A/V Data Compression Basics

Peter Bubestinger-Steindl
(p.bubestinger@av-rd.com)

November 2019

Compressed? Uncompressed?

Speaker notes

What does that mean? What is “uncompressed”? What is “data compression”?

It’s not only technical, but highly political… Including Fear, Uncertainty and Doubt (FUD).

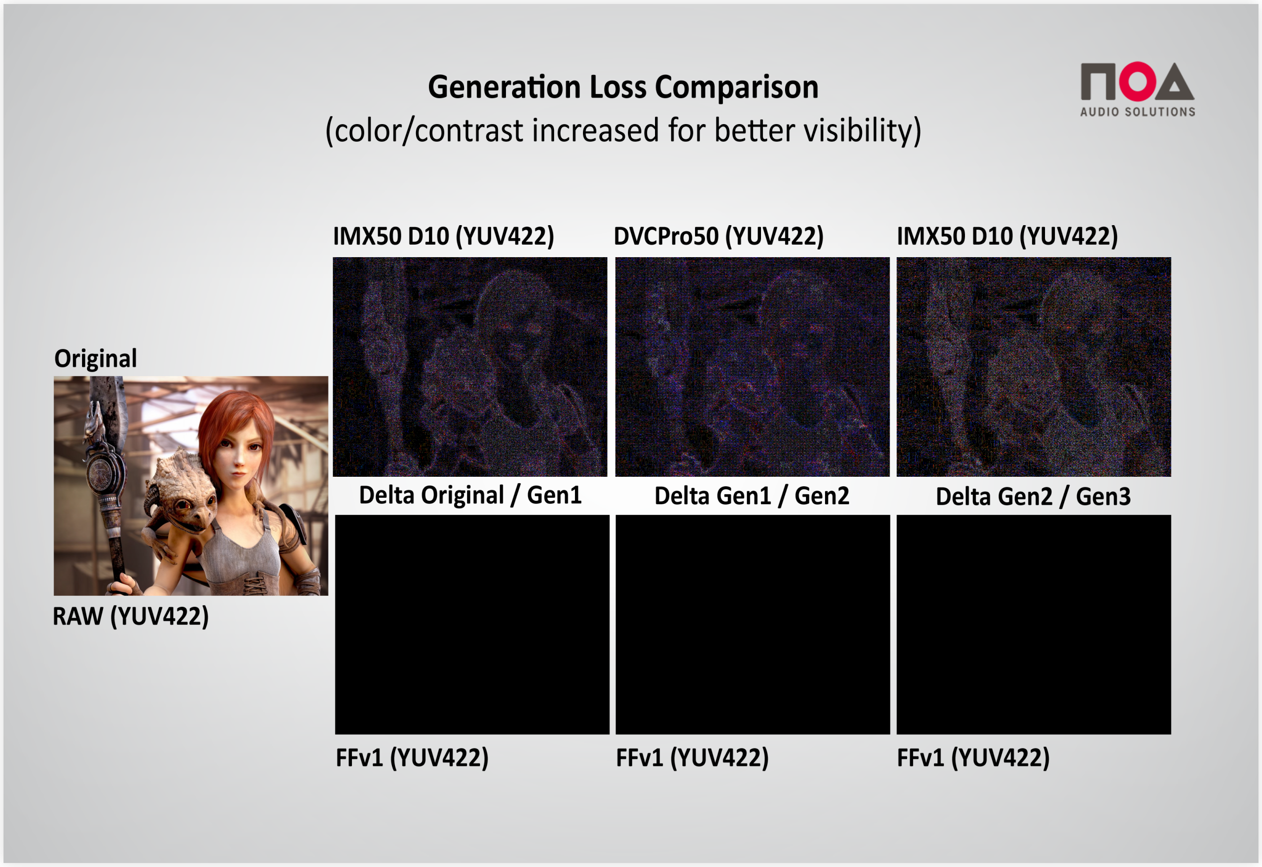
Current default = Lossy compression

Speaker notes

Of course this image is exaggerated, but it shows pretty well what lossy compression is and what artefacts a typical MPEG-like compression algorithm produces.

btw: This is a snapshot image of the highest-quality version of the video on the original website (around 2009).

Generation Loss



Generation Loss Comparison

(color/contrast increased for better visibility)

Original

RAW (YUV422)

IMX50 D10 (YUV422)

DVCPro50 (YUV422)

IMX50 D10 (YUV422)

Delta Original / Gen1

Delta Gen1 / Gen2

Delta Gen2 / Gen3

FFv1 (YUV422)

FFv1 (YUV422)

FFv1 (YUV422)

Speaker notes

Here’s an example where one can see the difference between each encoded version in popular broadcast codecs.

