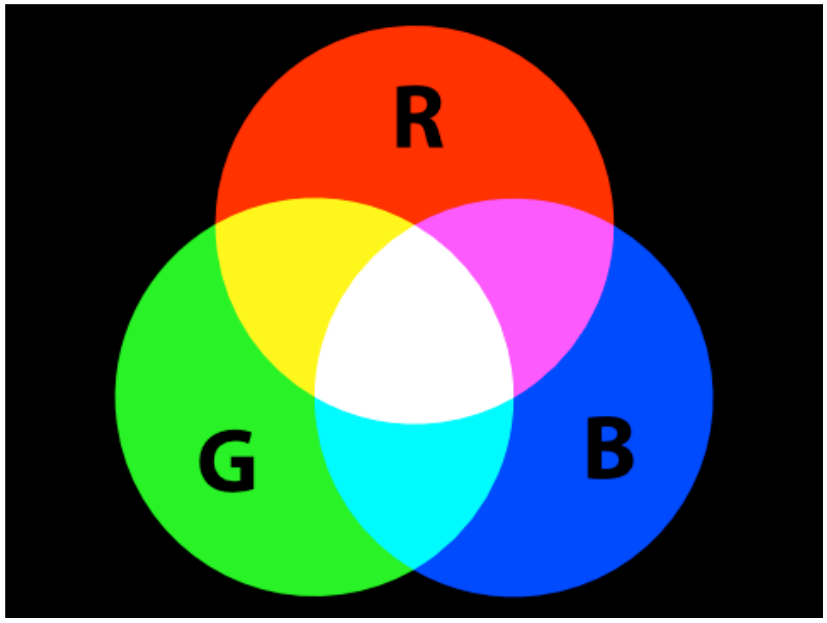


13

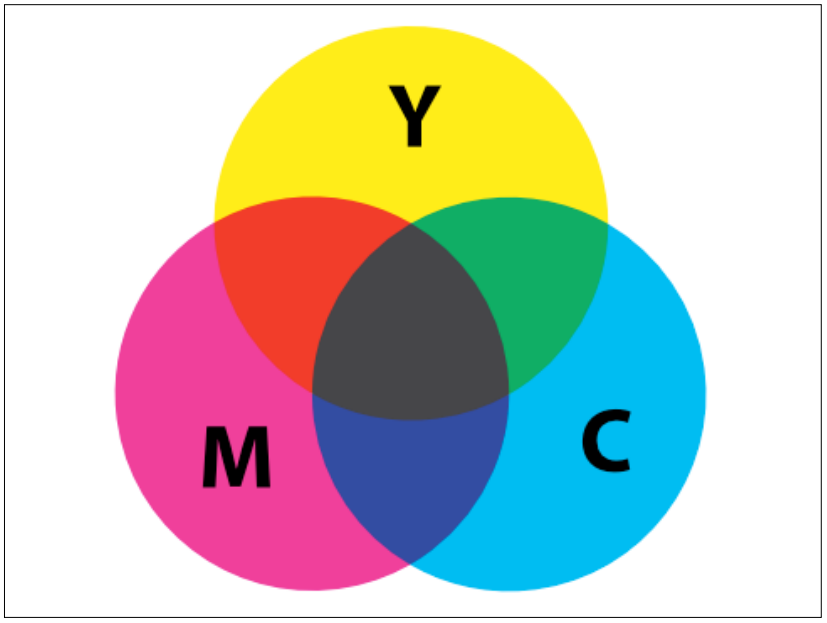
Colour Model

- XYZ, L*a*b*
- RGB / R′G′B′ / CMY / C′M′Y′
- Y′IQ / Y′UV / Y′D_BD_R
- Y′C_BC_R / Y′C_OC_G
- Y′P_BP_R

14



15



16

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.396523 \\ 1 & -0.342793 & -0.711348 \\ 1 & 1.765078 & 0 \end{pmatrix} \begin{pmatrix} Y' \\ C_B \\ C_R \end{pmatrix}$$

$$\begin{pmatrix} Y' \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168074 & -0.329965 & 0.498039 \\ 0.498039 & -0.417947 & -0.080992 \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}$$

17

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} Y' \\ C_O \\ C_G \end{pmatrix}$$

$$\begin{pmatrix} Y' \\ C_O \\ C_G \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}$$

18

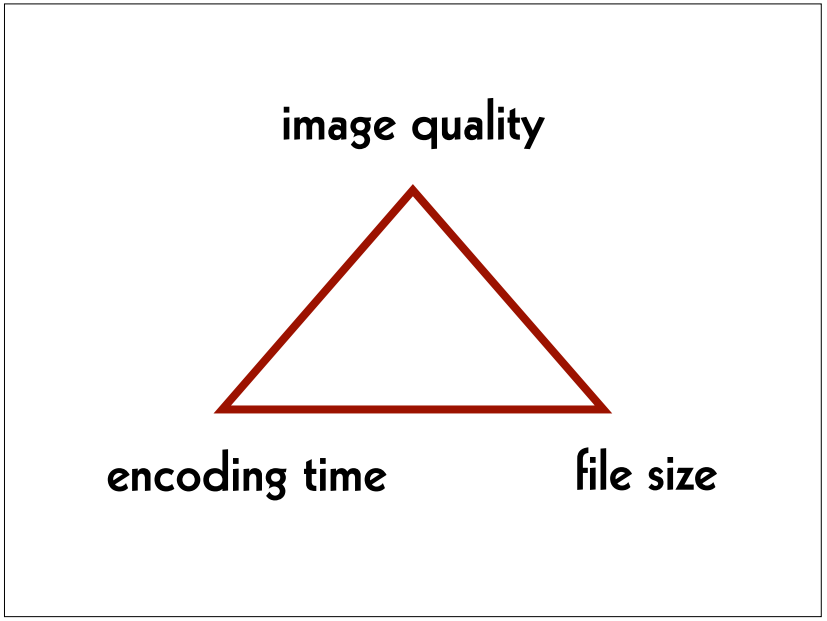
19

RGB24

00000000	11111111	00000000	00000000
00000000	00000000	11111111	00000000
00000000	00000000	00000000	11111111
00000000	11111111	11111111	11111111
11111111	00000000	11111111	11111111
11111111	11111111	00000000	11111111

