Greener Digital Archiving

Reto Kromer • AV Preservation by reto.ch

FIAF Commissions' Thursday
Online Workshops
14 November 2024

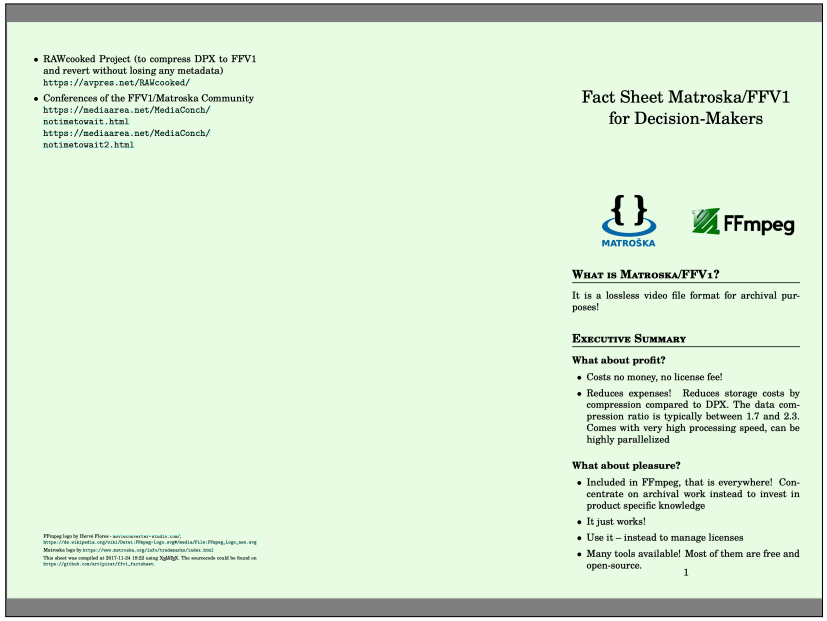
Summary

- no time to wait
- compression
- checksums
- outlook

No Time to Wait

No Time to Wait

- Berlin 2016
- Wien 2017
- London 2018
- Budapest 2019
- ~~Hilversum~~ 2021 virtual conference
- Den Haag 2022
- Praha 2023
- Karlsruhe 2024



- RAWcooked Project (to compress DPX to FFV1 and revert without losing any metadata) https://avpres.net/RAWcooked/
- Conferences of the FFV1/Matroska Community https://mediaarea.net/MediaConch/notimetowait.html https://mediaarea.net/MediaConch/notimetowait2.html

FFmpeg logo by Hervé Pierre - [illegible] https://de.wikipedia.org/wiki/Datei:FFmpeg-Logo.svg#/media/File:FFmpeg_Logo_new.svg
Matroska logo by https://www.matroska.org/[illegible]/index.html
This sheet was compiled at 2017-11-24 19:22 using [illegible]. The sourcecode could be found on https://github.com/[illegible]/ffv1_factsheet.

Fact Sheet Matroska/FFV1 for Decision-Makers

MATROŠKA

FFmpeg

What is Matroska/FFV1?

It is a lossless video file format for archival purposes!

Executive Summary

What about profit?

- Costs no money, no license fee!
- Reduces expenses! Reduces storage costs by compression compared to DPX. The data compression ratio is typically between 1.7 and 2.3. Comes with very high processing speed, can be highly parallelized

What about pleasure?

- Included in FFmpeg, that is everywhere! Concentrate on archival work instead to invest in product specific knowledge
- It just works!
- Use it – instead to manage licenses
- Many tools available! Most of them are free and open-source.

1

5

What about power?

- Hey, it is open-source! There is no vendor lock-in!
- Free and well documented standardization process (IETF)! No paywalls (as with ISO and SMPTE).
- Very friendly community!

What about prestige?

- Archives should preserve knowledge, should they not share knowledge?
- You may be the first to develop new tools!
- You can be the first to evolve it for special workflows!
- You can use a **real** archival file format!
- You can participate in the community and increase your visibility!

Use cases

- Normalization to reduce format diversity in the archive
- Archival file format for retro-digitization of film (linear, 16 bit) or video
- File format to exchange and work with it
- Comes with integrated checksum information
- Reduce storage and transfer cost
- Replacement of lossy mezzanine video files for editing

FAQ - frequently asked questions

What are the general requirements for archival file formats?

Each file format should be

- ✓ Open specified
- ✓ Very popular
- ✓ Robust
- ✓ Self-documented
- ✓ Validatable
- ✓ Tidy and concise

Matroska/FFV1 fulfills these requirements.

What means lossless?

