

# Presets for FFV1 and MKV: Choosing the right parameters for the job.

Peter Bubestinger-Steindl  
(p.bubestinger @ AV-RD.com)

November 2019

# FFV1 Video Codec

# Parameters

- coder
- context
- slices
- slicecrc
- level

# Coder

- 0 = **Golomb Rice** (Default 8bpc)
- 1 = **Range Coder** (Default > 8bpc)  
(with **default** transition table)
- 2 = Range Coder  
(with **custom** transition table)

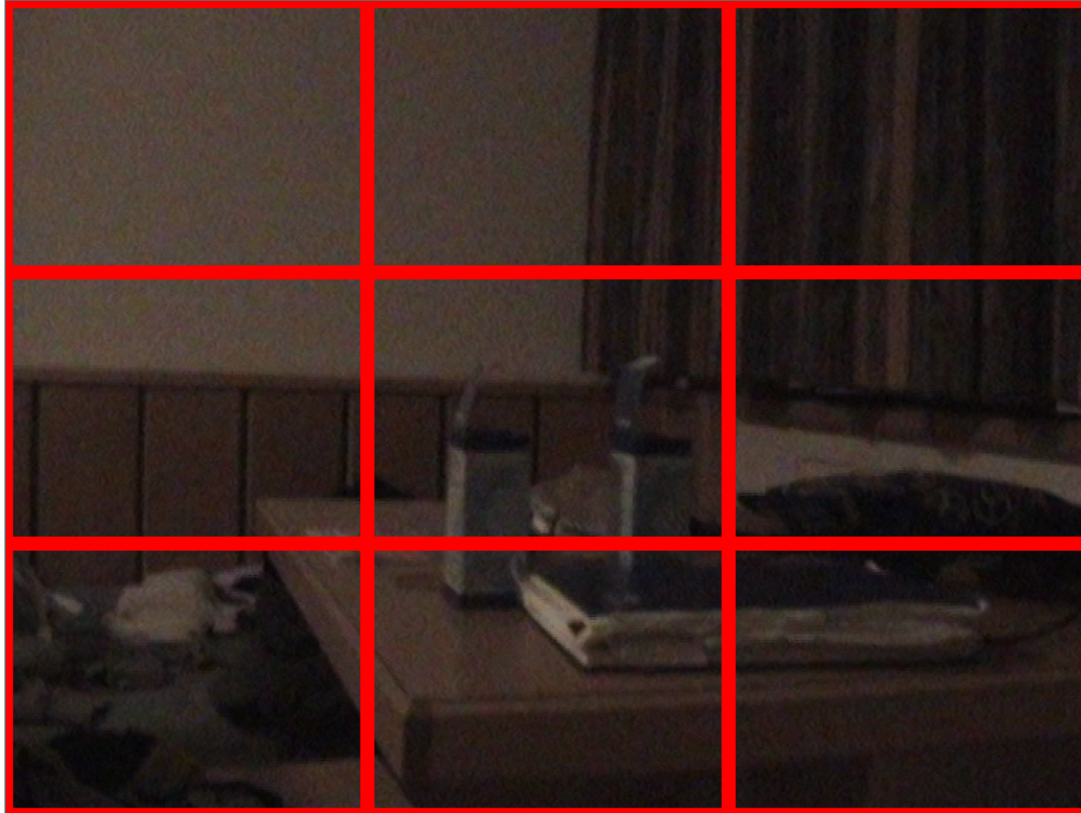
# Context

- 0 = small (Default)
- 1 = large

# Slices

- Image is divided into separate areas
- Used for parallel encoding
- Only with FFV1 version  $\geq 2$
- Each slice has its header
- But: More slices = slightly more space  
(a few bytes/slice)

# 9 Slices



# SliceCRC

- CRC = **Cyclic Redundancy Check**
- Checksum per slice
- More slices = more safety
- To *know* if a slice is okay.
- Decoders can choose:  
“what to do if it’s not?”



# Level = FFV1 version

- 0
- 1 (Default)
- **3** (Recommended)

# Scan type

*FFV1 can store field-order information independent of its container.*

# Scan type

aka “*picture structure*” (IETF specs)

- 0 = unknown
- 1 = top field first (TFF)
- 2 = bottom field first (BFF)
- 3 = progressive
- Other = reserved for future use

# Default values

- level: 1 (!)
- coder:
  - = 8bpc: 0 (golomb rice)
  - > 8bpc: 1 (range)
- context = 0 (small)
- slices = 4
- slicecrc = 1

# Matroska Container (MKV)

# Some Parameters

- SegmentUID
- SeekHead
- Scan type
- Color information:
  - Sample range
  - Color primaries
  - Transfer characteristics
  - Matrix coefficients

# SegmentUID

*“A randomly generated unique ID to identify the Segment amongst many others (128 bits).”*

# SegmentUID

*[Segment:] “The Root Element that contains all other Top-Level Elements (Elements defined only at Level 1). A Matroska file is composed of 1 Segment.”*

– **Matroska.org**



# SeekHead

*“Contains the Segment Position of other Top-Level Elements.”*

– **Matroska.org**

It can be used to uniquely identify *this* MKV file and is part of the “MetaSeek” section.

# MetaSeek

*“The Metaseek section contains an index of where all of the other groups are in the file are located [...].”*

*“This element isn’t technically required, but you would have to search the entire file to find all of the other Level 1 elements if you did not have it.*

*This is because any of the items can occur in any order.”*

*– **Matroska.org***



# Fixity

*CRC-32 per Element.*

The magic's inside EBML.

# Scan type

# FlagInterlaced

- 0 = undetermined
- 1 = interlaced
- 2 = progressive

# FieldOrder

- 0 = progressive
- 1 = top field first (TFF)
- 2 = undetermined
- 6 = bottom field first (BFF)
- 9 = BFF (swapped)
- 14 = TFF (swapped)

# Color information (MKV/FFmpeg)

- **colour-range** (full, broadcast)

```
-color_range mpeg
```

- **colour-primaries**

```
-color_primaries bt470bg
```

- **colour-transfer-characteristics**

```
-color_trc bt709
```

- **colour-matrix-coefficients**

```
-colorspace bt470bg
```

Source: **“mkvpropedit -l”**, **vrecord**



# Tagging

aka “Descriptive Metadata”.

*“[...] contains all of the Tags that relate to the the file and each of the tracks. These tags are just like the ID3 tags found in MP3’s.”*

# Attachments

*“The Attachment section is for attaching any type of file you want to a Matroska file. You could attach anything, pictures, webpages, programs, even the codec needed to play back the file.”*

– **Matroska.org**

# FFmpeg recipe (PAL)

```
ffmpeg -i VIDEO_IN \  
-c:v ffv1 -level 3 -coder 1 -context 0 -slices 24 -slicecrc 1 \  
-color_primaries bt470bg \  
-color_trc bt709 \  
-colorspace bt470bg \  
-color_range mpeg \  
-map 0 \  
-top 1 \  
-c:a copy \  
-g 1 -pix_fmt + \  
VIDEO_OUT.mkv
```

# MXF-like profiles or presets

- Currently don't exist.
- But might make sense to define some.
- Don't overdo it.
- Works well without yet.

# References

- **FFV1 Encoding (FFmpeg Wiki)**
- **FFV1 IETF Draft**
- **Matroska Specifications**
- **Matroska IETF Draft**
- **MKVToolNix**
- **EBML IETF Draft**

Questions?

Comments?