20

21



22

Compression

- uncompressed
- lossless compression
- lossy compression
- chroma subsampling
- born compressed

23

Uncompressed

+ data simpler to process
+ software runs faster
− bigger files
− slower writing, transmission and reading

Examples: TIFF, DPX, DNG, OpenEXR

24

Lossless Compression

+ smaller files
+ faster writing, transmission and reading
− data processing complexer
− software runs slower

Examples: JPEG 2000, FFV1

Lossy Compression

- optimised for (image acquisition and) postproduction
- optimised for access

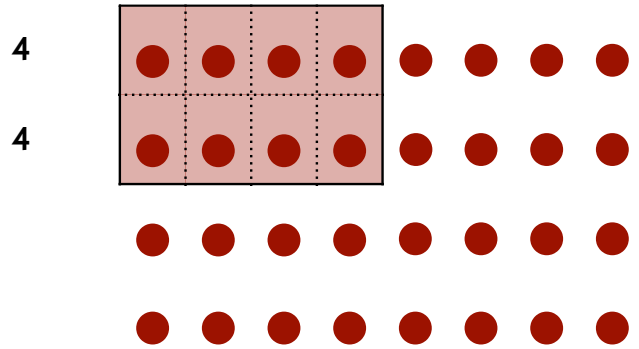
Examples ("mezzanine"): ProRes 422, ProRes 4444; DNxHD, DNxHR

Examples (access): H.264 (AVC), H.265 (HEVC), H.266 (VVC); AV1

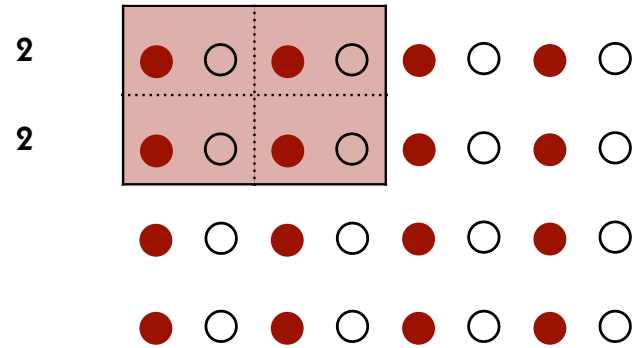
Chroma Subsampling

- 4:4:4
- 4:2:2
- 4:2:0 / 4:1:1

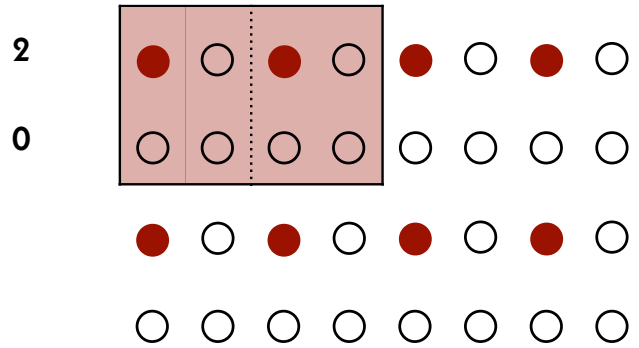
4:4:4



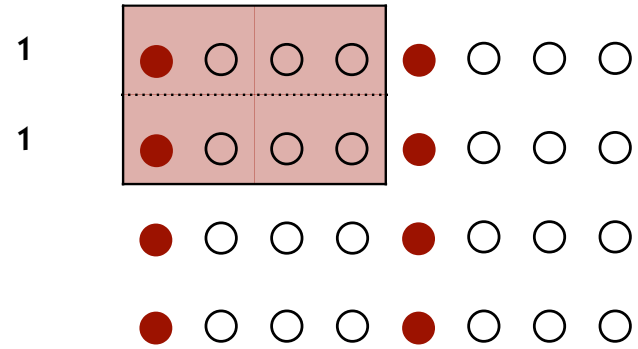
4:2:2



4:2:0



4:1:1



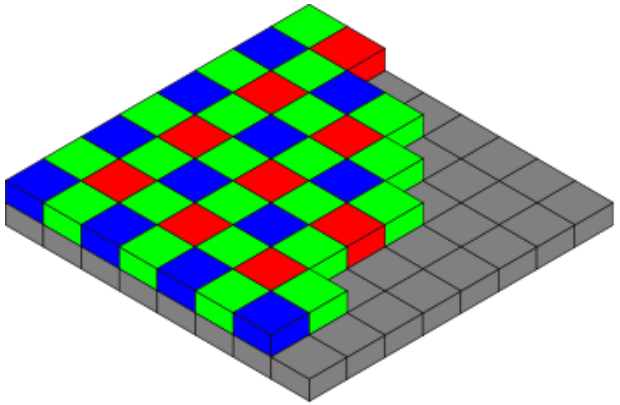
Born Compressed

- optimised for both image acquisition and postproduction

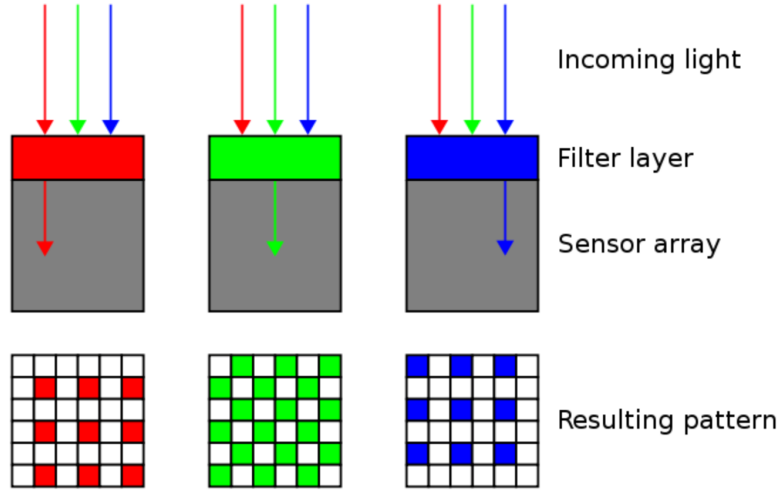
Examples: CineForm RAW, ProRes RAW, Blackmagic RAW