It means there is no information loss. The original source can be fully reconstructed. This also means that migrations can be tested automatically!

Who use it?

There are a lot of institutions. For example:

- Filmarchiv Austria
- Österreichische Mediathek
- City of Vancouver Archives
- SMIDAK Filmproduktion
- Irish Film Institute
- RTV Slovakia
- AV Preservation by reto.ch
- The National Archives (United Kingdom)
- Indiana University
- New York Public Library
- NOA-Archive http://noa-archive.com

Matroska/FFV1 is also recommended and examined by

- Federal Agencies Digital Guidelines Initiatives (FADGI) http://www.digitizationguidelines.gov/guidelines/video_reformatting_compare.html
- nestor AG AV-Medien, „nestor materialien 19 – Leitfaden für die digitale Langzeitarchivierung audiovisueller Medien" http://nbn-resolving.de/urn:nbn:de:0008-2016102107
- PACKED vzw: Centre of Expertise in Digital Heritage, „A short guide to choosing a digital format for video archiving masters" https://www.scart.be/?q=en/content/short-guide-choosing-digital-format-video-archiving-masters
- „Memoriav Recommendations: Digital Archiving of Film and Video: Principles and Guidance" http://memoriav.ch/wp-content/uploads/2017/09/Empfehlungen_Digitale-Archivierung_Version1.1_EN.pdf
- IANUS, „IT-Empfehlungen für den nachhaltigen Umgang mit digitalen Daten in den Altertumswissenschaften" https://www.ianus-fdz.de/it-empfehlungen/video

Additional resources

- FFmpeg https://ffmpeg.org
- Wikipedia page on FFV1 https://en.wikipedia.org/wiki/FFV1
- „Codec Encoding for LossLess Archiving and Realtime transmission" (Cellar), standardization working group of the IETF https://tools.ietf.org/wg/cellar/
- Preforma-Project http://www.preforma-project.eu/
- FAQ „FFV1 at the Österreichische Mediathek" https://www.mediathek.at/digitalisierung/ffv1-at-the-mediathek-engl/

6

Stream:	Internet Engineering Task Force (IETF)
RFC:	8794
Category:	Standards Track
Published:	July 2020
ISSN:	2070-1721
Authors:	S. Lhomme D. Rice M. Bunkus

RFC 8794
Extensible Binary Meta Language

Abstract

This document defines the Extensible Binary Meta Language (EBML) format as a binary container format designed for audio/video storage. EBML is designed as a binary equivalent to XML and uses a storage-efficient approach to build nested Elements with identifiers, lengths, and values. Similar to how an XML Schema defines the structure and semantics of an XML Document, this document defines how EBML Schemas are created to convey the semantics of an EBML Document.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the

7

Stream:	Internet Engineering Task Force (IETF)
RFC:	9043
Category:	Informational
Published:	August 2021
ISSN:	2070-1721
Authors:	M. Niedermayer D. Rice J. Martinez

RFC 9043
FFV1 Video Coding Format Versions 0, 1, and 3

Abstract

This document defines FFV1, a lossless, intra-frame video encoding format. FFV1 is designed to efficiently compress video data in a variety of pixel formats. Compared to uncompressed video, FFV1 offers storage compression, frame fixity, and self-description, which makes FFV1 useful as a preservation or intermediate video format.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for

8

Stream:	Internet Engineering Task Force (IETF)
RFC:	9559
Updates:	8794
Category:	Standards Track
Published:	October 2024
ISSN:	2070-1721
Authors:	S. Lhomme M. Bunkus D. Rice

RFC 9559
Matroska Media Container Format Specification

Abstract

This document defines the Matroska audiovisual data container structure, including definitions of its structural elements, terminology, vocabulary, and application.

This document updates RFC 8794 to permit the use of a previously reserved Extensible Binary Meta Language (EBML) Element ID.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the

Matroska and FFV1

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

Data Migration

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

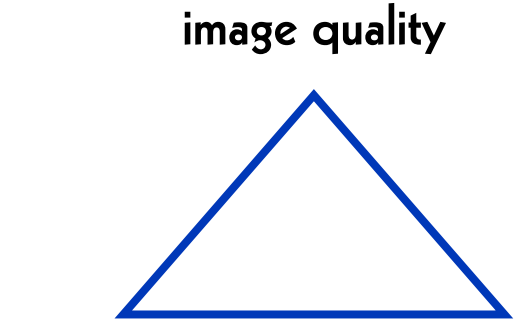
➔ retokromer.ch/publications/IASA_49

Bayer-Type Data

Reto Kromer: **From the Sensor to the File: Different Ways to Work with Bayer-Type Data**, […] 2025

