

Test Framework Specs

Reference:

https://arewecompressedyet.com/?job=CL97541_p5_tx_count_cpu3%402019-10-17T09%3A26%3A07.137Z&job=CL97321_p6_txsize_kf_60f_cpu3%402019-10-14T13%3A08%3A04.892Z

Use cases

Exact Match

Comparing encoded results to original media for comparison of RGB 2 YCrCb conversion. For this case we want minimal encoding and want to compare exactly.

The source input would be a known set of still frames, and the tests would be limited number of conversions. Typically the sources would be single frames, and probably not more than 2-3 image sequences.

<https://richardssam.github.io/ffmpeg-tests/tests/chip-chart-yuvconvert/compare.html> is a example output.

Need: Critical.

These would be used as part of a unit-test framework.

Approximate match to baseline.

Matching encoded results to original media, but results will not be exact. We want to use VMAF to help with approximate comparison.

We would be using this to confirm that known baselines are creating the same result.

The sources would number no more than 2-3 image sequences.

Need: Critical.

These would be used as part of a unit-test framework.

Wedging results

What we would be using this for is to provide end-users an idea of what certain settings produce.

We would like to be able to produce charts like in:

<https://codecalamity.com/hardware-encoding-4k-hdr10-videos/>

- VMAF vs Bitrate for 4 different encoders – 5 different bitrate settings, for a single piece of media.

<https://codecalamity.com/encoding-uhd-4k-hdr10-videos-with-ffmpeg/>

- VMAF vs. CRF for 10 settings. (Seemingly testing for a range of CRF calies)
- Duration vs. CRF

Note, these are probably good for one off tests, to create documents, although might be good for re-running tests for new versions of encoders.

The creation of graphs would be nice to be automatic, but definately a bonus, if somebody ended up doing those "by hand" it would be fine.

It would also be good if any resulting graph could be easily embedded into MD content.