Uncomfortable Truths

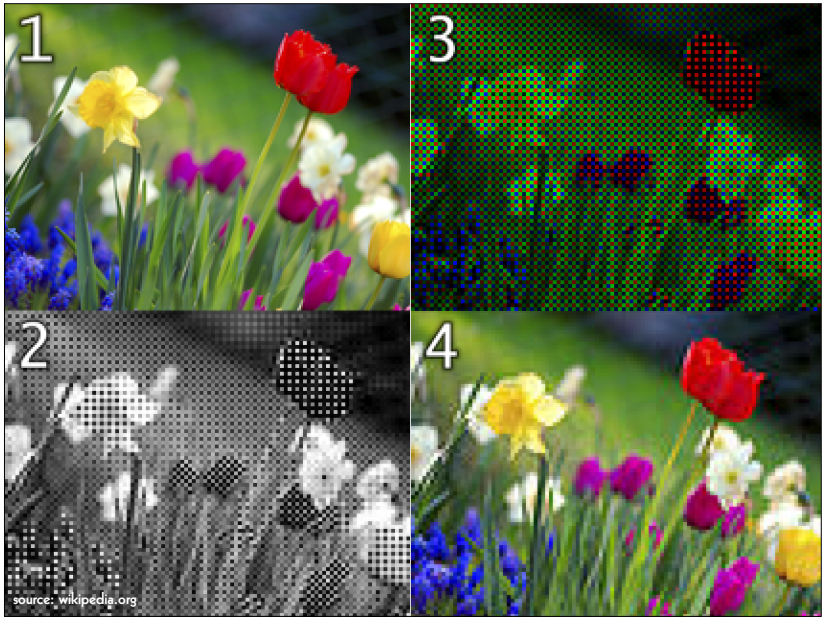
- sensors are colour blind
- Bayer sensors do not generate full RGB



source: wikipedia.org



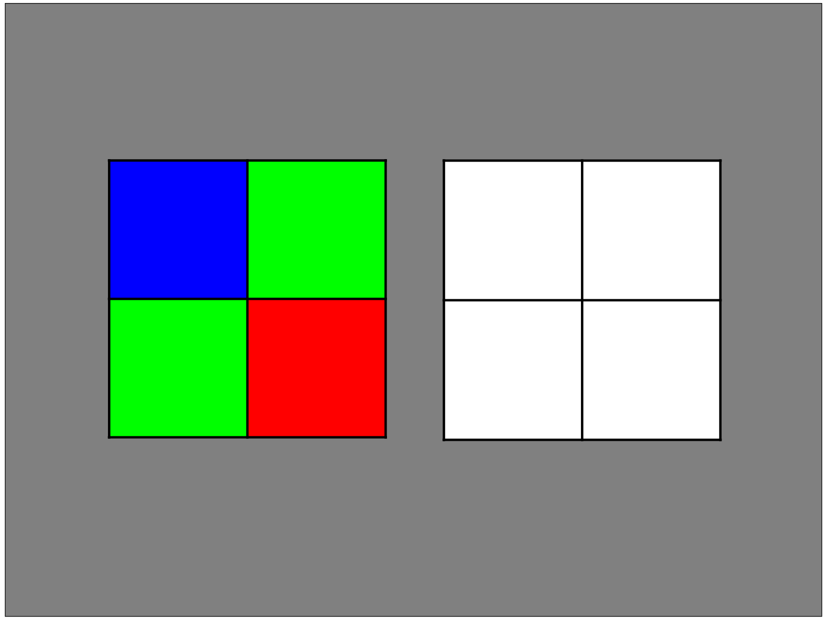
source: wikipedia.org



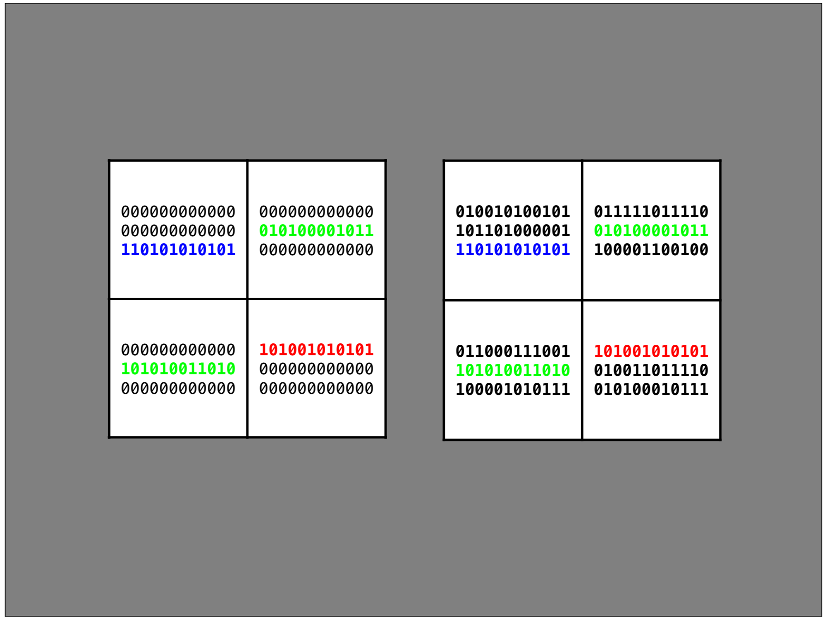
source: wikipedia.org

```
011101010010101010001011010101011110
010011010101010101010000101110101010
011101010010101010001011010101011110
000111010101010101010100001011101010
011010101001010101000101101010101111
001010101010101010000101110101010000
011101010010101010001011010101011110
010101010101010101000010111010100110
100101110101001010101000101101010101
111001010101010101010000101110101010
011101010010101010001011010101011110
010101010101010100110101010000000001
001010001010101010100101010101010101
```