➔ link coming soonish

Compression



Compression

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

Uncompressed

+ data processing is simpler
+ software runs faster

– bigger files
– slower writing, transmission and reading

Examples: TIFF, DPX, DNG, OpenEXR

Lossless Compression

+ smaller files
+ faster writing, transmission and reading
- data processing is more complex
- software runs slower

Examples: JPEG 2000, FFV1

JPEG 2000 or FFV1

- similar compression rate
- compression time of JPEG 2000 higher than that of FFV1
- GPU compression for JPEG 2000
- not yet a public GPU compression for FFV1

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

Single Images and Streams

RAWcooked (CLI)

➔ mediaarea.net/RAWcooked

```
# to encode:
rawcooked input_path_of_folder
# to decode:
rawcooked rawcooked_input_file.mkv
# to get help:
rawcooked -h
man rawcooked
```

RAWcooked

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

Nicholas Camardo
Digital Imaging & Content Manager at University of Southern California
17 Std.

#RAWCooked stats:

- 🎞️ Input DPX: 4K; 16-bit; RGB; 37,476 frames; 2.49TiB
- 💾 Output MKV: Losslessly compressed to 1.37TiB
- ⬇️ Compression ratio: 55%
- 🖥️ Hardware: Intel Xeon E5-2680 V3 @ 2.50GHz x 48
- ⚙️ Software: Ubuntu 20.04; ffmpeg 7.0, RAWCooked 23.12
- 🔀 Transcode time: Approx. 21.5 hours
- ⏳ Total processing time (including MD5 manifest, reversibility checks & checksum validation): Approx. 87 hours

source: Nicholas Camardo, LinkedIn, 2024-11-12

British Film Institute

- large scale re-encoding and re-wrapping
- Stephen McConnachie, Joanna White

Lossy Compression

- **optimised for** image acquisition and **postproduction**

Examples: ProRes 422, ProRes 4444; DNxHD, DNxHR

- **optimised for access**

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

SMPTE RDD 36:2015

SMPTE REGISTERED DISCLOSURE DOCUMENT

Apple ProRes Bitstream Syntax and Decoding Process

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.

25

Primary -> Access

```
ffmpeg
  -f image2
  -framerate 24
  -i input_file_regex.extention
  -pix_fmt yuv420p
  -c:v libx264
  -preset veryslow
  -crf 23
  -movflags +faststart
  output_file.mp4
```

26

-bash

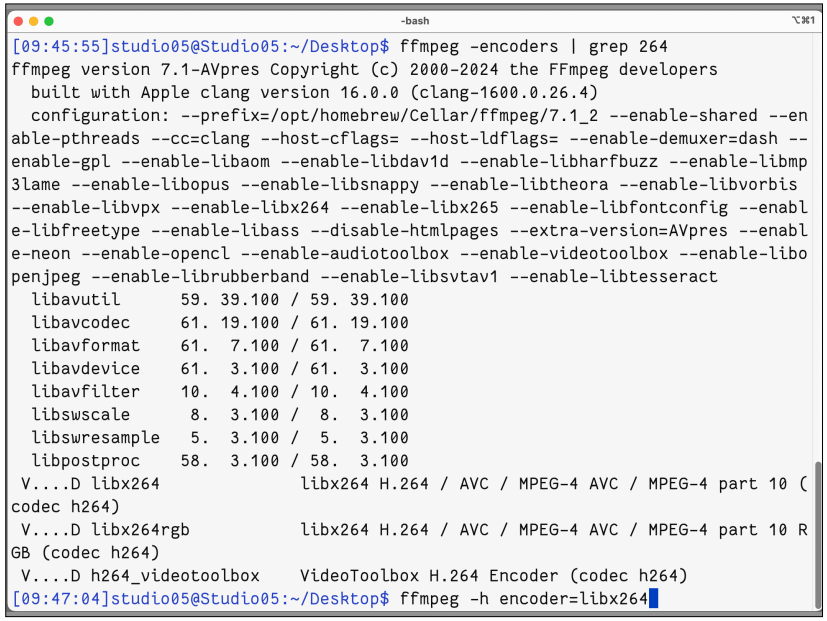
```
[09:43:43]studio05@Studio05:~/Desktop$ ffmpeg -encoders | grep 264
```