Uncompressed

- No generation loss
- Dead simple (=preserves well)
- The largest possible version (ca. 1.3 GB/minute PAL SD)
- There's *more* than just 1 "uncompressed"

Calculating filesize: IMAGE

- Width(px) x Height(px)
- x Bits-Per-Pixel(bpp)
- x FPS
- / 8 = **1 second (in Byte)**

Calculating filesize: AUDIO

- Samplerate x bit-depth
- x channels (even if silence!)
- / 8 = **1 second (in Byte)**

Speaker notes

So how large is an 8bpc SD PAL minute - with 4 channels audio at SDI standard (48kHz/16bit)?
Or a 2k 16bpc(!) scan (including 6 channels audio at 48kHz/24bit)?
Or 74min. of audio CD (red-book standard: 44.1kHz/16bit)?

Lossless

> *"It's like ZIP for film!"*

- No generation loss
- Way larger than lossy
- But: Smaller than uncompressed

Behind the Scenes: Algorithms

Speaker notes

There are several different algorithms used for compression. Each has its pros and cons. Here are on

Run Length Enoding (RLE)

- Uncompressed: `BBBBBBBBGGBBBBBRRRBBBB` = 19 Bytes
- Compressed: `8B2G5B3R4B` = 10 Byte

Speaker notes

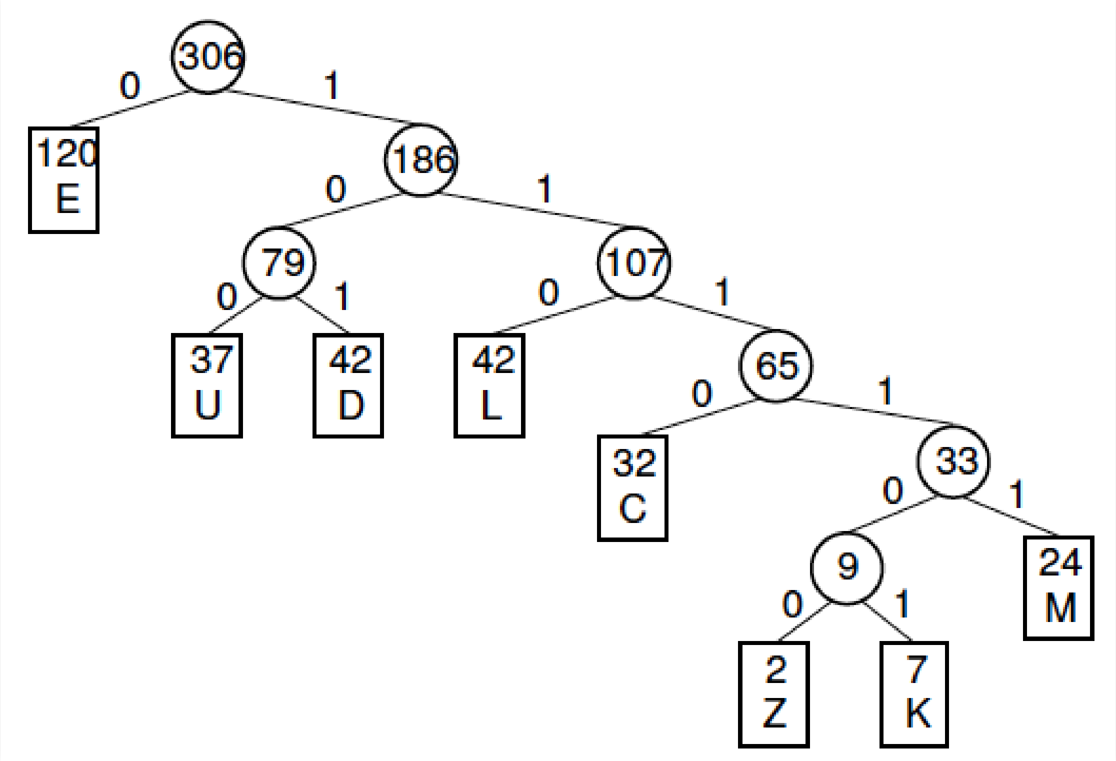
This could be seen as “delta coding”, since only *changes* between 2 characters really require space.