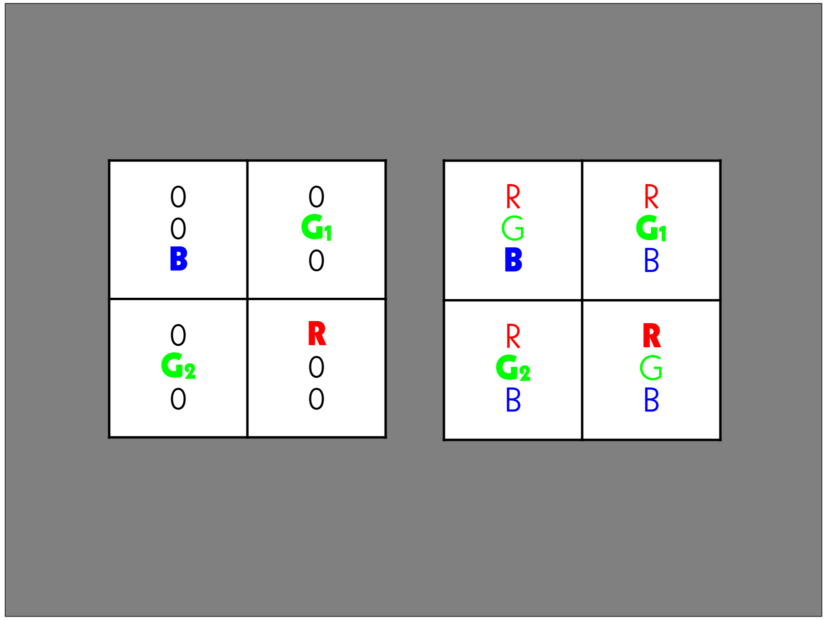
37



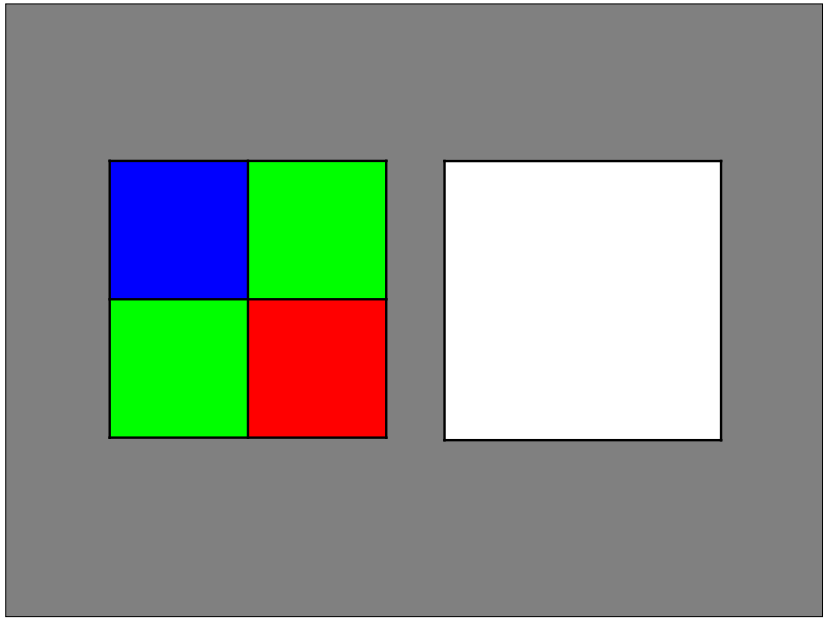
38



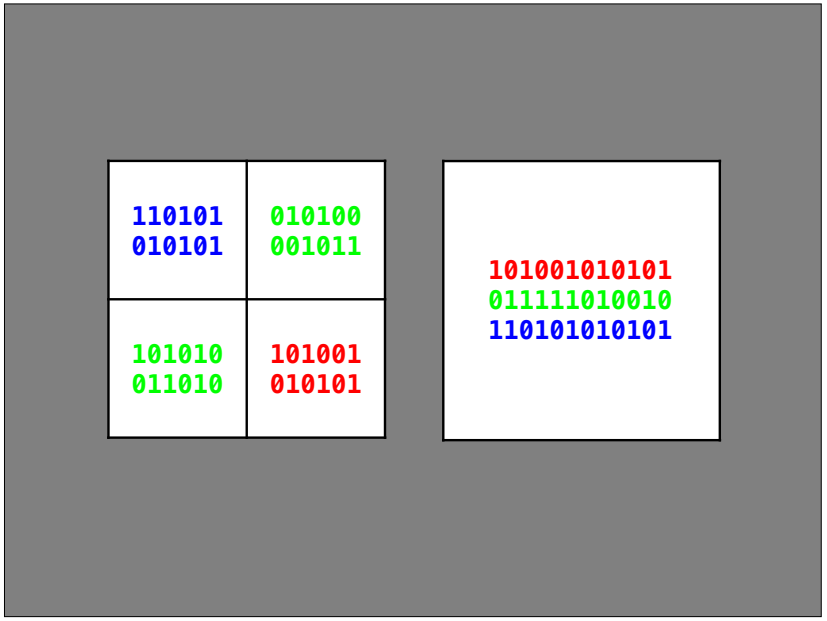
39



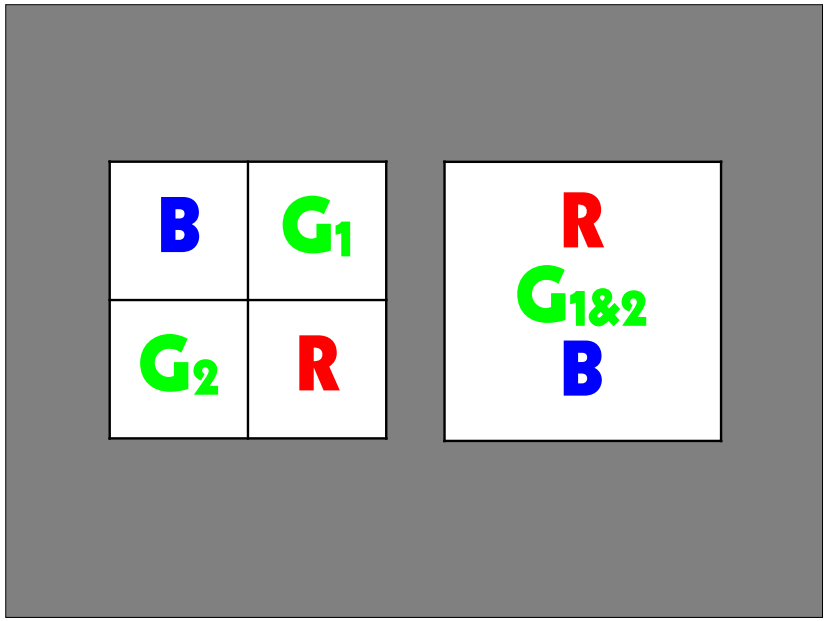
40



41



42



43

Ways to use Bayer-type data

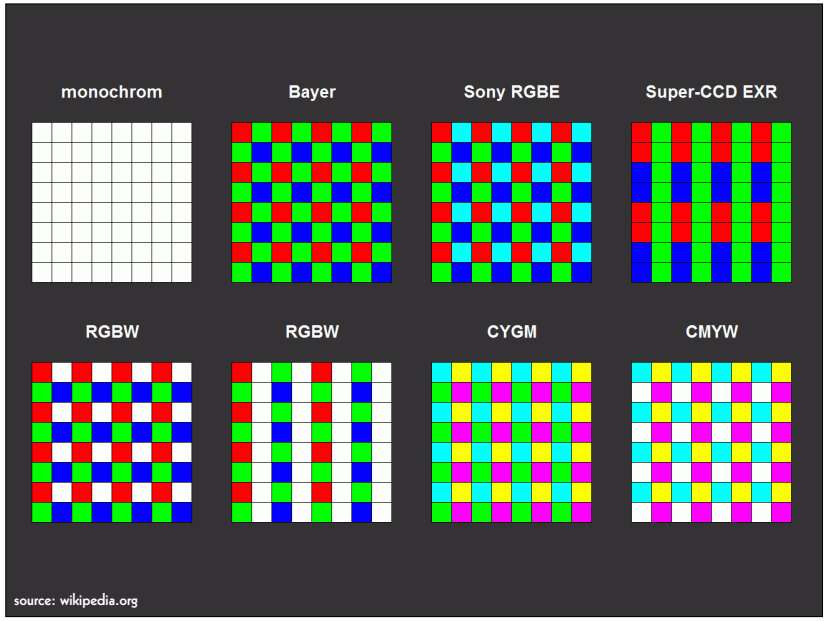
digital blow-up to RGB

- 3 times the amount of the generated data
- the file has the full sensor resolution
- only ⅓ of the data are real

digital reduction to RGB

- ¾ the amount of the generated data