27

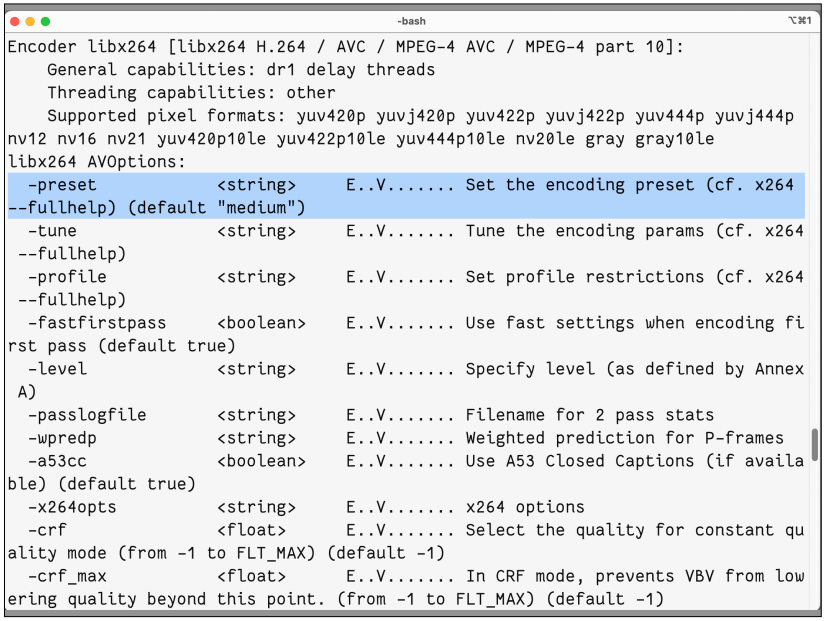
-bash

```
[09:43:43]studio05@Studio05:~/Desktop$ ffmpeg -encoders | grep 264
ffmpeg version 7.1-AVpres Copyright (c) 2000-2024 the FFmpeg developers
  built with Apple clang version 16.0.0 (clang-1600.0.26.4)
  configuration: --prefix=/opt/homebrew/Cellar/ffmpeg/7.1_2 --enable-shared --en
able-pthreads --cc=clang --host-cflags= --host-ldflags= --enable-demuxer=dash --
enable-gpl --enable-libaom --enable-libdav1d --enable-libharfbuzz --enable-libmp
3lame --enable-libopus --enable-libsnappy --enable-libtheora --enable-libvorbis
--enable-libvpx --enable-libx264 --enable-libx265 --enable-libfontconfig --enabl
e-libfreetype --enable-libass --disable-htmlpages --extra-version=AVpres --enabl
e-neon --enable-opencl --enable-audiotoolbox --enable-videotoolbox --enable-libo
penjpeg --enable-librubberband --enable-libsvtav1 --enable-libtesseract
  libavutil      59. 39.100 / 59. 39.100
  libavcodec     61. 19.100 / 61. 19.100
  libavformat    61.  7.100 / 61.  7.100
  libavdevice    61.  3.100 / 61.  3.100
  libavfilter    10.  4.100 / 10.  4.100
  libswscale      8.  3.100 /  8.  3.100
  libswresample   5.  3.100 /  5.  3.100
  libpostproc    58.  3.100 / 58.  3.100
 V....D libx264              libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (
codec h264)
 V....D libx264rgb           libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 R
GB (codec h264)
 V....D h264_videotoolbox    VideoToolbox H.264 Encoder (codec h264)
[09:44:14]studio05@Studio05:~/Desktop$
```