Huffman

Letter	Freq
E	120
D	42
L	42
U	37
C	32
M	24
K	7
Z	2

Source: Indiana University

Huffman Tree



Huffman Code

Letter	Freq	Code	Bits
E	120	0	1
D	42	101	3
L	42	110	3
U	37	100	3
C	32	1110	4
M	24	11111	5
K	7	111101	6
Z	2	111100	6

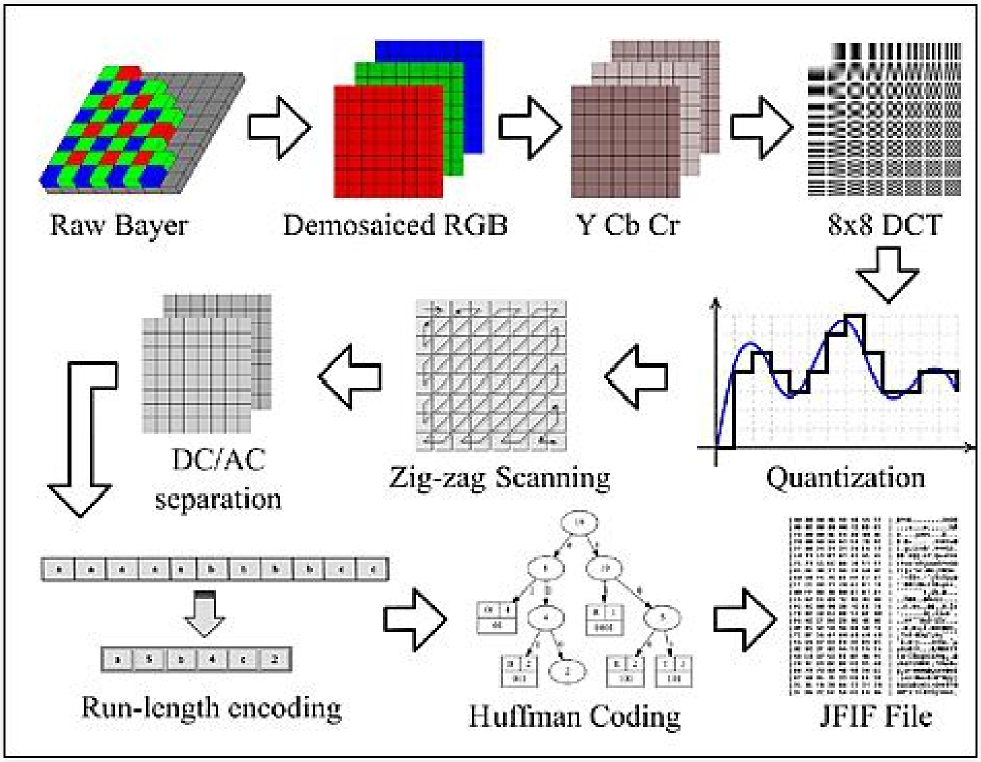
Speaker notes

Interesting: For very infrequent characters, the Huffman coding might even take more bits than an uncompressed representation.

This is especially problematic for noise signals, because: noise = chaos = no frequency pattern.

Lossy

JPEG



Terms used

- lossy
- lossless
- uncompressed
- data reduced format
- compressed
- near-lossless
- visually lossless
- mathematically lossless

Format examples: VIDEO

Lossy	**Lossless**	**Uncompressed**
MPEG-1,2,4	FFV1	v210
JPEG2000	JPEG2000	UYVY
ProRes	Dirac	YUY2
H.264	H.264	H.264
H.265	H.265	RGBA
...	...	...

Compression: Spatial and Temporal

A/V allows to apply compression in 2 ways:

- **Spatial:** Inside an image
- **Temporal:** Between images

Speaker notes

The 1st one doesn’t differ much from “image only” compression algorithms (JPEG, RLE, etc).

Temporal compression can be seen as “delta coding”, similar to RLE: You only write down what *changes* between 2 images/samples, as you would only write down what changes between 2 characters in text compression.

Spatial compression: Examples

- **Audio:** Differential pulse-code modulation (DPCM)
- **Video:** GOP, Motion estimation

Data rate / Bitrate

> *“Data per time”*

- Mbps / 8 = MB / second
- MB/s * 60 = MB / minute
- MB/min * 60 = MB / hour

Speaker notes

For lossy formats, this is a tuning parameter that can be used to adjust the final filesize and quality. For lossless or uncompressed formats, the actual bitrate can be calculated but *not* adjusted. Because the format cannot throw anything away, so there’s the bitrate is determined by content+algorithm and not a parameter.

Bitrate means “data per time”, so this means:

- Higher bitrate = larger files, but higher quality
- Lower bitrate = smaller files, but lower quality The gain of newer encoding algorithms is always to produce smaller files with higher quality.

Data rate / Bitrate

- **Higher bitrate** =
  larger files, higher quality
- **Lower bitrate** =
  smaller files, lower quality

Links

- Huffman coding (Indiana.edu)
- EETimes: Data compression tutorial: Part 1, Part 2 (LZW), Part 3 (JPEG + MPEG)
- Introduction to Data Compression (blellochcs.cmu.edu, 2013)