- the file has ½ of the sensor resolution
- all data are real

44



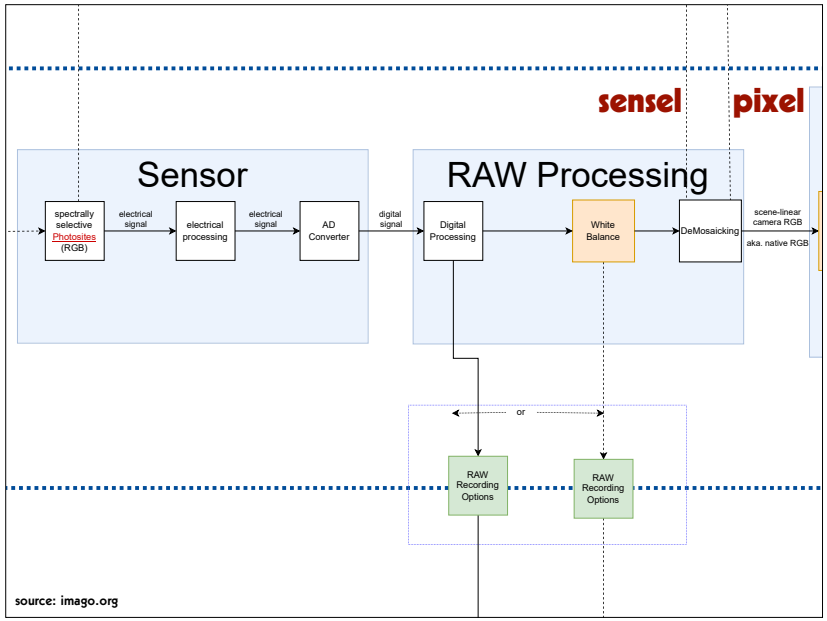
Proposed Terminology

pixel

= picture element

sensel

= sensor element



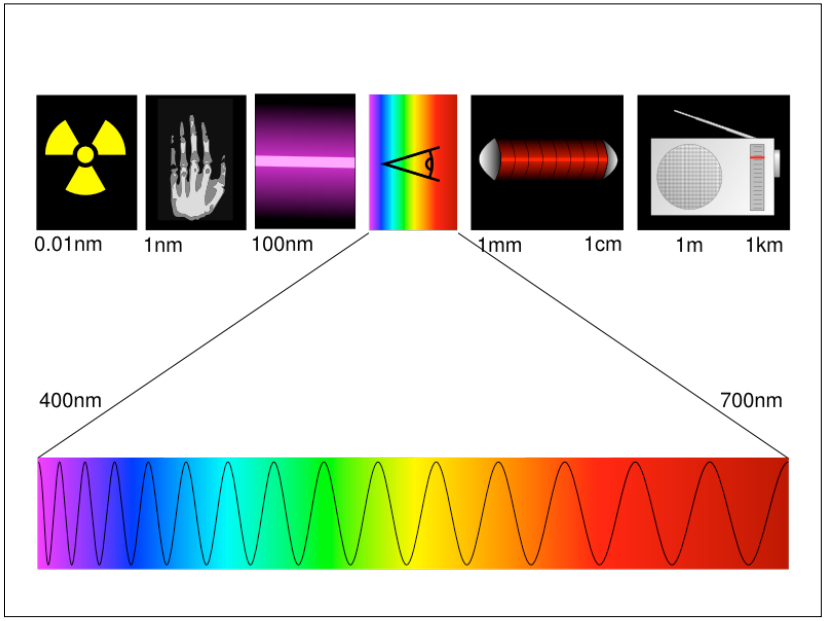
Ways to store Bayer-type data

- pixel values generated by one de-mosaicking algorithm (digital blow-up)
- pixel values generated by mixing two green sensel values into one (digital reduction)
- raw sensel values