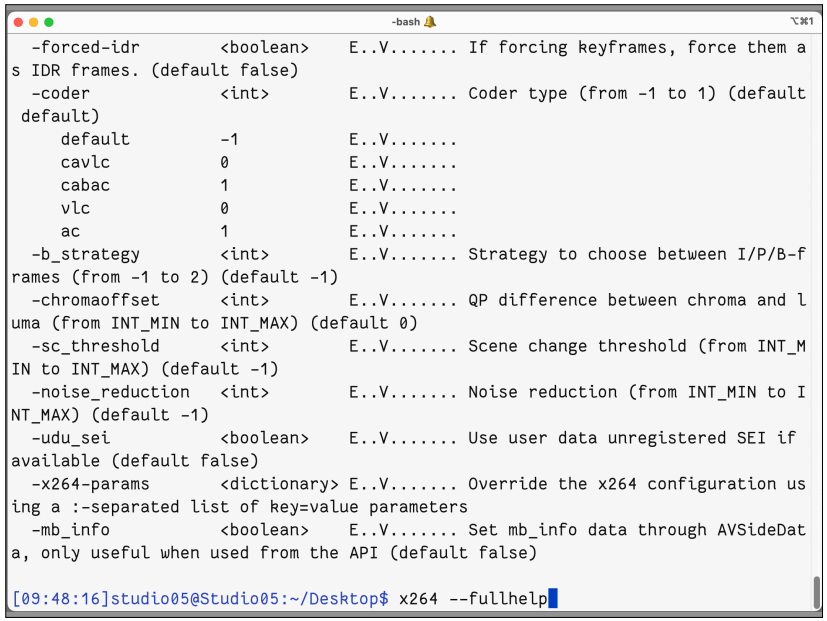
28


-bash
[09:45:55]studio05@Studio05:~/Desktop$ ffmpeg -encoders | grep 264
ffmpeg version 7.1-AVpres Copyright (c) 2000-2024 the FFmpeg developers
built with Apple clang version 16.0.0 (clang-1600.0.26.4)
configuration: --prefix=/opt/homebrew/Cellar/ffmpeg/7.1_2 --enable-shared --enable-pthreads --cc=clang --host-cflags= --host-ldflags= --enable-demuxer=dash --enable-gpl --enable-libaom --enable-libdav1d --enable-libharfbuzz --enable-libmp3lame --enable-libopus --enable-libsnappy --enable-libtheora --enable-libvorbis --enable-libvpx --enable-libx264 --enable-libx265 --enable-libfontconfig --enable-libfreetype --enable-libass --disable-htmlpages --extra-version=AVpres --enable-neon --enable-opencl --enable-audiotoolbox --enable-videotoolbox --enable-libopenjpeg --enable-librubberband --enable-libsvtav1 --enable-libtesseract
libavutil 59. 39.100 / 59. 39.100
libavcodec 61. 19.100 / 61. 19.100
libavformat 61. 7.100 / 61. 7.100
libavdevice 61. 3.100 / 61. 3.100
libavfilter 10. 4.100 / 10. 4.100
libswscale 8. 3.100 / 8. 3.100
libswresample 5. 3.100 / 5. 3.100
libpostproc 58. 3.100 / 58. 3.100
V....D libx264 libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (codec h264)
V....D libx264rgb libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 RGB (codec h264)
V....D h264_videotoolbox VideoToolbox H.264 Encoder (codec h264)
[09:47:04]studio05@Studio05:~/Desktop$ ffmpeg -h encoder=libx264