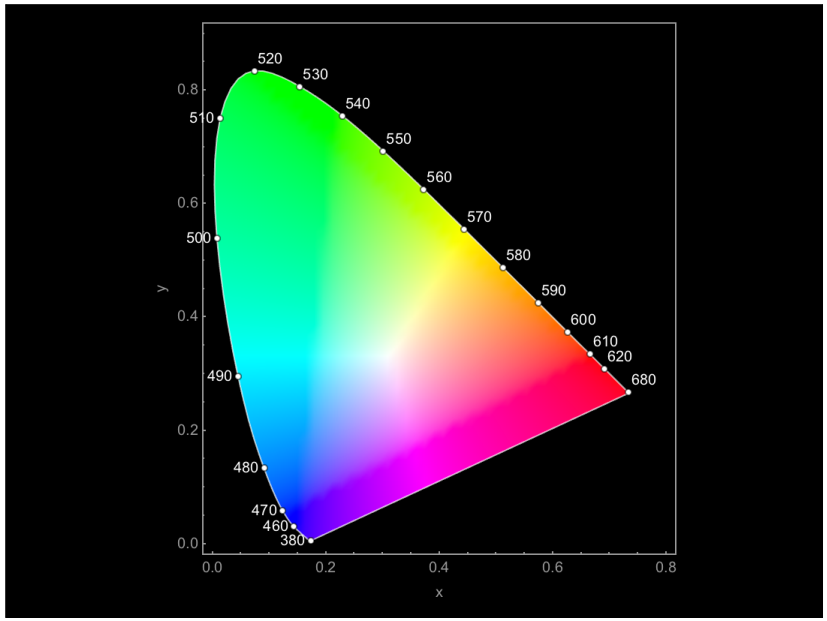
Standard Illuminant

- D50
- D55
- D65
- D75

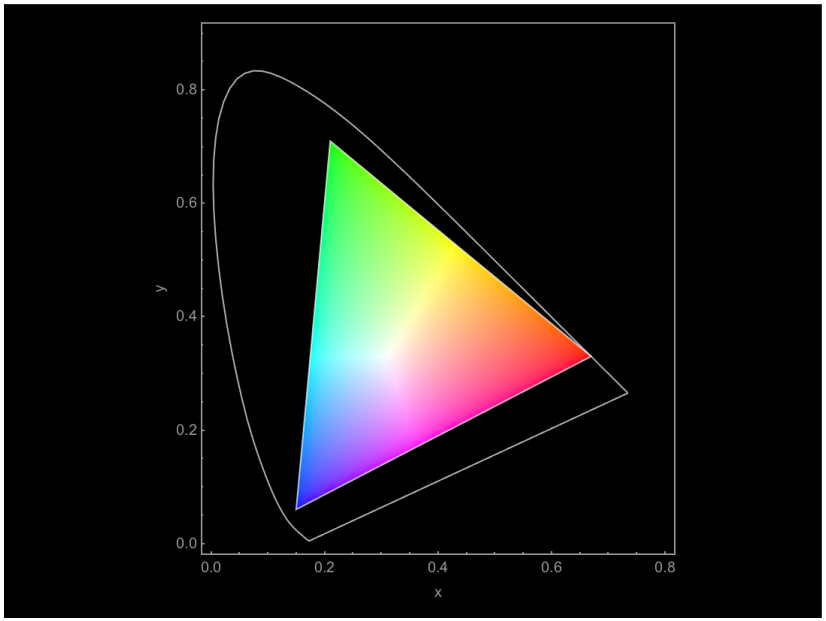
49



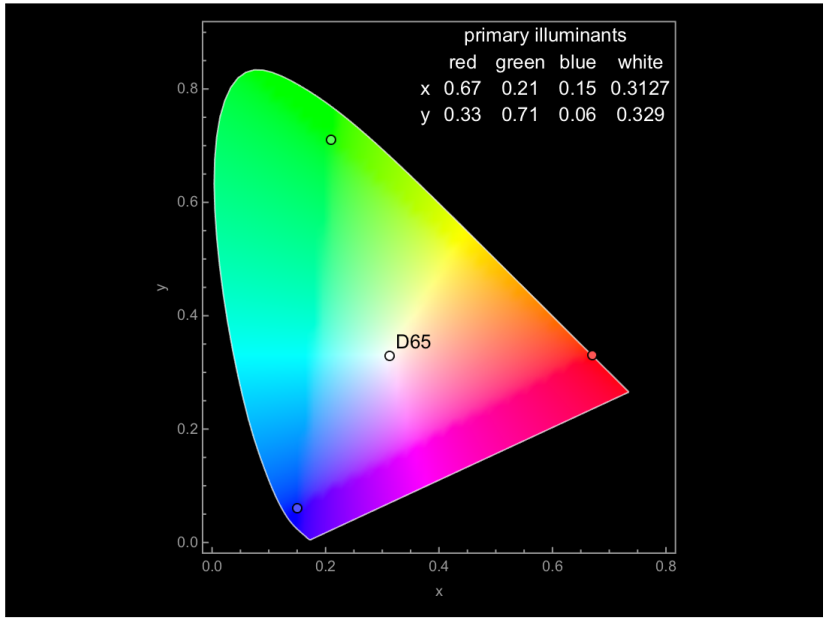
50



51



52



53

File Structure

54

File Structure

audio-visual container (wrapper)

audio codec

video codec

audio data

video data

subtitles codec and data

metadata

55

Audio-Visual Container

- MP4
- QuickTime (.mov)
- AVI
- MXF
- Matroska (.mkv)
- Flash

56

Single Images

- folder
- TAR
- ZIP
- MXF
- Matroska (.mkv)
- CinemaDNG
- Motion JPEG

Audio Codec

- WAVE
- BWF
- AAC
- MP3
- FLAC

Video Codec (Master)

images

- TIFF
- DPX
- JPEG 2000
- OpenEXR
- DNG

streams

- $Y'C_BC_R$ 8 bit
- $Y'C_BC_R$ 10 bit
- HuffYUV
- FFV1

Video Codec (Mezzanine)

- ProRes 422, ProRes 4444, ProRes RAW
- DNxHD, DNxHR
- CineForm RAW
- Blackmagic RAW

Video Codec (Access)

- H.264 (AVC)
- H.265 (HEVC)
- H.266 (VVC)

- AV1

RAW data are cooked.

Audio Data

- pcm_s16le
- pcm_s24le
- pcm_s32le

Video Data

- rgb48le
- rgb24
- rgb72le

- bayer_bggr16le
- bayer_bggr24le

- yuv444p16le
- yuv422p10le
- uyvy422
- yuv420p
- yuv444p24le

What is inside my DPX?

- log neg encoding
- log RGB encoding or quasi-log encoding
- gamma encoding or power function encoding
- scene-linear encoding

File Formats

Principles

- **The archive must be able to handle the file formats it holds.**

- open source
- simple to use and well documented
- widely used by the community

Different Purposes

archive master format:
➔ for preservation

mezzanine format:
➔ for professional use in post-production

dissemination formats:
➔ for widely spreading and easy access

Elena Rossi-Snook:

Archiving without access isn't preservation, it's hoarding.

Archive Master (Today)

film
- folder, TIFF, 2K, RGB, 16 bit
- MXF, DPX, 2K, R'G'B', 10 bit

video
- AVI, "raw", HD, $Y'C_BC_R$, 4:2:2, 10 bit
- Matroska, FFV1, HD, $Y'C_BC_R$, 4:2:2, 10 bit

audio
- BWF, 96 kHz, 24 bit
- FLAC, 96 kHz, 24 bit

Mezzanine (Today)

video
- ProRes 4444, 2K
- DNxHR, 2K
- ProRes 422 HQ, HD
- DNxHD 175x, HD

audio
- BWF, 48 kHz, 24 bit
- WAVE, 48 kHz, 24 bit

Dissemination (Today)

MP4

video
- H.264, SD, yuv420p, lossy
- H.264, "HD", yuv420p, lossy

audio
- AAC, 44.1 kHz, 16 bit
- AAC, 48 kHz, 16 bit

Archive Master and Mezzanine

film
- Matroska, FFV1, 2K, RGB, 16 bit
- Matroska, FFV1, 2K, R′G′B′, 12 bit

video
- Matroska, FFV1, “HD”, $Y'C_BC_R$ 4:2:2, 10 bit
- Matroska, FFV1, “HD”, $Y'C_BC_R$ 4:4:4, 12 bit

audio
- Matroska, FLAC, 96 kHz, 24 bit
- Matroska, FLAC, 192 kHz, 24 bit

Access

video
- H.265, “HD”, yuv420p
- H.266, “HD”, yuv420p
- AV1, “HD”, yuv420p

audio
- FLAC, 48 kHz, 16 bit
- FLAC, 96 kHz, 16 bit

Reading

Reto Kromer: **Matroska and FFV1: One File Format for Film and Video Archiving?**, in «Journal of Film Preservation», n. 96 (April 2017), FIAF, Brussels, Belgium, p. 41–45

➔ retokromer.ch/publications/JFP_96.html

Pros & Cons

container:

- folder
- TAR
- ZIP
- MXF
- Matroska
- AXF

video codec:

- TIFF
- DPX
- JPEG 2000
- FFV1
- OpenEXR
- CineForm RAW
- ProRes RAW
- Blackmagic RAW

	avantages	disavantages
TIFF DPX OpenEXR	data easier to process	bigger files
JPEG 2000 FFV1	smaller files	data complexer to process

The Bridge

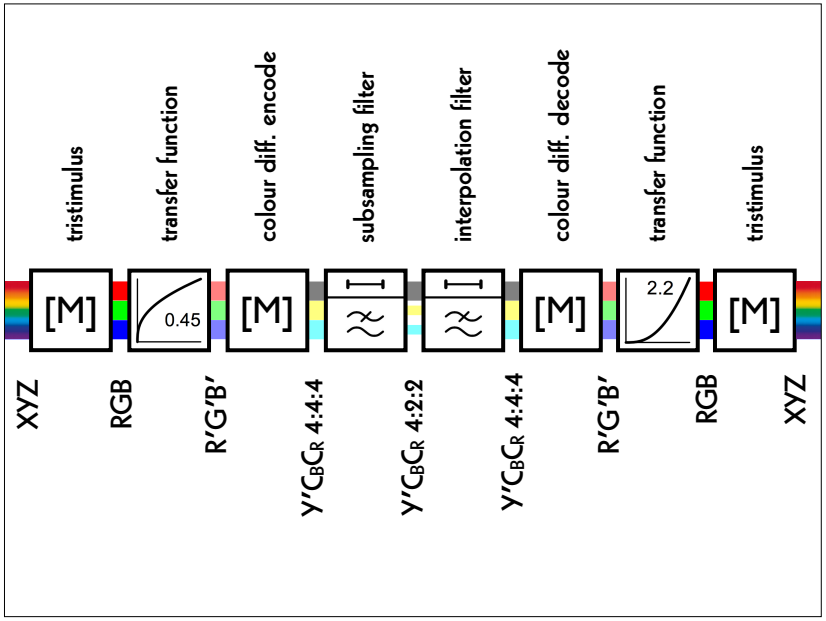
RAWcooked (CLI)