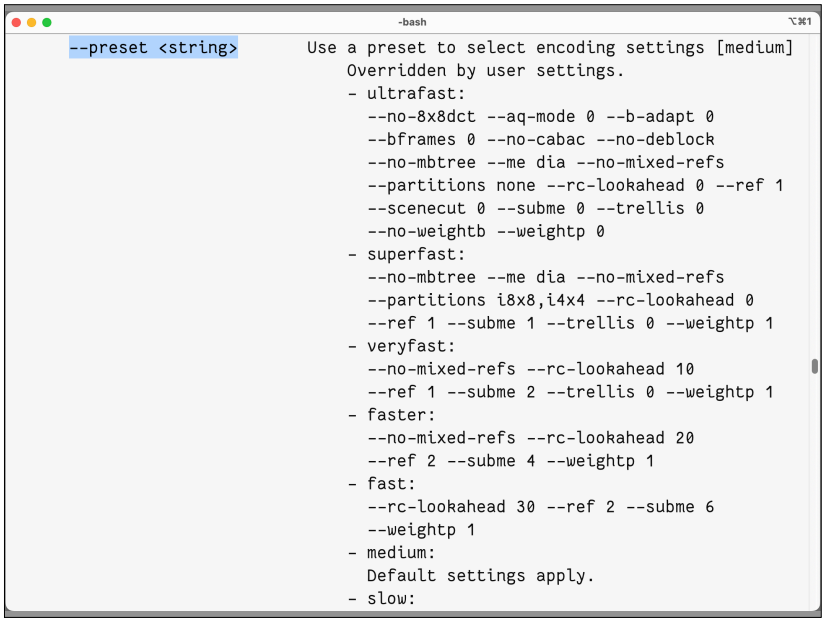
29


-bash
Encoder libx264 [libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10]:
General capabilities: dr1 delay threads
Threading capabilities: other
Supported pixel formats: yuv420p yuvj420p yuv422p yuvj422p yuv444p yuvj444p nv12 nv16 nv21 yuv420p10le yuv422p10le yuv444p10le nv20le gray gray10le
libx264 AVOptions:
-preset <string> E..V..... Set the encoding preset (cf. x264 --fullhelp) (default "medium")
-tune <string> E..V..... Tune the encoding params (cf. x264 --fullhelp)
-profile <string> E..V..... Set profile restrictions (cf. x264 --fullhelp)
-fastfirstpass <boolean> E..V..... Use fast settings when encoding first pass (default true)
-level <string> E..V..... Specify level (as defined by Annex A)
-passlogfile <string> E..V..... Filename for 2 pass stats
-wpredp <string> E..V..... Weighted prediction for P-frames
-a53cc <boolean> E..V..... Use A53 Closed Captions (if available) (default true)
-x264opts <string> E..V..... x264 options
-crf <float> E..V..... Select the quality for constant quality mode (from -1 to FLT_MAX) (default -1)
-crf_max <float> E..V..... In CRF mode, prevents VBV from lowering quality beyond this point. (from -1 to FLT_MAX) (default -1)

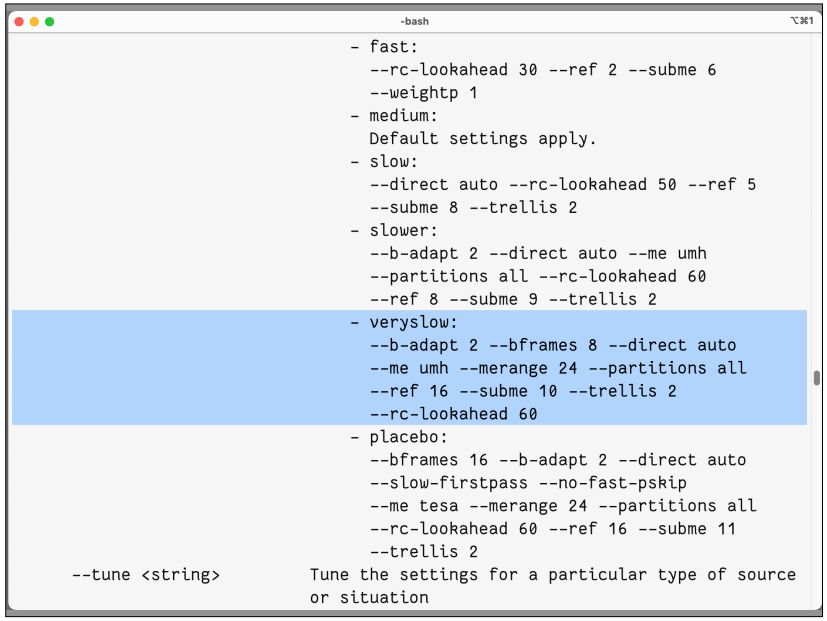
30


-bash
-forced-idr <boolean> E..V..... If forcing keyframes, force them as IDR frames. (default false)
-coder <int> E..V..... Coder type (from -1 to 1) (default default)
default -1 E..V.....
cavlc 0 E..V.....
cabac 1 E..V.....
vlc 0 E..V.....
ac 1 E..V.....
-b_strategy <int> E..V..... Strategy to choose between I/P/B-frames (from -1 to 2) (default -1)
-chromaoffset <int> E..V..... QP difference between chroma and luma (from INT_MIN to INT_MAX) (default 0)
-sc_threshold <int> E..V..... Scene change threshold (from INT_MIN to INT_MAX) (default -1)
-noise_reduction <int> E..V..... Noise reduction (from INT_MIN to INT_MAX) (default -1)
-udu_sei <boolean> E..V..... Use user data unregistered SEI if available (default false)
-x264-params <dictionary> E..V..... Override the x264 configuration using a :-separated list of key=value parameters
-mb_info <boolean> E..V..... Set mb_info data through AVSideData, only useful when used from the API (default false)
[09:48:16]studio05@Studio05:~/Desktop$ x264 --fullhelp

31


-bash
--preset <string> Use a preset to select encoding settings [medium]
Overridden by user settings.
- ultrafast:
--no-8x8dct --aq-mode 0 --b-adapt 0
--bframes 0 --no-cabac --no-deblock
--no-mbtree --me dia --no-mixed-refs
--partitions none --rc-lookahead 0 --ref 1
--scenecut 0 --subme 0 --trellis 0
--no-weightb --weightp 0
- superfast:
--no-mbtree --me dia --no-mixed-refs
--partitions i8x8,i4x4 --rc-lookahead 0
--ref 1 --subme 1 --trellis 0 --weightp 1
- veryfast:
--no-mixed-refs --rc-lookahead 10
--ref 1 --subme 2 --trellis 0 --weightp 1
- faster:
--no-mixed-refs --rc-lookahead 20
--ref 2 --subme 4 --weightp 1
- fast:
--rc-lookahead 30 --ref 2 --subme 6
--weightp 1
- medium:
Default settings apply.
- slow:

32



Born Compressed

- optimised for both image acquisition and postproduction

Examples: CineForm RAW, ProRes RAW, Blackmagic RAW

Uncomfortable Truths

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

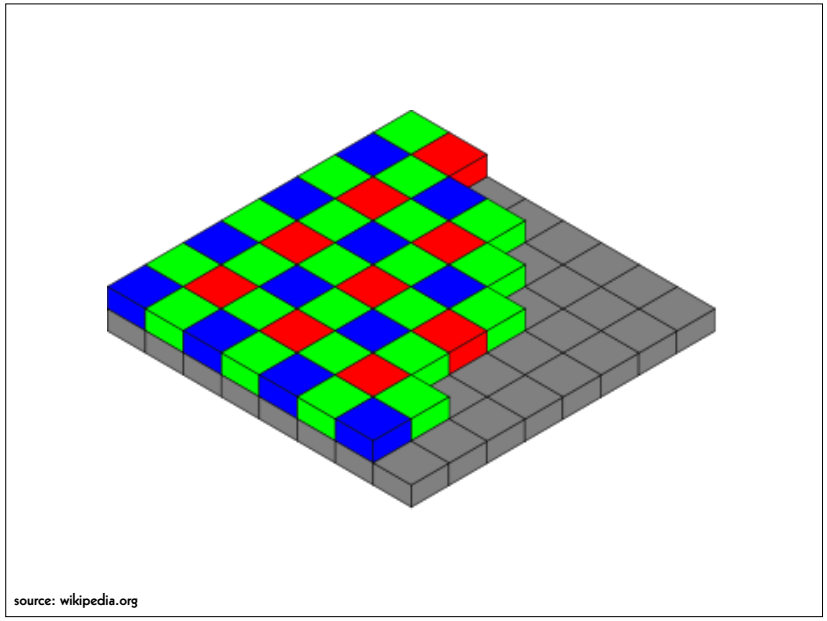
Proposed Terminology

pixel

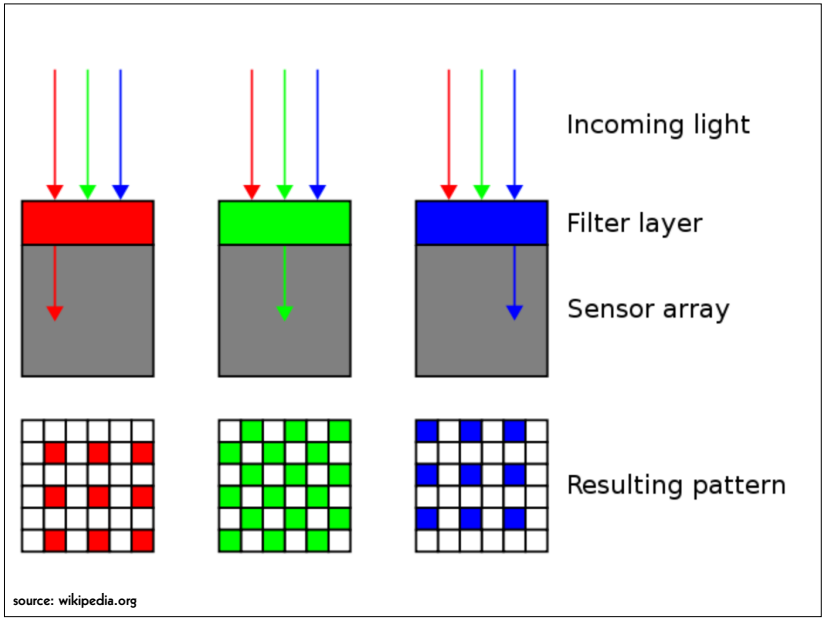
= picture element

sensel

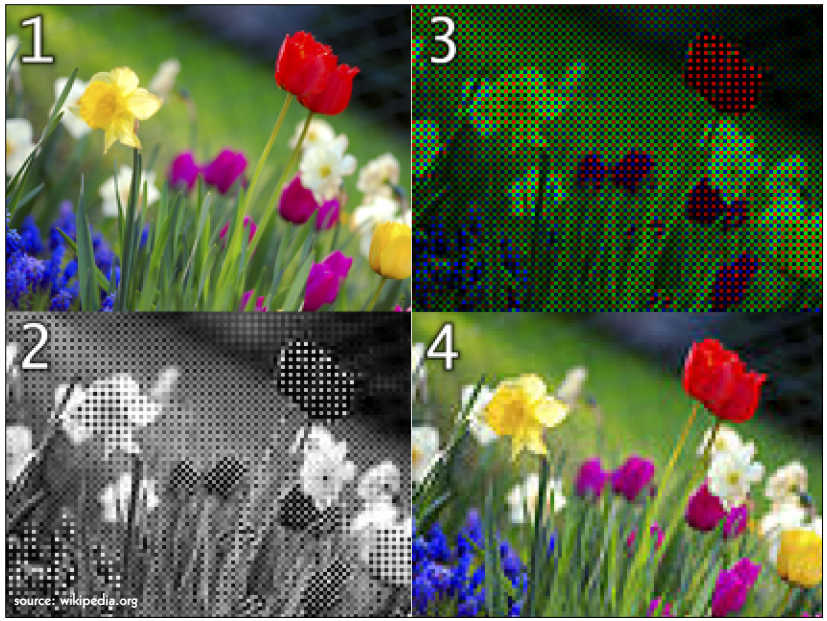
= sensor element


source: wikipedia.org

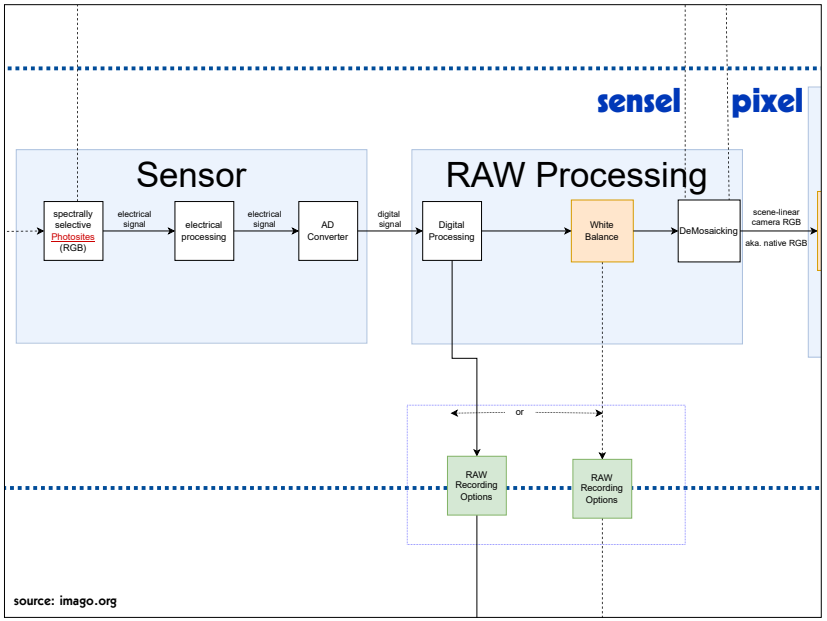
37


Incoming light
Filter layer
Sensor array
Resulting pattern
source: wikipedia.org

38


1
2
3
4
source: wikipedia.org

39


sensel
pixel
Sensor
RAW Processing
spectrally selective Photosites (RGB)
electrical signal
electrical processing
electrical signal
AD Converter
digital signal
Digital Processing
White Balance
DeMosaicking
scene-linear camera RGB
aka. native RGB
or
RAW Recording Options
RAW Recording Options
source: imago.org

40

The sensel values
are the raw ingredients
to cook the pixel values.

Ways to use Bayer-type data

digital blow-up to RGB

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

digital reduction to RGB

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

Ways to store Bayer-type data

- pixel values generated by a de-mosaicking algorithm (digital blow-up)
- pixel values generated by mixing two green sensel values into one (digital reduction)
- raw sensel values without de-mosaicking

Checksums

Algorithms

cryptographic

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

non-cryptographic

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

Checksum in Filename

from:

- title.extension

to:

- title_checksum.extension

Example:
Film_9d5084b5b0a08d5022b39e0e75241d12.mkv

Recommendations

- use non cryptographic algorithms
- compute checksums at file generation
- integrate checksums into filenames

Outlook

Analogue Preservation

- keep the source elements as long as possible by increasing prevention:
  - → better insulation
  - → more efficient air conditioning
  - → less handling
- make digital primary files and access copies

Digital Preservation

- keep born-digital source files at their highest quality
- avoid transcodig
- chose solid media container formats and video and audio codex'

Sustainability

from 2004 to 2012
- all electricity from the grid

from 2013 to 2019
- no electricity from the grid for the IT

from 2020 to 2024
- no electricity at all from the grid

AV Preservation by
reto.ch

Sandrainstrasse 3
3007 Bern
Switzerland

reto.ch
info@reto.ch