➔ mediaarea.net/RAWcooked

RAWcooked

- encoding into Matroska (.mkv) using FFV1 video codec and FLAC audio codec
- all metadata preserved
- decoding with bit-by-bit reversibility
- possibility to embed sidecar files (e.g. MD5, LUT, XML)
- compatibility with media players

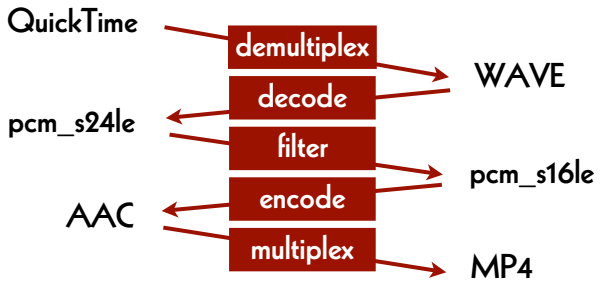
Transformations



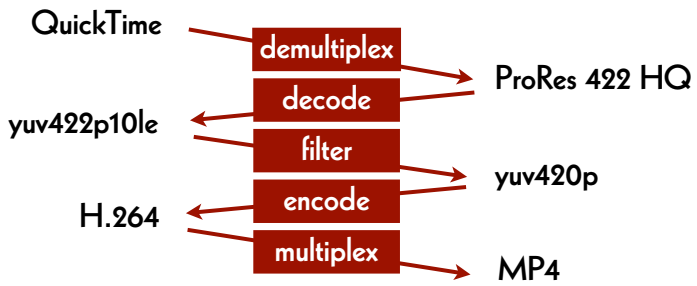
Data Transformations

demultiplex
decode
filter
encode
multiplex

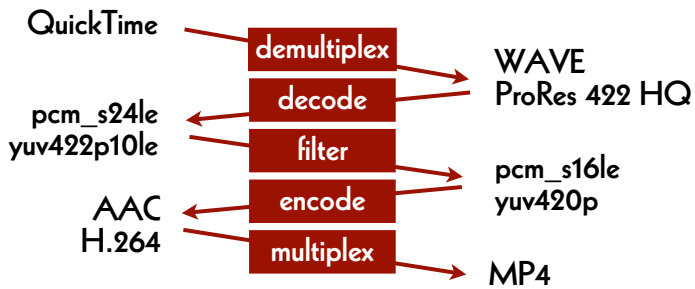
Audio Exemple



Video Exemple



Audio-Visual Exemple



Data Maintenance

Plan the Next Migration

- file naming
- barcodes
- checksums
- write the full index onto the cartridge
- technical metadata
- code to retrieve the files

File Naming (Example)

- title_codec.container
- title_codec_container_algorithm.txt

- film_H264.mp4
- film_H264_mp4_md5.txt

Checksums

cryptographic

- MD5
- SHA-1
- SHA-256
- SHA-512

non-cryptographic

- CRC-32
- xxHash 32
- xxHash 64
- xxHash 128

Longterm

- storage of the cartridges
- three copies …
- … in geographically distant locations
- data integrity check
- data migration
- availability of LTO desks

Data Migrations

2014

- our internal archive from LTO-4 to LTO-6 (5.7 PB)

2014–2021

- two dozen migrations for clients

2021

- our internal archive from LTO-6 to LTO-8 (25.2 PB)

Reading

Reto Kromer: **On the Bright Side of Data Migrations**, in «IASA Journal», n. 49 (December 2018), IASA, p. 18–22

➔ retokromer.ch/publications/IASA_49.html

read | script | write

script to modify

- container
- codec
- both container and codec
- metadata
- filename

#1: ProRes-born Content

from:

- ProRes stored in a QuickTime (.mov) container

to:

- ProRes stored in a Matroska (.mkv) container

Update the Container

➔ read file from source LTO

➔ demultiplex file

- ProRes 422, 10 bit [yuv422p10le]
- ProRes 4444, 10 bit [yuv444p10le or yuva444p10le] or 12 bit [yuv444p12le]

➔ multiplex file

➔ write file to destination LTO

SMPTE RDD 36:2015

SMPTE REGISTERED DISCLOSURE DOCUMENT

Apple ProRes Bitstream Syntax and Decoding Process

Page 1 of 39 pages

The attached document is a Registered Disclosure Document prepared by the sponsor identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Every attempt has been made to ensure that the information contained in this document is accurate. Errors in this document should be reported to the proponent identified below, with a copy to eng@smpte.org.

#2: Video

from:

- AVI / 8-bit and 10-bit uncompressed
- MOV / 8-bit and 10-bit uncompressed
- MP4 / 8-bit and 10-bit uncompressed

to:

- Matroska / FFV1

Container and Codec

- → read file from source LTO
- → demultiplex file
- → decode file
  - Y′C_BC_R, 4:2:2, 8 bit, uyvy422
- → encode file
- → multiplex file
- → write file to destination LTO

Container and Codec

- → read file from source LTO
- → demultiplex file
- → decode file
  - Y′C_BC_R, 4:2:2, 10 bit, yuv422p10le
- → encode file
- → multiplex file
- → write file to destination LTO

#3: Filename

from:

- Title_YUV422.mkv

to:

- Title_YCbCr422_9d5084b5b0a08d5022b39e0e75241d12.mkv

Always remember:

To do nothing is **never** an option!

Live in the real world!

There is only one efficient way:

- keep the analogue source elements as long as possible
- more prevention:
  → better insulation
  → more efficient air conditioning
- less handling of the source elements
- make digital masters and access copies

Acknowledgements (1)

- Swiss Federal Institute of Technology
- Massachusetts Institute of Technology
- Kinemathek Lichtspiel, Bern

- Charles Poynton
- Dave Rice & Misty De Meo
- Agathe Jarczyk & David Pfluger

Acknowledgements (2)

- Tommy Aschenbach
- Claudio Weidmann
- Jim Lindner
- Carl Eugen Hoyos
- Peter Bubestinger-Steindl
- Jérôme Martinez
- Michael Niedermayer

AV Preservation by

reto.ch

Sandrainstrasse 3
3007 Bern
Switzerland

Web: reto.ch
Email: info@reto.ch